

Deepak Gupta · Zdzislaw Polkowski ·
Ashish Khanna ·
Siddhartha Bhattacharyya ·
Oscar Castillo *Editors*

Proceedings of Data Analytics and Management

ICDAM 2021, Volume 2

Lecture Notes on Data Engineering and Communications Technologies

Volume 91

Series Editor

Fatos Xhafa, Technical University of Catalonia, Barcelona, Spain

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

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

Indexed by SCOPUS, INSPEC, EI Compendex.

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

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

Deepak Gupta · Zdzislaw Polkowski ·
Ashish Khanna · Siddhartha Bhattacharyya ·
Oscar Castillo
Editors

Proceedings of Data Analytics and Management

ICDAM 2021, Volume 2



Springer

Editors

Deepak Gupta
Maharaja Agrasen Institute of Technology
New Delhi, Delhi, India

Zdzislaw Polkowski
Jan Wyzykowski University
Polkowice, Poland

Ashish Khanna
Maharaja Agrasen Institute of Technology
New Delhi, Delhi, India

Siddhartha Bhattacharyya
Rajnagar Mahavidyalaya
Birbhum, West Bengal, India

Oscar Castillo
Tijuana Institute of Technology
Tijuana, Mexico

ISSN 2367-4512

ISSN 2367-4520 (electronic)

Lecture Notes on Data Engineering and Communications Technologies

ISBN 978-981-16-6284-3

ISBN 978-981-16-6285-0 (eBook)

<https://doi.org/10.1007/978-981-16-6285-0>

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

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

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

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

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

Dr. Deepak Gupta would like to dedicate this book to his father Sh. R. K. Gupta, his mother Smt. Geeta Gupta for their constant encouragement, his family members including his wife, brothers, sisters, kids, and to my students close to my heart.

Dr. Zdzislaw Polkowski would like to dedicate this book to his Wife, Daughter, and Parents.

Dr. Ashish Khanna would like to dedicate this book to his mentors Dr. A. K. Singh and Dr. Abhishek Swaroop for their constant encouragement and guidance and his family members including his mother, wife, and kids. He would also like to dedicate this work to his (Late) father Sh. R. C. Khanna with folded hands for his constant blessings.

Prof. (Dr.) Siddhartha Bhattacharyya would like to dedicated this book to Dr. Sujit Pal, Joint Director of Public Instructions, Department of Higher Education, Government of West Bengal.

ICDAM Committees

ICDAM-2021 Steering Committee Members

Patrons

Dr. Tadeusz Kierzyk, Professor of UJW, Rector of Jan Wyzykowski University, Polkowice, Poland
Shri Rakesh Tayal, Panipat Institute of Engineering and Technology, India
Shri Hariom Tayal, Panipat Institute of Engineering and Technology, India
Shri Shubham Tayal, Panipat Institute of Engineering and Technology, India
Shri Suresh Tayal, Panipat Institute of Engineering and Technology, India
Dr. Ashok Gupta, IIS Deemed to be University, Jaipur
Shri B.S Yadav, Hon'ble Chancellor-IES University, Bhopal
Dr. Sunita Singh, Hon'ble Pro Chancellor, IES University, Bhopal
Shri Devansh Singh, CEO-IES University, Bhopal

General Chairs

Prof. Dr. Janusz Kacprzyk, Polish Academy of Sciences, Systems Research Institute, Poland
Prof. Dr. Cesare Alippi, Polytechnic University of Milan, Italy
Prof. Dr. Siddhartha Bhattacharyya, CHRIST (Deemed to be University), Bangalore, India
Prof. Dr. Shakti Kumar, Panipat Institute of Engineering and Technology, India
Prof. T. N. Mathur, IIS Deemed to be University, Jaipur
Dr. Jyotiram Sawale, Registrar, IES University, Bhopal

Honorary Chairs

Prof. Dr. Aboul Ella Hassanien, Cairo University, Egypt
Prof. Dr. Vaclav Snasel, Rector, VSB-Technical University of Ostrava, Czech Republic
Dr. Raakhi Gupta, IIS Deemed to be University, Jaipur
Dr. Naveen Chandra, Vice Chancellor, IES University, Bhopal

Conference Chairs

Dr. Zdzislaw Polkowski, Professor of UJW, Jan Wyzykowski University, Polkowice, Poland

Prof. Dr. Joel J. P. C. Rodrigues, Universidade Estadual do Piau Teresina, Brazil

Prof. Dr. Abhishek Swaroop, Bhagwan Parshuram Institute of Technology, Delhi, India

Prof. Dr. Anil K Ahlawat, KIET Group of Institutes, Ghaziabad, India

Prof. Dr. Vijay Athavale, Panipat Institute of Engineering and Technology, India

Dr. Suresh Chand Gupta, Panipat Institute of Engineering and Technology, India

Prof. Vijay Singh Rathore, IIS Deemed to be University, Jaipur

Dr. O. P. Modi, Principal IES Institute of Technology and Management, IES University, Bhopal

Dr. G. K. Pandey, Principal, IES College of Technology, Bhopal

Technical Program Chairs

Dr. Stanislaw Piesiak, Professor of UJW, Jan Wyzykowski University, Polkowice, Poland

Dr. Jan Walczak, Jan Wyzykowski University, Polkowice, Poland

Dr. Anna Wojciechowicz, Jan Wyzykowski University, Lubin, Poland

Dr. Anju Bhandari, Panipat Institute of Engineering and Technology, India

Dr. Dinesh Verma, Panipat Institute of Engineering and Technology, India

Dr. Arti Jain, Jaypee Institute of Information Technology (JIIT)

Dr. Sushil Kumar Singh, Seoul National University of Science and Technology, Seoul, South Korea

Dr. Pallavi Bhatnagar, IES College of Technology, Bhopal

Dr. Pramod Kumar Patel, IES College of Technology, Bhopal

Dr. Anil Kumar Yadav, IES College of Technology, Bhopal

Dr. Nikhat Raza Khan, IES College of Technology, Bhopal

Dr. Rajesh Kumar Nema, IES College of Technology, Bhopal

Dr. Jitendra Mathur, IES College of Technology, Bhopal

Prof. Khushbu Kriplani, IES College of Technology, Bhopal

Prof. Sonu Lal, IES College of Technology, Bhopal

Prof. Jamvant Omkar, IES College of Technology, Bhopal

Technical Program Co-chairs

Prof. Dr. Victor Hugo C. de Albuquerque, Universidade de Fortaleza, Brazil

Dr. Gulshan Shrivastava, Sharda University, Gr. Noida, India

Dr. Akhilesh Kumar Mishra, Panipat Institute of Engineering and Technology, India

Dr. Mukesh Chawla, Panipat Institute of Engineering and Technology, India

Rattendeep Aneja, Panipat Institute of Engineering and Technology, India

Conveners

Dr. Ashish Khanna, Maharaja Agrasen Institute of Technology (GGSIPU), New Delhi, India

Dr. Deepak Gupta, Maharaja Agrasen Institute of Technology (GGSIPU), New Delhi, India

Dr. Pradeep Kumar Mallick, KIIT Deemed to be University Bhubaneswar, Odisha, India

Dr. Akash Kumar Bhoi, Sikkim Manipal University, India

Dr. Bhawna Singla, Panipat Institute of Engineering and Technology, India

Prof. K. S. Sharma, IIS Deemed to be University, Jaipur

Publication Chairs

Dr. Jerzy Widerski, Professor of UJW, V-Rector of Jan Wyzykowski University, Polkowice, Poland

Dr. Vicente García Díaz, University of Oviedo, Spain

Akanksha, Panipat Institute of Engineering and Technology, India

Sandeep Jaglan, Panipat Institute of Engineering and Technology, India

Shakti Arora, Panipat Institute of Engineering and Technology, India

Publicity Chairs

Dr. Paweł Gren, Professor of UJW, V-Rector of Jan Wyzykowski University, Polkowice, Poland

Dr. Aditya Khamparia, Lovely Professional University, Punjab, India

Saurab Gupta, Panipat Institute of Engineering and Technology, India

Tarun Miglani, Panipat Institute of Engineering and Technology, India

Ms. Ginni, Panipat Institute of Engineering and Technology, India

Co-conveners

Dr. Deepak Wadhwa, Panipat Institute of Engineering and Technology, India

Mr. Moolchand Sharma, Maharaja Agrasen Institute of Technology, India

Dr. Vaishali Mehta, Panipat Institute of Engineering and Technology, India

Dr. Anubha Jain, IIS Deemed to be University, Jaipur

Advisory Committee

Dr. Tadeusz Kierzyk, Professor of UJW, Rector of Jan Wyzykowski University, Polkowice, Poland

Prof. Vincenzo Piuri, University of Milan, Italy

Prof. Aboul Ella Hassanien, Cario University, Egypt

Prof. Marcin Paprzycki, Polish Academy of Science, Poland

Prof. Valentina Emilia Balas, Aurel Vlaicu University of Arad, Romania

Prof. Marius Balas, Aurel Vlaicu University of Arad, Romania

Prof. Mohamed Salim Bouhlel, Sfax University, Tunisia

Prof. Cenap Ozel, King Abdulaziz University, Saudi Arabia

Prof. Ashiq Anjum, University of Derby, Bristol, UK

Prof. Mischa Dohler, King's College London, UK

- Prof. David Camacho, Universidad Autonoma de Madrid, Spain
Prof. Parmanand, Dean, Galgotias University, UP, India
Prof. Maryna Yena, Medical University of Kiev, Ukraine
Prof. Giorgos Karagiannidis, Aristotle University of Thessaloniki, Greece
Prof. Tanuja Srivastava, Department of Mathematics, IIT Roorkee
Dr. D. Jude Hemanth, Karunya University, Coimbatore
Prof. Tiziana Catarci, Sapienza University, Rome, Italy
Prof. Salvatore Gaglio, University Degli Studi di Palermo, Italy
Prof. Bozidar Klicek, University of Zagreb, Croatia
Prof. A. K. Singh, NIT Kurukshetra, India
Prof. Anil Kumar, KIET Group of Institutes, India
Prof. Chang-Shing Lee, National University of Tainan, Taiwan
Dr. Paolo Bellavista, Alma Mater Studiorum—Università di Bologna
Prof. Sanjay Misra, Covenant University, Nigeria
Prof. Benatiaillah Ali, Adrar University, Algeria
Prof. Suresh Chandra Satapathy, PVPSIT, Vijayawada, India
Prof. Marylene Saldon-Eder, Mindanao University of Science and Technology
Prof. Özlem Onay, Anadolu University, Eskisehir, Turkey
Prof. Kei Eguchi, Department of Information Electronics, Fukuoka Institute of Technology
Prof. Zoltan Horvath, Kasetart University
Dr. A. K. M. Matiul Alam Vancouver British Columbia, Canada
Prof. Joong Hoon Jay Kim, Korea University
Prof. Sheng-Lung Peng, National Dong Swa University, Taiwan
Dr. Dusanka Boskovic, University of Sarajevo, Sarajevo
Dr. Periklis Chat Zimisios, Alexander TEI of Thessaloniki, Greece
Dr. Nhu Gia Nguyen, Duy Tan University, Vietnam
Dr. Ahmed Faheem Zobaa, Brunel University, London
Prof. Ladjel Bellatreche, Poitiers University, France
Prof. Victor C. M. Leung, The University of British Columbia, Canada
Prof. Huseyin Irmak, Cankiri Karatekin University, Turkey
Dr. Alex Norta, Tallinn University of Technology, Estonia
Prof. Amit Prakash Singh, GGSIPU, Delhi, India
Prof. Abhishek Swaroop, Bhagwan Parshuram Institute of Technology, Delhi
Prof. Christos Douligeris, University of Piraeus, Greece
Dr. Brett Edward Trusko, President & CEO (IAOIP) and Assistant Professor, Texas A&M University, Texas
Prof. Joel J. P. C. Rodrigues, National Institute of Telecommunications (Inatel), Brazil; Instituto de Telecomunicações, Portugal
Prof. Victor Hugo C. de Albuquerque, University of Fortaleza (UNIFOR), Brazil
Dr. Atta ur Rehman Khan, King Saud University, Riyadh
Dr. João Manuel R. S. Tavares, FEUP—DEMec
Prof. Ku Ruhana Ku Mahamud, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Malaysia
Prof. Ghasem D. Najafpour, Babol Noshirvani University of Technology, Iran

- Prof. Sanjeevikumar Padmanaban, Aalborg University, Denmark
Prof. Frede Blaabjerg, President (IEEE Power Electronics Society), Aalborg University, Denmark
Prof. Jens Bo Holm Nielson, Aalborg University, Denmark
Dr. Abu Yousuf, University Malaysia Pahang Gambang, Malaysia
Dr. Ahmed A. Elngar, Faculty of Computers and Information, Beni-Suef University, Egypt
Prof. Dijana Oreski, Faculty of Organization and Informatics, University of Zagreb, Varazdin, Croatia
Prof. Prasad K. Bhaskaran, Ocean Engineering and Naval Architecture, IIT Kharagpur
Dr. Yousaf Bin Zikria, Yeungnam University, South Korea
Dr. Sanjay Sood, C-DAC, Mohali
Prof. Ajay Rana, Senior Vice President and Advisor—Amity Education Group, Amity University, Noida, India
Dr. Florin Popentiu Vladicescu, University Politehnica of Bucharest, Romania
Dr. Paweł Gren, Professor of UJW, Jan Wyżykowski University, Polkowice, Poland
Prof. Joanna Jozefowska, Pro-Rector for Research (etc.) of Poznan University of Technology
Prof. Gerhard-Wilhelm Weber, Poznan University of Technology, Poland
Prof. Dr. Sung-Bae Cho, Yonsei University, South Korea
Prof. Carlos A. Coello Coello, CINVESTA, Mexico
Dr. L. C. Jain, Founder, KES International and Adjunct Professor, University of Canberra, Australia
Dr. Debahuti Mishra, ITER, SOA University, Odisha, India
Dr. Ebrahim Aghajari, Islamic Azad University of Ahvaz, (IAUA), Iran
Dr. Hongyan Yu, Department of Computer Science Shanghai Maritime University, Shanghai, China
Dr. Benson Edwin, Raj Higher College of Technology Fujairah Women's College, United Arab Emirates
Dr. Mohd. Hussain, Faculty of Computer Science and Information System, Islamic University, Madina Saudi, Arabia
Dr. Vahid Esmaeelzadeh, Department of Computer Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran
Dr. Avinash Konkani Clinical Engineer, University of Virginia Health System Charlottesville, Virginia, USA
Prof. Yu-Min Wang, National Chi Nan University, Taiwan
Dr. Ganesh R. Naik, Centre for Health Technologies (CHT), Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia
Dr. Yiguang Liu, Institutes of Image and Graphics, College of Computer Science and Engineering, Yihuan Road, Chengdu Sichuan, China
Dr. Karol Morawski, The Karkonosze University of Applied Sciences, Jelenia Gora, Poland

Preface

We hereby are delighted to announce that Jan Wyżykowski University, Polkowice, Poland, Panipat Institute of Engineering & Technology, IIS University, and IES University, India, has hosted the eagerly awaited and much coveted International Conference on Data Analytics and Management (ICDAM-2021). The second version of the conference was able to attract a diverse range of engineering practitioners, academicians, scholars, and industry delegates, with the reception of abstracts including more than 2600 authors from different parts of the world. The committee of professionals dedicated toward the conference is striving to achieve a high-quality technical program with tracks on data analytics, data management, big data, computational intelligence, and communication networks. All the tracks chosen in the conference are interrelated and are very famous among the present-day research community. Therefore, a lot of research is happening in the above-mentioned tracks and their related sub-areas. More than 650 full-length papers have been received, among which the contributions are focused on theoretical, computer simulation-based research, and laboratory-scale experiments. Among these manuscripts, 131 papers have been included in Springer proceedings after a thorough two-stage review and editing process. All the manuscripts submitted to ICDAM-2021 were peer-reviewed by at least two independent reviewers, who were provided with a detailed review proforma. The comments from the reviewers were communicated to the authors, who incorporated the suggestions in their revised manuscripts. The recommendations from two reviewers were taken into consideration while selecting a manuscript for inclusion in the proceedings. The exhaustiveness of the review process is evident, given the large number of articles received addressing a wide range of research areas. The stringent review process ensured that each published manuscript met the rigorous academic and scientific standards. It is an exalting experience to finally see these elite contributions materialize into the two book volumes as ICDAM proceedings by Springer entitled “Data Analytics and Management: Proceedings of ICDAM.”

ICDAM-2021 invited three key note speakers, who are eminent researchers in the field of computer science and engineering, from different parts of the world. In addition to the plenary sessions on each day of the conference, twelve concurrent technical sessions are held every day to assure the oral presentation of around

131 accepted papers. Keynote speakers and session chair(s) for each of the concurrent sessions have been leading researchers from the thematic area of the session. The delegates were provided with a book of extended abstracts to quickly browse through the contents, participate in the presentations, and provide access to a broad audience of the audience. The research part of the conference was organized in a total of 40 special sessions. These special sessions provided the opportunity for researchers conducting research in specific areas to present their results in a more focused environment.

An international conference of such magnitude and release of ICDAM-2021 proceedings by Springer has been the remarkable outcome of the untiring efforts of the entire organizing team. The success of an event undoubtedly involves the painstaking efforts of several contributors at different stages, dictated by their devotion and sincerity. Fortunately, since the beginning of its journey, ICDAM-2021 has received support and contributions from every corner. We thank them all who have wished the best for ICDAM-2021 and contributed by any means toward its success. The edited proceedings volumes by Springer would not have been possible without the perseverance of all the steering, advisory, and technical program committee members.

All the contributing authors owe thanks from the organizers of ICDAM-2021 for their interest and exceptional articles. We would also like to thank the authors of the papers for adhering to the time schedule and for incorporating the review comments. We wish to extend our heartfelt acknowledgment to the authors, peer reviewers, committee members, and production staff whose diligent work put shape to ICDAM-2021 proceedings. We especially want to thank our dedicated team of peer reviewers who volunteered for the arduous and tedious step of quality checking and critique on the submitted manuscripts. We wish to thank our faculty colleague Mr. Moolchand Sharma for extending their enormous assistance during the conference. The time spent by them and the midnight oil burnt are greatly appreciated, for which we will ever remain indebted. The management, faculties, administrative, and support staff of the college have always been extending their services whenever needed, for which we remain thankful to them.

Lastly, we would like to thank Springer for accepting our proposal for publishing ICDAM-2021 proceedings. Help received from Mr. Aninda Bose, the acquisition senior editor, in the process has been very useful.

New Delhi, India

Ashish Khanna
Deepak Gupta
Organizers, ICDAM-2021

About This Book

International Conference on Data Analytics and Management (ICDAM-2021) was held on June 26, 2021, via virtual mode and jointly organized by Jan Wyzykowski University, Polkowice, Poland, Panipat Institute of Engineering & Technology, IIS University, and IES University, India. This conference was able to attract a diverse range of engineering practitioners, academicians, scholars, and industry delegates, with the reception of papers including more than 2600 authors from different parts of the world. Only 131 papers have been accepted and registered with an acceptance ratio of 20% to be published in the two volumes of prestigious Springer *Lecture Notes on Data Engineering and Communications Technologies* series. This volume includes a total of 65 papers.

Contents

CNN Based Feature Extraction for Visual Speech Recognition in Malayalam	1
Shabina Bhaskar and T. M. Thasleema	
Effective Rate of Minority Class Over-Sampling for Maximizing the Imbalanced Dataset Model Performance	9
Forhad An Naim, Ummae Hamida Hannan, and Md. Humayun Kabir	
Malaria Cell Image Classification Using Convolutional Neural Networks (CNNs)	21
Drishti Agarwal, K. Sashanka, Sajal Madan, Akshay Kumar, Preeti Nagrath, and Rachna Jain	
Automating Live Cricket Commentary Using Supervised Learning	37
Aniket S. Hegde, Kaustubh Jha, S. Suganthi, and Prasad B. Honnavalli	
Real-Time Object Detection and Distance Approximation	49
Rohit Beniwal and Ashish Singh	
Spectral Efficiency Analysis of Massive MIMO	61
Shubham Mittal, Anuj Singal, Kuldeep Singh, and Manisha Jangra	
Recommendations for DDOS Threats Using Tableau	73
Sagar Pande, Aditya Kamparia, and Deepak Gupta	
Sinkhole Attack Detection in Wireless Sensor Networks	85
Aina Mehta, Jasminder Kaur Sandhu, Meena Pundir, Rajwinder Kaur, and Luxmi Sapra	
HPGAB3C: A Novel Hybridized Optimization Approach	95
Rattan Deep Aneja, Amit Kumar Bindal, and Shakti Kumar	
COVID-19 Identification on Chest X-rays with Deep Learning Technique	113
Preeti Sharma and Devershi Pallavi Bhatt	

IoT-Cloud Enabled Statistical Analysis and Visualization of Air Pollution Data in India	125
Manzoor Ansari and Mansaf Alam	
Dental Cavity Detection Using YOLO	141
Apurva Sonavane and Rachna Kohar	
Development of Data Set for Automatic News Telecast System for Deaf Using ISL Videos	153
Annu Rani, Vishal Goyal, and Lalit Goyal	
A Comprehensive Survey on Content-Based Image Retrieval Using Machine Learning	165
Milind V. Lande and Sonali Ridhorkar	
Keyphrase Extraction from Twitter Data—A Supervised Deep Learning Approach	181
K. B. J. Lemuel and V. Subramaniyaswamy	
Improving the Yield and Revenue of Indian Crop Production Using Data Engineering	197
Jayashree Domala, Manmohan Dogra, Kevin Dsouza, Dwayne Fernandes, and Anuradha Srinivasaraghavan	
Designing an LSTM and Genetic Algorithm-based Sentiment Analysis Model for COVID-19	209
Poonam Rani, Jyoti Shokeen, Arjun Majithia, Amit Agarwal, Ashish Bhatghare, and Jigyasu Malhotra	
Machine Learning Techniques for Keystroke Dynamics	217
Kirty Shekhawat and Devershi Pallavi Bhatt	
Detection of Denial-of-Service Attacks Using Stacked LSTM Networks	229
Deepa Krishnan	
Concept of Hybrid Models in Background Subtraction: A Review of Recent Trends	241
Saumya Maurya and Mahipal Singh Choudhry	
Artificial Neural Network Approach for Multimodal Biometric Authentication System	253
M. J. Sudhamani, Ipsita Sanyal, and M. K. Venkatesha	
Malware Classification and Defence Against Adversarial Attacks	267
Aayush Kamath, Vrinda Bhau, Tejas Paranjape, and Rupali Sawant	
A Novel Ensemble Machine Learning Model for Prediction of Zika Virus T-Cell Epitopes	275
Syed Nisar Hussain Bukhari, Amit Jain, and Ehtishamul Haq	

A Deep Learning Approach for Detection of SQL Injection Attacks Using Convolutional Neural Networks	293
Ayush Falor, Manav Hirani, Henil Vedant, Priyank Mehta, and Deepa Krishnan	
Machine Learning, Deep Learning and Image Processing for Healthcare: A Crux for Detection and Prediction of Disease	305
Charu Chhabra and Meghna Sharma	
Dynamic Pricing-Based E-commerce Model for the Produce of Organic Farming in India: A Research Roadmap with Main Advertence to Vegetables	327
Sita Rani, Vivek Arya, and Aman Kataria	
Deep Learning Techniques for Detection of Autism Spectrum Syndrome (ASS)	337
Anshu Sharma and Poonam Tanwar	
Test Suite Minimization Based upon CMIMX and ABC	347
Neeru Ahuja and Pradeep Kumar Bhatia	
Feature Selection for Bi-objective Stress Classification Using Emerging Swarm Intelligence Metaheuristic Techniques	357
Prableen Kaur, Ritu Gautam, and Manik Sharma	
Regulated Energy Harvesting Scheme for Self-Sustaining WSN in Precision Agriculture	367
Kunal Goel and Amit Kumar Bindal	
Empirical Analysis of Facial Expressions Based on Convolutional Neural Network Methods	387
Rohit Pratap Singh and Laiphrajkpam Dolendro Singh	
An Advanced Hybrid Algorithm for Real-World Optimization Problem	397
Raghav Prasad Parouha	
An Online Document Emoji-Based Classification Using Twitter Dataset	409
Shelley Gupta, Archana Singh, and Jayanthi Ranjan	
Approaches to Optimize Memory Footprint for Elephant Flows	419
Vivek Kumar, Dilip K. Sharma, and Vinay K. Mishra	
Statistical Significance of Wilson Amplitude Towards the Identification and Classification of Murmur from Phonocardiogram	431
P. Careena, M. Mary Synthuja Jain Preetha, and P. Arun	

Comparative Analysis on Machine Learning Methodologies for the Effective Usage of Medical WSNs	441
Shivani G. Dharmale, Snehal A. Gomase, and Sagar Pande	
Convolutional Neural Networks for Malaria Image Classification	459
Kanchan M. Pimple, Praveen P. Likhitkar, and Sagar Pande	
Identification of Characters (Digits) Through Customized Convolutional Neural Network	471
Swati C. Tawalare, Nikhil E. Karale, Sagar Pande, and Aditya Khamparia	
Breast Cancer Detection Using Image Processing and CNN Algorithm with K-Fold Cross-Validation	481
Pruthvi Tilekar, Purnima Singh, Nagnath Aherwadi, Sagar Pande, and Aditya Khamparia	
Pilot Decontamination Using Sector Base Method in Massive MIMO System	491
Dikshit Kalyal, Paras Chawla, and Rajpreet Singh	
Applications of Deep Learning in Diabetic Retinopathy Detection and Classification: A Critical Review	505
Preeti Kapoor and Shaveta Arora	
A Study of Recommendation System on OTT Platform and Determining Similarity and Likeliness Among Users for Recommendation of Movies	537
Mohd Saquib, Aqeel Khalique, and Imran Hussain	
Plant Leaf Disease Identification and Prescription Suggestion Using Deep Learning	547
P. Y. V. N. Dileep Kumar, Purnima Singh, Sagar Pande, and Aditya Khamparia	
Machine Learning Based Data Quality Model for COVID-19 Related Big Data	561
Pranav Vigneshwar Kumar, Ankush Chandrashekhar, and K. Chandrasekaran	
Paradigm of Handling Data Linked to Cloud Database Impacting Cloud Computing: A Case Study Based on Simulation	573
Zdzislaw Polkowski and Sambit Kumar Mishra	
Provision and Allocation of Large Scaled Data in Virtual Environment: A Case Study with Simulation Approach	585
Sambit Kumar Mishra and Zdzislaw Polkowski	
Implementation of Secure Communication Framework for Wireless Sensor Network	595
Pankaj Kumar Sharma and U. S. Modani	

Contents	xxi
EMBRACE: Electronic Medical Record Safety, Blockchain to the Rescue	605
Parth Khandelwal, Rahul Johari, Medha Chugh, and Anmol Goel	
Stress Prediction Using Machine Learning and IoT	615
Vividha, Drishti Agarwal, Paras Gupta, Soham Taneja, Preeti Nagrath, and Bhawna Gupta	
Mitigation of DDoS Attacks Using Honeypot and Firewall	625
V. Harikrishnan, H. S. Sanket, K. S. Sahazeer, Siddarth Vinay, and Prasad B. Honnavalli	
Predicting Student's Performance Using Linear Kernel Principal Component Analysis and Recurrent Neural Network (LKPCA-RNN) Model	637
Amita Dhankhar and Kamna Solanki	
Efficient Spectrum Allocation in Wireless Networks Using Channel Aggregation Fragmentation with Reservation Channels	647
N. Suganthi and K. Suresh Kumar	
Toxic Comment Classification Using Bi-directional GRUs and CNN	665
Ritambhra Vatsya, Shreyasi Ghose, Nishi Singh, and Anchal Garg	
VGG-16-Based Framework for Identification of Facemask Using Video Forensics	673
Sunpreet Kaur Nanda, Deepika Ghai, and Sagar Pande	
BlockSIoT: A Blockchain-Based Secure Data Sharing in SIoT	687
J. Chandra Priya, R. N. Karthika, K. Suresh Kumar, and P. Valarmathie	
Hybrid Feature Selection Method for Binary and Multi-class High Dimension Data	701
Ravi Prakash Varshney and Dilip Kumar Sharma	
An Innovative Approach to Establish, Maintain and Review Quality Standards in Higher Education through Quality Assurance Tool	713
Sangeeta Arora and Anil Ahlawat	
Assessment of 3-Dimensional Hand Pose by PosePrior Network for Images	721
Pallavi Malavath, Nagaraju Devarakonda, Zdzislaw Polkowski, and Challapalli Jhansi rani	
Climate Dependent Crop Management Through Data Modeling	739
Narinder Kaur and Vishal Gupta	

Translate2Classify: Machine Translation for E-Commerce Product Categorization in Comparison with Machine Learning & Deep Learning Classification	769
Priyanshi Gupta and Shatakshi Raman	
Fake Feedback Detection to Enhance Trust in Cloud Using Supervised Machine Learning Techniques	789
Harsh Taneja and Supreet Kaur	
Face Recognition Using Artificially Intelligent Methodologies on FERET and FEI Datasets	797
Nilay Pant, Devanshu Rathee, and Rahul Gupta	
Early Prognosis of Acute Myocardial Infarction Using Machine Learning Techniques	815
Abhisht Joshi, Harsh Gunwant, Moolchand Sharma, and Vikas Chaudhary	
Multimodal Biometric Authentication by Slap Swarm-Based Score Level Fusion	831
G. Elavarasi and M. Vanitha	
Hybrid Metaheuristic Algorithm-Based Clustering with Multi-Hop Routing Protocol for Wireless Sensor Networks	843
S. Jagadeesh and I. Muthulakshmi	
Author Index	857

About the Editors

Dr. Deepak Gupta received a B.Tech. degree in 2006 from the Guru Gobind Singh Indraprastha University, India. He received M.E. degree in 2010 from Delhi Technological University, India and Ph. D. degree in 2017 from Dr. A. P. J. Abdul Kalam Technical University, India. He has completed his Post-Doc from Inatel, Brazil. With 13 years of rich expertise in teaching and two years in the industry; he focuses on rational and practical learning. He has contributed massive literature in the fields of Intelligent Data Analysis, BioMedical Engineering, Artificial Intelligence, and Soft Computing. He has served as Editor-in-Chief, Guest Editor, Associate Editor in SCI and various other reputed journals (IEEE, Elsevier, Springer, and Wiley). He has actively been an organizing end of various reputed International conferences. He has authored/edited 43 books with National/International level publishers (IEEE, Elsevier, Springer, Wiley, Katson). He has published 162 scientific research publications in reputed International Journals and Conferences including 83 SCI Indexed Journals of IEEE, Elsevier, Springer, Wiley and many more.

Dr. Zdzislaw Polkowski is an Adjunct Professor at Faculty of Technical Sciences at the Jan Wyzykowski University, Poland. He is also the Rector's Representative for International Cooperation and Erasmus Programme and former dean of the Technical Sciences Faculty during the period of 2009–2012 His area of research includes Management Information Systems, Business informatics, IT in business and administration, IT security, Small Medium Enterprises, CC, IoT, Big Data, Business Intelligence, and Block chain. He has published around 60 research articles. He has served the research community in the capacity of author, professor, reviewer, keynote speaker, and co-editor. He has attended several international conferences in the various parts of the world. He is also playing the role principal investigator.

Dr. Ashish Khanna has expertise in Teaching, Entrepreneurship, and Research & Development of 16 years. He received his Ph.D. degree from National Institute of Technology, Kurukshetra in March 2017. He has completed his M.Tech. and B.Tech. from GGSIPU, Delhi. He has completed his PDF from Internet of Things Lab at Inatel, Brazil. He has around 100 research papers along with book

chapters including more than 40 papers in SCI indexed journals with cumulative impact factor of above 100 to his credit. Additionally, He has authored, edited and editing 19 books. Furthermore, he has served the research field as a Keynote Speaker/Session Chair/Reviewer/TPC member/Guest Editor and many more positions in various conferences and journals. His research interest includes image processing, Distributed Systems and its variants, and Machine learning. He is currently working at the CSE, Maharaja Agrasen Institute of Technology, Delhi. He is convener and organizer of ICICC Springer conference series.

Dr. Siddhartha Bhattacharyya (FRSA, FIET (UK), FIEI, FIETE, LFOSI, SMIEEE, SMACM, SMIETI, LMCSI, LMISTE) is currently the Principal of Rajnagar Mahavidyalaya, Birbhum, India. Prior to this, he was a Professor at CHRIST (Deemed to be University), Bangalore, India. He also served as the Principal of RCC Institute of Information Technology, Kolkata, India. He has served VSB Technical University of Ostrava, Czech Republic as a Senior Research Scientist. He is the recipient of several coveted national and international awards. He received the Honorary Doctorate Award (D.Litt.) from The University of South America and the SEARCC International Digital Award ICT Educator of the Year in 2017. He was appointed as the ACM Distinguished Speaker for the tenure 2018–2020. He has been appointed as the IEEE Computer Society Distinguished Visitor for the tenure 2021–2023. He is a co-author of six books and the co-editor of 75 books and has more than 300 research publications in international journals and conference proceedings to his credit.

Oscar Castillo holds the Doctor in Science degree (Doctor Habilitatus) in Computer Science from the Polish Academy of Sciences (with the Dissertation “Soft Computing and Fractal Theory for Intelligent Manufacturing”). He is a Professor of Computer Science in the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico. Currently, he is President of HAFSA (Hispanic American Fuzzy Systems Association) and Past President of IFSA (International Fuzzy Systems Association). Professor Castillo is also Chair of the Mexican Chapter of the Computational Intelligence Society (IEEE). His research interests are in Type-2 Fuzzy Logic, Fuzzy Control, Neuro-Fuzzy and Genetic-Fuzzy hybrid approaches. He has published over 300 journal papers, 10 authored books, 40 edited books, 200 papers in conference proceedings, and more than 300 chapters in edited books, in total 865 publications according to Scopus (H index = 60), and more than 1000 publications according to Research Gate (H index = 72 in Google Scholar).

CNN Based Feature Extraction for Visual Speech Recognition in Malayalam



Shabina Bhaskar and T. M. Thasleema

Abstract Visual speech recognition is the technique of recognizing speech by using visual cues obtained during speech. The current research in this area makes use of visual cues such as mouth or lip movement for the recognition of speech. This paper introduced a method of visual speech recognition from the face by giving importance to the facial expressions while a person speaking. Facial expression is an important feature for hearing impaired speech recognition because they are more expressive while speaking. As an initial study in this area, we have introduced a Malayalam audio-visual speech emotion database of hearing people. The experimental studies in this database prove that facial expressions play a pivotal role in visual speech recognition.

Keywords Audio visual speech recognition · Convolutional neural network · Hearing impaired · Visual speech recognition

1 Introduction

In communication, both audio and visual signals play significant roles, and this technique is applied in audio-visual speech recognition tasks. By using both signals, one can provide complementary information to the other, so that the recognition rate can be improved. In the case of hearing-impaired (HI) communication, it is entirely different. They communicate through sign language, but it is difficult to understand for others who do not know sign language. Another method they used for the understanding of speech is by observing lip movement. Vast research is carried out in sign language recognition and lipreading, but speech recognition studies for the hearing impaired are rarely available. Also, in their speech, the audio content is very limited, so for their speech recognition we have to focus more on visual signal-based studies. The current studies in Visual Speech Recognition (VSR) are more

S. Bhaskar (✉) · T. M. Thasleema

Department of Computer Science, Central University Kerala, Periyar, Kerala, India
e-mail: thasleema@cukerala.ac.in

pointed toward the lipreading-based studies, but we can also consider features from eyes, eyebrows, nose, etc. This paper presented a method of recognizing speech by utilizing information from all facial parts.

At present, the research in VSR is carried out by selecting the mouth area or lip area as a region of interest. The face parts such as eyes, nose, eyebrows, etc., are given less importance. The currently available lipreading feature extraction methods are divided into mainly two categories. The first one is traditional manual feature extraction methods, and the second is feature extraction based on deep learning methods. Face detection and lip localization are important steps in the traditional manual feature extraction method. Then, the lip region is extracted, and after that, the feature extraction algorithms apply to this region. Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), and Gabor features are the commonly used algorithms in visual feature extraction. Finally, appropriate classifiers such as Support Vector Machine (SVM) and Hidden Markov Model (HMM) are used for classification. In deep learning methods, automatically more features are extracted from videos or images, unlike traditional feature extraction methods. The classification is carried out based on the scores of each category through the deep model. The network has its error correction and backpropagation mechanism to adjust the network model parameters according to the labels of the training data. This helps to improve the final classification result of the system.

Automatic lipreading works started in the year 1984 to improve the recognition rate of Automatic Speech Recognition (ASR) systems [1]. In 1997, Goldschens et al. [2] extracted motion features such as optical flow for lipreading. In 2002, [3] used HMM to model features and achieved good recognition results. In 2007, [4] the spatiotemporal Local Binary Pattern (LBP) feature extraction method proposed in OULU database for the recognition of isolated phrases. Now the researchers are more focused on deep learning-based studies instead of traditional feature extraction and classification methods. The learning ability of deep learning models produces good and powerful features according to the problem objectives. These features often give good performance results for different scenarios. In 2011, the deep learning-based studies introduced in audio-visual speech recognition such as the authors Ngiam et al. [5] proposed depth autoencoder and Lee et al. proposed Restricted Boltzmann Machines (RBM) [6]. The deep learning-based visual feature extraction methods for multimodal speech recognition are first time introduced in these papers. In 2014, [7] Noda et al. CNN based feature extraction method for lip image was experimented, and it has got high accuracy in isolated word recognition. The author compared the results with the traditional method, but CNN outperformed all of them. In 2016, Long Short-Term Memory (LSTM) applied on GRID database for lipreading problem achieved an accuracy of 79.6% [8]. The first large-scale lipreading database in English was founded by Chung and Zisserman [9] in 2016. This database was recorded in natural conditions according to the BBC program. The spatial-temporal Convolution Network and Recurrent Neural Network for LipNet proposed in the year 2017 by Assael et al. [10]. They used connectionist temporal classification (CTC) as a network

loss function in the LipNet network. The biggest lipreading database in Chinese was published in 2019 [11] which was real-time data of China CCTV programs.

This paper presents deep learning-based feature extraction and classification model for VSR. The paper is arranged in the following manner. Introduction and the related studies are given in Sect. 1. Section 2 described about the proposed model, and Sect. 3 give description about the database. The result analysis is made in Sect. 4, and 5 deals with conclusion.

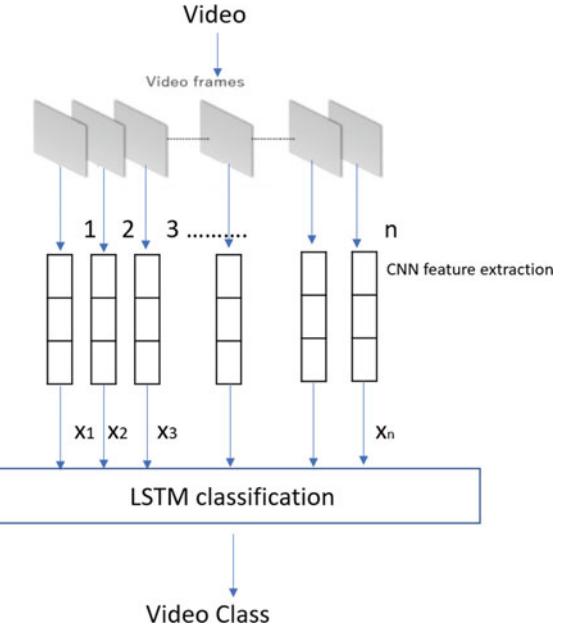
2 Proposed Model

Several two-dimensional related studies have been proposed for VSR, and in most of them, time-related information is not considered. Most of the studies given importance to lip shape or appearance information so lip or mouth detection and localization performed first. In this paper, the key feature is to identify the changes that are happening in the face during speaking. For that, we considered all frames from the video so that to include the changes in the speaker's face concerning time. The paper presents CNN based feature extraction method from video and LSTM used for the classification. The proposed method includes mainly the phases such as video feature extraction and classification. First, the video is edited using Adobe Premiere Pro software to normalize the duration of each video. In this work, no preprocessing stage like face detection and mouth localization is not done. The complete architecture is given Fig. 1.

2.1 Video Feature Extraction

The CNN based video feature extraction method is applied directly to the raw video [12]. The videos are converted into image sequences (frames), and the number of frames varies from 30 to 90 frames for each video. To extract features from these image sequences a pretrained CNN model, GoogLeNet is used. At first, the image sequence passed through the sequence input and sequence folding layer after that the image sequences are passed through the convolutional layer. Then, it passed through sequence unfolding and flatten layer. We will obtain the final feature vector from GoogLeNet last pooling layer ("pool5-7x7_s1"). The feature vector is represented as $m \times n$ array where m is the number of features and n is number of frames corresponding to each video. Here, CNN extracts 1024 features from each frame, and the feature set corresponding to each video is represented as $1024 \times \text{number of frames}$. The network architecture is given in Fig. 2.

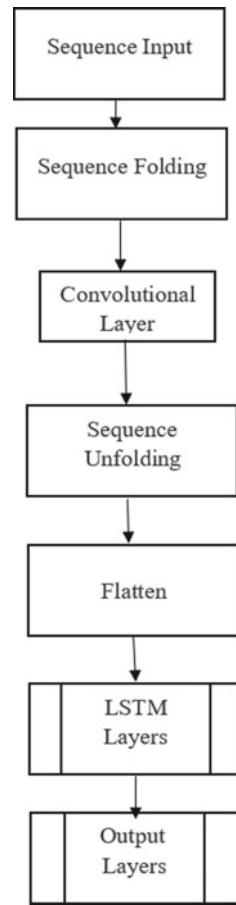
Fig. 1 Proposed architecture for VSR



2.2 Classification

The LSTM is a bidirectional sequence-one classification model. After the feature extraction stage, the sequence feature corresponding to each image passed through this model for analyzing the changes in each frame concerning the change in time. The forward and backward analysis helps to include the details of previous and upcoming frames with the current frame. After that, the results are passed through softmax function and the final classification layer. The complete architecture is given in Fig. 2.

The image sequences are first go through the sequence input layer, then it passed through bidirectional LSTM layer with 200 hidden units. Next, the output passed to the dropout layer, and the output is transformed into a size suitable for classification. This transformation is done by fully connected layer and after that the output passed to final classification layer. Here, we have used Adam optimizer which is used as the default optimization method used for deep learning-based image classification [13, 14]. Adam optimizer got the benefits of both commonly used optimization algorithms such as adaptive gradient algorithm and root mean square propagation in deep learning. It improves the performance for the problem such as image and video classification and also performs well in the case of noisy data. The CNN sequence feature vector trained using LSTM network with 30 epochs by selecting batch size as 16. From the dataset, 80% is used for training and 20% used for testing with an adaptive moment estimation (adam) optimizer. The initial learning rate of 0.0001 with exponential decay rate 0.9 and 0.99 for the first and second-moment estimates.

Fig. 2 Network architecture

3 Malayalam Audio-Visual Database

Audio-visual database study in Malayalam already started by researchers and the studies are focused on linguistic analysis. We aim to identify how much facial information can be utilized to recognize the speech of HI persons. As a stepping stone to this study, we have developed an audio-visual Malayalam database for an unimpaired person. In this, we have given importance to both facial expression and speech. The data is recorded as a video and to get the real-time effect the audio is not recorded in a pure noise-free environment. We have selected words like fever, cold, headache, wound, breathing problem, allergy, cancer, surgery, cured, healed from health vocabulary, and its Malayalam translation is used for recording. Each word is uttered with expression and in the face also expression is visible. Two speakers participated in the recording one male and one female. A total of 100 samples corresponding to each word recorded from each speaker, so that a total of 2000 videos were recorded from

both speakers. The videos are recorded in mp4 format with a resolution of 1280*720 having a frame rate of 29.9 fps. The audio files are combined at a sampling rate of 44 KHz and are recorded using Rode mic.

4 Result Analysis

Most of the available audio-visual studies are concentrated on speech recognition, and the audio-visual emotional studies are interested in emotion recognition, affect recognition, etc. The research related to both facial expression and speech recognition studies is rarely available. This developed database can be utilized for both expression and speech recognition related works. As an initial study, we performed a speech recognition study from facial data and that result is described in this section.

Before that, we have experimented with the presented method for VSR in another audio-visual database called Oulu VS. This database was specially created for speech recognition studies which include 20 subjects uttering 10 phrases. Many results associated with VSR and audio-visual speech recognition researches are available for Oulu VS database. Most of the studies are lip region-based, but we utilized full facial features for the recognition of studies. We have conducted the speaker-independent VSR experiment Oulu VS database, and the obtained result is given in Table 1. The accuracy rate is very low because the facial expression is almost the same for all utterances, so it is difficult to recognize speech from the face.

Next, we experimented with the same method with our audio-visual speech emotional database, and the results are discussed below. The speaker-dependent and independent experiments are performed on this database. For the speaker-independent experiment, from out of a total of 2000 videos in which 1600 data were taken for training and the remaining for testing. Speaker-dependent experiment is carried out for each speaker separately. For that, 800 samples were taken for training and 200 samples for testing. The system is tested based on accuracy parameters, and Table 2 gives you information about it. The highest accuracy is obtained for speaker-dependent experiment for speaker 1 which is 89%.

Table 1 Result for speaker independent experiment: Oulu VS database

Batch size	Accuracy	Training time
16	53	10 min

Table 2 Results: Malayalam audio-visual database

<i>Speaker dependent</i>			
	Batch size	Accuracy	Training time
Speaker 1	16	89	46 min
Speaker 2	16	88	27 min
<i>Speaker independent</i>			
	16	84.75	72 min

5 Conclusion

In this paper, a new method for Malayalam visual speech recognition from the full face is introduced. The experimental analysis carried out in this database reported that the audio-visual data with facial expression give additional visual information which helps to improve the speech recognition accuracy. Even though wide research is available in the field of facial expression analysis for emotion recognition, affect recognition, sign language recognition, etc., the proposed system is introduced a new method of recognizing speech from facial information.

In the future, we will extend our work to practical applications such as a real-time speech recognition system based on our proposed method. Also, considering this domain, a good AVSR database design for the hearing impaired is another future direction. Furthermore, continuous speech recognition seems to be one of the important future in this domain.

Acknowledgements We thank the Kevees Studio team for the support they have made for the database development.

References

- Petajan ED (1984) Automatic lipreading to enhance speech recognition. In: Proceedings of global telecommunication conference, pp 265–272
- Goldschen AJ, Garcia ON, Petajan ED (1997) Continuous automatic speech recognition by lipreading. In: Shah M, Jain R (eds) Motion-based recognition. Computational imaging and vision, vol 9. Springer, Dordrecht
- Goldschen AJ, Garcia ON, Petajan E (2002) Continuous optical automatic speech recognition by lipreading. In: Proceedings of 28th asilomar conference signals, systems computing, pp 572–577
- Zhao G, Pietikäinen M, Hadid A (2007) Local spatiotemporal descriptors for visual recognition of spoken phrases. In: Proceedings ACM international multimedia conference exhibition, pp 57–66
- Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY (2011) Multimodal deep learning. In: Proceedings of 28th international conference machine learning (ICML), pp 689–696
- Lee H, Ekanadham C, Ng AY (2008) Sparse deep belief net model for visual area V2: In: Proceedings of advance neural information processing system, pp 873–880

7. Noda K, Yamaguchi Y, Nakadai K, Okuno HG, Ogata T (2014) Lipreading using convolutional neural network. In: Proceedings of conference international speech communication Association, pp 1149–1153
8. Wand M, Koutnik J, Schmidhuber J (2016) Lipreading with long short-term memory. In: Proceedings of IEEE international conference acoustics, speech signal process. (ICASSP), pp 6115–6119
9. Chung JS, Zisserman A (2016) Lip reading in the wild. In: Proceedings of Asian conference computing visual. Springer, Cham, Switzerland, pp 87–103
10. Assael YM, Shillingford B, Whiteson S, de Freitas N (2016) LipNet: end-to-end sentence-level lipreading. Available: <http://arxiv.org/abs/1611.01599>
11. Yang S, Zhang Y, Feng D, Yang M, Wang C, Xiao J, Long K, Shan S, Chen X (2019) LRW-1000: a naturally-distributed large-scale benchmark for lip reading in the wild. In: Proceedings of 14th IEEE international confernce automation FaceGesture recognition (FG), pp 1–8
12. Vakhshiteh F, Almasganj F (2019) Exploration of properly combined audiovisual representation with the entropy measure in audiovisual speech recognition. Circ Syst Sig Process 38:2523–2543
13. Fabelo H et al (2019) In-vivo hyperspectral human brain image database for brain cancer detection. IEEE Access 7:39098–39116. <https://doi.org/10.1109/ACCESS.2019.2904788>
14. Wong SC, Stamatescu V, Gatt A, Kearney D, Lee I, McDonnell MD (2017) Track everything: limiting prior knowledge in online multi-object recognition. In: IEEE transactions on image processing, vol 26(10), pp 4669–4683. <https://doi.org/10.1109/TIP.2017.2696744>

Effective Rate of Minority Class Over-Sampling for Maximizing the Imbalanced Dataset Model Performance



Forhad An Naim, Ummae Hamida Hannan, and Md. Humayun Kabir

Abstract Over-sampling is a resampling technique that has been designed to balance the imbalanced class distribution by duplicating samples of the minority class for a classification dataset. It is challenging to determine what rate of sample duplicating will be effective to maximize the model accuracy. In this research, we have proposed a method to determine an effective rate of the minority class over-sampling by which to maximize the performance of the machine learning model. We have used five over-sampling methods named Random over-sampling, SMOTE, SVMSMOTE, SMOTE Nominal, and Borderline SMOTE to evaluate the proposed method with five publicly available datasets. During the training period, we have over-sampled the minority class based on the majority class samples between the percentage ranges from 0 to 50%. Random Forest (RF) has been used as a machine learning classifier because its default hyperparameters already return great results. F1-score has been used as evaluation matrices because it is effective for imbalanced datasets. It has been seen that the proposed model has achieved a top f1-score when the minority class was over-sampled by 30–45% of the majority class samples.

Keywords Resampling · Over-sampling · Over-sampling rate · Imbalanced class distribution · Minority class · Machine learning over-sampling · SMOTE

F. A. Naim (✉)

Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh

U. H. Hannan

Department of Computer Science, American International University Bangladesh, Dhaka, Bangladesh

Md. Humayun Kabir

Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram, Bangladesh

1 Introduction

The fourth industrial revolution has already emerged which is known as Industry 4.0. Industry 4.0 is automating traditional manufacturing practice using smart technology with the help of information and communication technology based on the components Internet of Things (IoT), Cloud Computing, Big data, BlockChain, etc. As a result, a huge volume of data is generating exponentially concerning time. Forbes says the reason behind the affluence of many international organizations is the proper management of big data by machine learning [1]. According to Deloitte Global, data is the future gold [2]. Data is the heart of machine learning systems. Machine learning can accelerate this process of maximizing the potential by uncovering trends and patterns with the help of decision-making algorithms.

In machine learning, imbalanced class distribution act like a beetle in the apple. Imbalanced class distribution is an emerging challenge in the machine learning area. Handling imbalanced class distribution can improve the model's predictive accuracy [3]. Generally, it happens when observations of one class are much higher or lower than others. An example, the distribution of class A is 97% but class B is 3%. Here, class A is the majority class and class B is the minority class. The instinct of machine learning is improving the accuracy and reducing the error that is not concerned about class distribution. As a result, in most cases, machine learning algorithms consider minority classes as noise. So, it can be lead to the worst machine learning performance matrices. Resampling methods are usually used to address class distribution problems. Traditionally, Resampling techniques are divided into two categories such as i) Under-sampling the majority class and ii) Over-sampling the minority class. Under-sampling refers to a group of techniques that have been designed to balance the class distribution by deleting or eliminating the samples from the majority class for a classification dataset [4]. Over-sampling refers to a group of techniques that have been designed to balance the class distribution by duplicating samples of the minority class for a classification dataset.

But, the major challenge is what rate of the sample or how many minority samples duplicating will be effective for maximizing model accuracy. Researchers try different rates or units of over-sampling to achieve the effective point by which to maximize the performance of the machine learning model. The process is very time-consuming, challenging, and requires heavy processing power simulation systems. This research aims to determine an effective rate of the minority class over-sampling by which to maximize the performance of the machine learning model. We have used five over-sampling methods named Random over-sampling, SMOTE, SVMSMOTE, SMOTE Nominal, and Borderline SMOTE with five publicly available datasets. During the training period, we have over-sampled the minority class based on the majority class samples between the percentage ranges from 0 to 50%. Random Forest (RF) has been used as a machine learning classifier. F1-score is used as a performance measure unit as the dataset is imbalanced.

2 Related Works

Several studies show that over-sampling techniques successfully balanced the imbalanced class distribution that leads better predictive machine learning model. Mohammed et al. [5] researched resampling methods and found that over-sampling performs better than under-sampling for different classifiers and obtains higher accuracy. Wang et al. [6] proposed a new over-sampling method named AGNES-SMOTE (Agglomerative Nesting-Synthetic Minority Over-sampling Technique) based on hierarchical clustering and improved SMOTE. Experimental results of AGNES-SMOTE indicated it had improved SVM classification performance on imbalanced datasets. Ren et al. [7] proposed a fuzzy representativeness difference-based over-sampling technique. The fuzzy representativeness was using affinity propagation and the chromosome theory of inheritance (FRDOAC). Experimental results showed better performance than other advanced imbalanced classification algorithms on 16 benchmark datasets. Jiang et al. [8] proposed a new over-sampling method named OS-CCD based on the classification contribution degree. OS-CCD follows the spatial distribution characteristics of original samples on the class boundary, as well as avoids over-sampling from noisy points. Bej et al. [9] proposed an over-sampling method named Localized Random Affine Shadow sampling (LoRAS). The study claimed the proposed approach overcame the limitation of SMOTE. LoRAS improved the F1-score than SMOTE for 14 publicly available imbalanced datasets.

3 Datasets

Imbalanced class distribution is an emerging challenge in the machine learning area. An Imbalanced dataset could lead to a low predictive model for the minority class. Handling imbalanced class distribution can improve the model's predictive accuracy [10]. This factor encouraged us to build the dataset for analyzing the effective rate of the over-sampling technique for maximizing machine learning performance. We have collected five publicly available highly imbalanced datasets from www.kaggle.com. Those are credit card fraud dataset, multiple insurance claim dataset, insurance lead generation dataset, employee schedule status, and bank marketing. The summary of the dataset is shown in Table 1. The datasets can be found in this link <https://github.com/Mithun1990/Resample-Dataset/>.

4 Feature Engineering

Feature engineering is a powerful technique that is used to improve the performance of machine learning algorithms. Feature engineering determines the good features from among the dataset features. To develop a good-quality model, researchers need

Table 1 Summary of the five datasets

Dataset	Targets	Datasets	Targets
Schedule status	Attended (1) → 9416 (51%) Personal (0) → 8744 (47.2%) Missed (2) → 342 (1.8%)	Credit card fraud	Yes (1) → 422 (1.7%) No (0) → 24,712 (98.3%)
Insurance claim	Yes (1) → 18,223 (95.5%) No (0) → 861 (4.5%)	Insurance lead generation	Lead (1) → 1492 (9%) No (0) → 25,106 (91%)
Bank marketing	Yes → 5298 (90%) No → 620 (10%)		

to consider many features like productivity, cost, result, and time. Because of too many different features that may be required while developing a model, it becomes hectic for researchers [11]. In machine learning, good features play a crucial role which has a higher predictive power [12]. Attribute reduction (AR) related to features selectors is the production of a minimal number of reductions representing the reliable meaning of all features. Feature engineering finds the features which are known as the most informative ones and the least possible features with least possible data loss [13]. Different feature engineering techniques such as elimination of correlated features, feature selection have been applied to select the features correlated to targets. The correlated feature is a big problem in machine learning. Correlated features are those values that have a mutual relationship with one another in linear space. Correlation measures the linear dependency, similarity, and association between features. Correlated features are only redundant data that increase the model complexity. If the correlation coefficient of two features is +1 then features are highly linearly dependent or correlated. If the features are unrelated then the correlated coefficient will be 0 [14]. However, from the proposed datasets, we have excluded the correlated features based on a correlated coefficient threshold value of 0.65.

5 Over-Sampling Approaches

Over-sampling increases the minority class data points randomly by replicating them to balance training data. In other words, over-sampling refers to a group of techniques that have been designed to balance the class distribution by duplicating samples of the minority class for a classification dataset. In this research, we have used five over-sampling approaches (i) Random under-sampling (ii) SMOTE (Synthetic Minority Over-sampling Technique) (iii) Support Vector SMOTE (SVMSMOTE) (iv) SMOTE

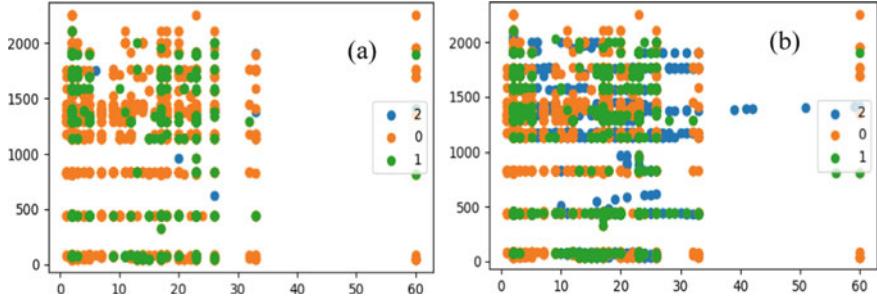


Fig. 1 Over-sampling example of schedule status by SMOTE **a** before over-sampling; **b** after over-sampling

for nominal (SMOTEN), and (v) Borderline SMOTE for evaluating the proposed method. Figure 1 shows the impact of SMOTE approach on the dataset.

Random over-sampling increases the minority class samples randomly by replicating them to balance training data. In SMOTE [15], the minority classes are increased to balance training data by generating new synthetic data. The fundamental concept is that generating new synthetic data points between each minority classes. SMOTE randomly picks up the minority class and calculates the k-nearest neighbor from that particular point. Eventually, the new generated synthetic data add placed between the neighbors and the chosen minority point. In SVMSMOTE [16], a support vector machine (SVM) is trained to predict targets based on focusing only on the minority class instances residing along the decision boundary. Because the region of the minority class instances has crucial involvement for establishing the decision boundary. Then, the minority instances are resampling is done by expanding fewer majority class instances with the regions of the minority class by extrapolation. Otherwise, the current boundary of the minority class would be consolidated by interpolation. In SMOTEN, with the help of the modified Value Difference Metric proposed by Cost and Salzberg [17], nearest neighbors are computed. The Value Difference Metric (VDM) looks at the overlap of feature values over all of the feature vectors. The borderline SMOTE [18] has been proposed based on the Synthetic minority over-sampling technique (SMOTE). Unlike SMOTE method, borderline SMOTE is the only oversample or strengthen the borderline minority examples. Firstly, find out the borderline examples then synthetic examples are generated from them and added to the original training set.

6 Proposed Method

An imbalanced dataset is a major challenge in machine learning. Most of the traditional machine learning classifiers or methods ignore the minority classes during the training period as noise. That can lead to constructing a poor machine learning model

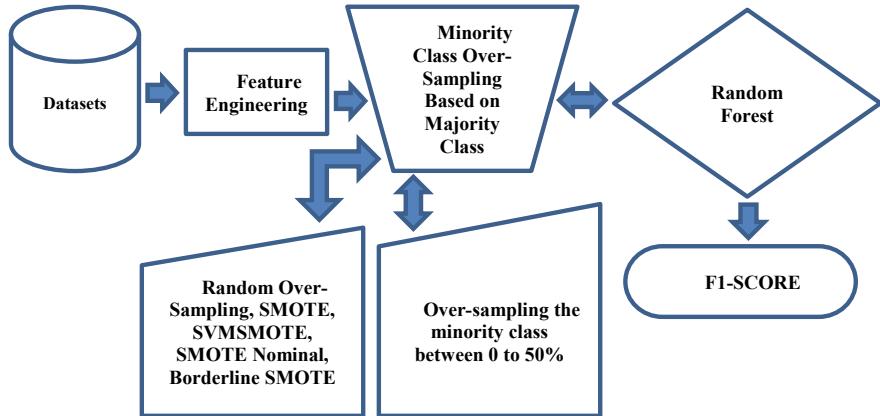


Fig. 2 Proposed model

for predicting infrequent classes. Several over-sampling methods have already been used widely to handle imbalanced datasets. In this research, we have proposed a method to determine an effective rate of the minority class over-sampling by which to maximize the performance of the machine learning model. We have used five over-sampling methods named Random over-sampling, SMOTE, SVMSMOTE, SMOTE Nominal, and Borderline SMOTE to evaluate the proposed method. During the training period for all five over-sampling methods, we have applied over-sampling to the minority class based on the percentage of the majority class samples. We have over-sampled the minority class based on the majority class samples percentage between the ranges from 0 to 50%. In each step, we have increased the percentage by 10% before achieved 20% over-sampling. After achieving 20% over-sampling then we have increased the percentage by 5%. The reason behind that during the training period we have seen that before 20% over-sampling increasing over-sampling rate by 5% the change of accuracy is negligible. But, after 20% over-sampling, each change in over-sampling rate by 5% could be effective. After that, we have calculated the f1-score and compare it with the result of the other step. F1-score is more effective than other machine learning evaluation matrices when the dataset is imbalanced [19]. We have used Random Forest (RF) as a classifier because its default hyperparameters already return great results and the system is great at avoiding overfitting and it works well for categorical and continuous values [20]. The proposed model is shown in Fig. 2.

7 Experimental Result and Analysis

7.1 Experimental Environment

Experimental environment is an important factor in the field of machine learning when dealing with multiple datasets, heavy processing over-sampling techniques, and machine learning classifiers. In this research, we have used Intel core-i5 processor, 16 GB ram, and 2 GB Graphics card. We have used python as programming language.

7.2 Train and Test Dataset

We have split the dataset into two parts as a training dataset and a test dataset. We have used 80% of the dataset as a training dataset and 20% of the dataset as a test dataset.

7.3 Results and Analysis

Performance measurement is a must in machine learning for evaluating how one classifier is better than other classifiers. Different evaluation matrices are widely used to evaluate model performance. In this research, we have presented the f1-score as a performance measurement for classification problems because F1-score is more effective than other machine learning evaluation matrices when the dataset is imbalanced. Evaluation matrices are measured based on four parameters such as True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). The percentage of positive identifications correctly predicted by classification methods for a particular class is called precision. The percentage of actual identifications correctly predicted by the classification methods for a particular class is called recall. F1-score makes equilibrium between precision (P) and recall (R). It is the harmonic mean of precision and recall.

Table 2 shows the f1-score of the insurance claim dataset for different minority over-sampling percentages between 0 and 50%. It can be seen that the four over-sampling methods such as SMOTE, SVMSMOTE, SMOTEN, Borderline SMOTE have achieved top f1-score when 30% over-sampling is applied to the minority class. Random over-sampling has achieved the top f1-score when 35% over-sampling is applied to the minority class.

Table 3 shows the f1-score of the schedule status dataset for different minority over-sampling percentages between 0 and 50%. It can be seen that the four over-sampling methods such as Random over-sampling and Borderline SMOTE have achieved top f1-score when 40% over-sampling is applied to the minority class.

Table 2 F1-score of the insurance claim dataset for five over-sampling methods

Percentage	Random over-sampling	SMOTE	SVMSMOTE	SMOTEN	Borderline SMOTE
0	0.89	0.89	0.89	0.89	0.89
10	0.89	0.86	0.89	0.88	0.86
20	0.90	0.92	0.90	0.91	0.90
25	0.90	0.90	0.91	0.91	0.88
30	0.91	0.94	0.92	0.94	0.92
35	0.94	0.90	0.90	0.87	0.88
40	0.90	0.90	0.90	0.87	0.88
45	0.90	0.89	0.91	0.87	0.87
50	0.89	0.90	0.90	0.87	0.88

Table 3 F1-score of the schedule status dataset for five over-sampling methods

Percentage	Random over-sampling	SMOTE	SVMSMOTE	SMOTEN	Borderline SMOTE
0	0.54	0.54	0.54	0.54	0.54
10	0.56	0.57	0.58	0.56	0.57
20	0.58	0.58	0.58	0.56	0.58
25	0.57	0.58	0.58	0.58	0.58
30	0.57	0.58	0.58	0.58	0.58
35	0.57	0.58	0.58	0.58	0.58
40	0.61	0.62	0.58	0.58	0.61
45	0.61	0.62	0.60	0.60	0.61
50	0.57	0.58	0.57	0.56	0.56

SVMSMOTE and SMOTE Nominal have achieved the top f1-score when 45% over-sampling is applied to the minority class.

Table 4 shows the f1-score of the credit card dataset for different minority over-sampling percentage between 0 and 50%. It can be seen that all the five over-sampling methods have achieved top f1-score when 40% over-sampling is applied to the minority class.

Table 5 shows the f1-score of the insurance lead generation dataset for different minority over-sampling percentage between 0 and 50%. It can be seen that all the five over-sampling methods have achieved top f1-score when 40% over-sampling is applied to the minority class.

Table 6 shows the f1-score of the bank marketing dataset for different minority over-sampling percentage between 0 and 50%. It can be seen that the four over-sampling methods except Borderline SMOTE have achieved top f1-score when 40% over-sampling is applied to the minority class.

Table 4 F1-score of the credit card dataset for five over-sampling methods

Percentage	Random over-sampling	SMOTE	SVMSMOTE	SMOTEN	Borderline SMOTE
0	0.51	0.51	0.51	0.51	0.51
10	0.50	0.50	0.50	0.50	0.50
20	0.50	0.51	0.50	0.51	0.50
25	0.50	0.50	0.50	0.50	0.50
30	0.51	0.51	0.51	0.50	0.50
35	0.52	0.50	0.52	0.52	0.51
40	0.54	0.53	0.55	0.55	0.54
45	0.54	0.53	0.55	0.55	0.54
50	0.52	0.52	0.51	0.52	0.52

Table 5 F1-score of the lead generation dataset for five over-sampling methods

Percentage	Random over-sampling	SMOTE	SVMSMOTE	SMOTEN	Borderline SMOTE
0	0.47	0.47	0.47	0.47	0.47
10	0.46	0.46	0.46	0.46	0.46
20	0.47	0.47	0.47	0.47	0.47
25	0.50	0.47	0.50	0.47	0.50
30	0.53	0.47	0.58	0.53	0.51
35	0.58	0.53	0.58	0.53	0.56
40	0.61	0.56	0.59	0.60	0.58
45	0.58	0.52	0.58	0.53	0.56
50	0.53	0.47	0.58	0.53	0.51

Table 6 F1-Score of the Bank Marketing dataset for five over-sampling methods

Percentage	Random over-sampling	SMOTE	SVMSMOTE	SMOTEN	Borderline SMOTE
0	0.47	0.47	0.47	0.47	0.47
10	0.47	0.47	0.47	0.47	0.47
20	0.47	0.47	0.47	0.47	0.47
25	0.47	0.48	0.47	0.47	0.48
30	0.47	0.48	0.47	0.47	0.48
35	0.49	0.5	0.49	0.49	0.54
40	0.55	0.57	0.54	0.53	0.54
45	0.55	0.57	0.54	0.53	0.57
50	0.55	0.57	0.54	0.53	0.57

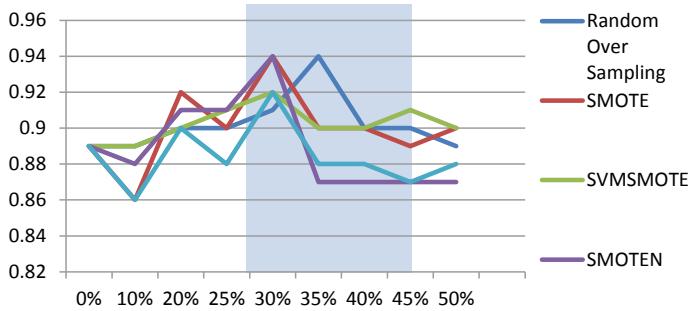


Fig. 3 Accuracy comparison curve of insurance claim dataset

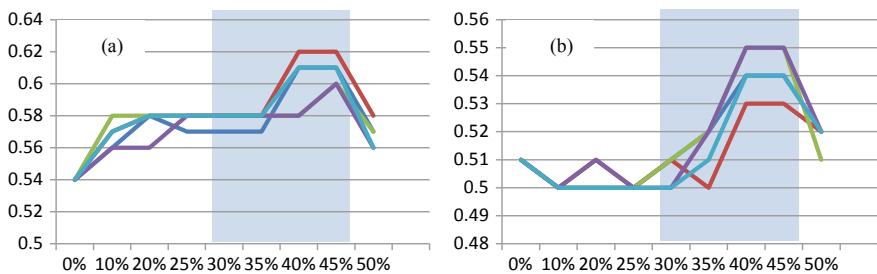


Fig. 4 Accuracy comparison curve of **a** schedule status, **b** credit card dataset

8 Discussions

- Feature engineering determines the good features from among the dataset features. Elimination of correlated features has reduced the model complexity and increased model accuracy. Correlated features are only redundant because correlated features are those values that have a mutual relationship with one another in linear space.
- Over-sampling methods have increased the minority class samples by duplicating the original minority samples as shown in Fig. 1 for SMOTE.
- The F1-score score is more useful as performance evaluation matrices when the dataset is imbalanced. The reason behind that is the F1-score makes a balance between precision and recall.
- F1-score has achieved better accuracy when different over-sampling methods are applied on the dataset than before over-sampling. Tables 2, 3, 4, 5 and 6 shows F1-score score was worst when the over-sampling rate was 0%.
- F1-score has achieved top score when the minority class was over-sampled by 30–45% of the majority class samples for all the five over-sampling methods. Figures 3, 4 and 5 shows the scenario for all the five datasets.

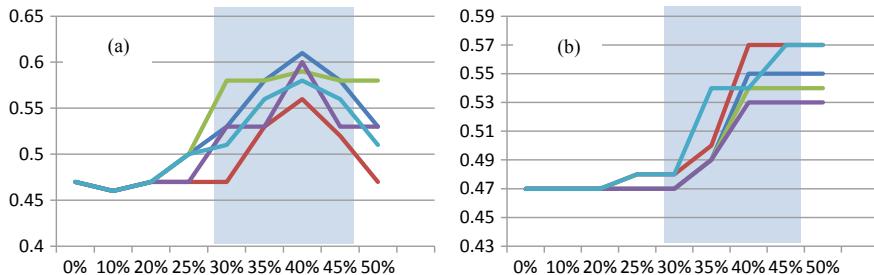


Fig. 5 Accuracy comparison curve of **a** insurance lead generation, **b** bank marketing dataset

9 Conclusion

The paper has described a proposed method to determine an effective rate of the minority class over-sampling by which to maximize the performance of the machine learning model. The proposed method achieved a top f1-score when the minority class was over-sampled by 30–45% of the majority class samples. We have only used five small volume datasets to evaluate the proposed method because the processing of large and multiple datasets is time-consuming and requiring a heavy processing machine. We will use at least ten small and large volume datasets in further research. We will also research under-sampling known as the resampling technique to determine an effective rate range of the minority class under-sampling by which to maximize the performance of the machine learning model.

References

1. How is big data analytics using machine learning? [Online]. Available: <https://www.forbes.com/sites/forbestechcouncil/2020/10/20/how-is-big-data-analytics-using-machine-learning/?sh=26b6e21771d2>. Accessed: 15 Jan 2021
2. Data is the new gold [Online]. Available: <https://www2.deloitte.com/global/en/pages/real-estate/articles/future-real-estate-data-new-gold.html>. Accessed: 15 Jan 2021
3. Ghorbani R, Ghousi R (2020) Comparing different resampling methods in predicting students performance using machine learning techniques. IEEE Access, pp 1–1. <https://doi.org/10.1109/access.2020.2986809>
4. Under-sampling algorithms for imbalanced classification [Online]. Available: <https://machelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/>. Accessed: 17 Jan 2021
5. Mohammed R, Rawashdeh J, Abdullah M (2020) Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th international conference on information and communication systems (ICICS), pp 243–248. <https://doi.org/10.1109/ICICS49469.2020.239556>
6. Wang X, Yang Y, Chen M, Wang Q, Qin Q, Jiang H, Wang H (2020) AGNES-SMOTE: an oversampling algorithm based on hierarchical clustering and improved SMOTE scientific programming. Hindawi. <https://doi.org/10.1155/2020/8837357>

7. Ren R, Yang Y, Sun L (2020) Oversampling technique based on fuzzy representativeness difference for classifying imbalanced data. *Appl Intell* 50:2465–2487. <https://doi.org/10.1007/s10489-020-01644-0>
8. Jiang Z, Pan T, Zhang C, Yang J (2021) A new oversampling method based on the classification contribution degree. *MDPI J* 13(2). <https://doi.org/10.3390/sym13020194>
9. Bej S, Davtyan N, Wolfien M et al (2021) LoRAS: an oversampling approach for imbalanced datasets. *Mach Learn* 110:279–301
10. Arbelaitz O, Gurrutxaga I, Muguerza J, Perez JM (2013) Applying resampling methods for imbalanced datasets to not so imbalanced datasets. Lecture notes in computer science, pp 111–120. https://doi.org/10.1007/978-3-642-40643-0_12
11. Gupta et al (2018) Usability feature extraction using modified crow search algorithm: a novel approach. *Neural Comput Appl* 32:10915–10925. <https://doi.org/10.1007/s00521-018-3688-6>
12. Rawat T, Khemchandani V (2019) Feature engineering (FE) tools and techniques for better classification performance. *Int J Innov Eng Technol (IJIET)*. <https://doi.org/10.21172/ijiet.82.024>
13. Alweshah M, Alzubi J, Alzubi OA (2016) Solving attribute reduction problem using wrapper genetic programming. *IJCSNS Int J Comput Sci Netw Secur* 16(5)
14. Blessie EC, Karthikeyan E (2012) Sigmis: a feature selection algorithm using correlation based method. *J Algorithms Comput Technol* 6
15. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
16. Nguyen HM, Cooper EW, Kamei K (2009) Borderline over-sampling for imbalanced data classification. In: Proceedings of the 5th international workshop on computational intelligence and applications, pp 24–29
17. Cost S, Salzberg S (1993) A weighted nearest neighbor algorithm for learning with symbolic features. *Mach Learn* 10(1):57–78
18. Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Proceedings of the 1st international conference on intelligent computing, pp 878–887
19. Naim FA (2021) Bangla aspect-based sentiment analysis based on corresponding term extraction. In: 2021 international conference on information and communication technology for sustainable development (ICICT4SD), Dhaka, Bangladesh, pp 65–69. <https://doi.org/10.1109/ICICT4SD50815.2021.9396970>
20. Why random forest is my favorite machine learning model [Online]. Available: <https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706/>. Accessed 5 Dec 2021

Malaria Cell Image Classification Using Convolutional Neural Networks (CNNs)



Drishti Agarwal, K. Sashanka, Sajal Madan, Akshay Kumar,
Preeti Nagrath, and Rachna Jain

Abstract This study provides an insight into malaria as a disease. Malaria is a disease caused due to plasmodium parasite. It requires a type of mosquito as its host. Hence, the bite of the mosquito leads to malaria. The impact it carries on the health of people around the world is extremely large and cannot be curtailed or controlled without quick and efficient diagnostics and treatment. Subsequent topics dwell on the constraints of detection of the malaria parasite. These constraints may include problems with the feasibility of certain types of tests, or not having access to a diagnostics center or problems with transportation of necessary infrastructure. We also must understand that traditional prognosis methods are very tedious and hence always have a chance for human error or oversight leading to devastating consequences. The ease or simplification of diagnosis of malaria upon the use of machine learning and deep learning is undeniable; hence, in our project, we aim to create a model, using CNN, which using feature extraction, can predict whether a sample image of a Red Blood Cell provided to the model is parasitized or unhealthy. This model has a primary goal of detecting malaria in Red Blood Cells from blood smears with the least number of losses. This allows for the most minimal number of malaria-infected cell to be mistakenly passed off as healthy cells. There will be a further comparison between the custom CNN model, VGG19 model with no fine-tuning, VGG19 model fine-tuned, and a ResNet50 model. All of these are models which have been pre-trained on a vast number of images previously with a set of weights termed as Imagenet.

Keywords Convolution neural network · Imagenet · Keras · Plasmodium · Red Blood Cell (RBC) · ReLU · ResNet50 · Sigmoid · Softmax · VGG19

D. Agarwal (✉) · K. Sashanka · S. Madan · A. Kumar · P. Nagrath · R. Jain
Department of Computer Science and Information Technology, Bharati Vidyapeeth College of Engineering, New Delhi 110023, India

1 Introduction

Malaria is a disease, caused due to a parasite of the scientific name Plasmodium. It is generally carried and spread through by a female Anopheles [1] mosquito.

Subsequently, when the victim gets bitten by the female Anopheles mosquito, the parasite, namely plasmodium, enters the victim's bloodstream. Finally, this parasite reaches the liver and starts reproducing. This leads to symptoms such as high fever, chills, body aches. It was finally summarized by many global health organizations that the main contributors to total malaria cases were South-East [2] Asian Countries and Sub-Saharan Africa (85%) in 2018. It must also be noted that the total reported cases in 2018 were 228 million, just lesser than that in 2017, namely 231 million, as per the World Malaria Report 2019 released by the World Health Organization (WHO) on December 4, 2019 [3]. A heat map provided shows the death toll due to malaria, in a country-wise manner [4]. This allows us to see that of total malaria cases, Africa and India by themselves had contributed to about 85 percent [5] of deaths. Of the total deaths due to malaria the country-wise distribution in the Sub-Saharan Africa region is as follows, Nigeria had 24%, the Democratic Republic of Congo had 11% cases, the United Republic of Tanzania had 5% of deaths, some countries like Angola, Niger, Mozambique, etc. had 4% deaths each [3]. In 2018, an estimated 228 million cases of malaria occurred worldwide (95% confidence interval [CI]: 206–258 million), compared with 251 million cases in 2010 (95% CI: 231–278 million) and 231 million cases in 2017 (95% CI: 211–259 million) [3]. In this study, we have built a deep CNN [6] model with multiple Conv2D layers. To ensure a more accurate and efficient outcome, prominent features such as size, color, and shape, CNN is the deep learning feature that allows us to achieve a very accurate outcome when using it in the form of prediction modeling as it provides great attribute extraction and categorization. Trained professionals need to examine an extremely large number of slides every year to confirm if there are malaria cells present in the sample or not. Subsequently, these trained professionals may have to count up to 5000 infected cells in a single thin smear [7]. This may lead to a human error with heavy repercussions [8]. Hence, we have created a CNN model to give a precise diagnosis [9]. A model which allows for great image processing (this may include image Resizing, shuffling, rotation, normalization, etc.). A subsequent Machine Learning Algorithm needs to be applied for the accurate determination of infected or healthy cells [10]. Furthermore, we have imported pre-trained models like VGG19 and ResNet50 and subsequently fine-tuned them as per our data and purpose. Models like VGG19 and ResNet50 are Deep Learning models which have been extensively trained on a large-scale database of images with a set of fine-tuned weights for superior performance and accuracy. We then compared the accuracies of all the models for a better understanding of individual efficiencies. Dataset: The dataset for the training [11] and testing [12] of these models was collected from the freely available data at Lister Hill National Centre for Biomedical Communications. Lister Hill National Centre for Biomedical Communications is a subgroup of the National Institute of Health (USA). They have extensively collected and stored images of various types of malaria-infected RBCs

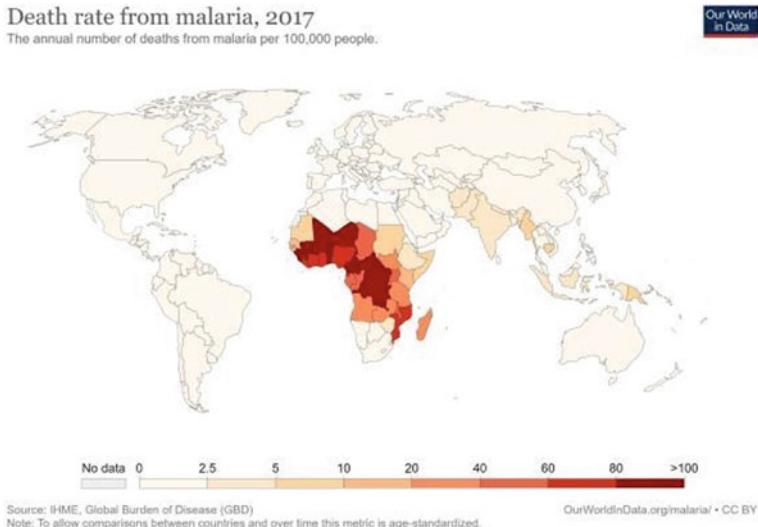


Fig. 1 Heat map of death toll due to malaria, worldwide, 2017 [4]

for educational and research purposes. These images have been collected over a large period. These images have been verified and classified to be either Healthy or Infected. The research is organized into various sections that can be seen below, starting with related works that present a brief idea about the findings and methods employed in this field, followed by the implementation section that presents the working of the model and our solution toward the problem. Further, the result and analysis section explains the outcomes and accuracy of the models along with a significant degree of comparison between them. Finally, the study is concluded under the conclusion section (Figs. 1 and 2).

2 Related Works

There are multiple factors at play when concerning the prognosis of malaria. Some are accuracy, dependability, quick results, and economic feasibility. Thus the most commonly used techniques for the prognosis are thick blood smears (for detection of presence) and thin blood smears (for identifying the type) (Centers for Disease Control and Prevention 2012) However, care must be taken to not cause a loss in accuracy or efficiency, as this would lead to completely wrong or inaccurate prediction [13]. Several studies have been done to analyze the diagnosis and categorization of malaria parasites and nonparasites [13–16]. Das et al. [17] created SVM and Naive Bayes machine learning classifier to create models with which one can get accuracies up to 84% and 83.5% respectively for automatic malaria detection. Ross et al. [18] proposed a three-layer neural network as a classifier for automated malaria

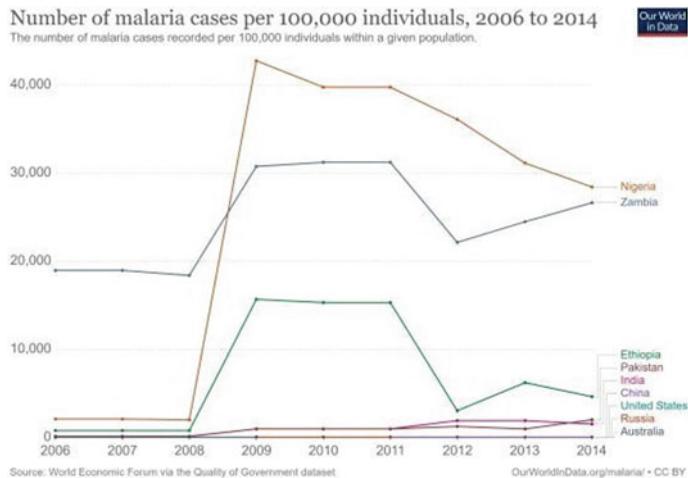


Fig. 2 Number of malaria death nation-wise (2006–14) [4]

diagnosis on thin blood smears with an accuracy of 85%. DenseNet is a version of CNN. It is also known as Densely Connected Convolutional Networks (DenseNet). The use of DenseNet has also been suggested by Huang et al. 2016 [19], where a model was created whose architecture suggested the linking of previous layers to the current layer. The Razavian et al. results (2014) provide CNNs with a very flexible, varied, diverse and huge training database with multiple suitable computer vision features, instead of the older and improved features (Busetouane and Morris 2015). There is a frequent and efficient use by researchers, creators, coders and people from multiple professions of Deep Learning software and applications for more efficient and accurate results [20] (Rajaraman et al. 2017; Suzuki 2017). Another idea was put forth which used optics to “offer quick recognition of malaria-infected red platelets (RBCs) at a lower value exists” [17] by Marcel Akpa Agnero.

3 Implementation

3.1 Data Abstraction

To train our model, we need an accurate training dataset is paramount. Our dataset has been created by NIH, which has cautiously collected all the data over a period of time from multiple sources, such that many possible cases were saved. This data consists of images [21] of Healthy and Plasmodium-infected RBCs also. The images were also downloaded from Lister Hill National Centre for Biomedical Communications Web site where there is a free link to the images of Infected and Healthy Red Blood Cells in the form of JPG format (Figs. 3 and 4).

Fig. 3 Sample images of the infected RBCs in blood

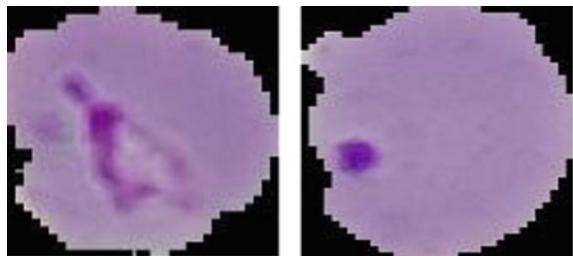
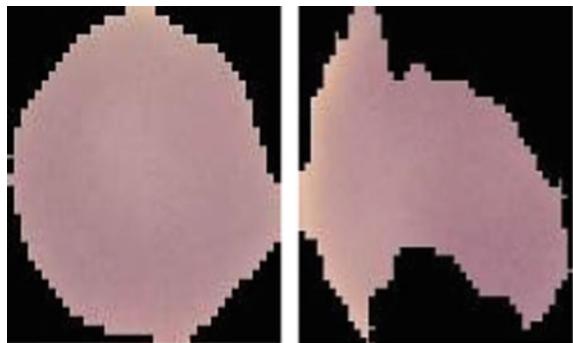


Fig. 4 Uninfected or healthy Red Blood Cells from blood smear image



3.2 Data Processing

To prepare the data for the classification of RBC as Healthy or infected Preprocessing is required. It consists of

- Defined a path to have a training set divided into two folders each containing plasmodium-infected cells and healthy cells respectively, in JPG format.
- Similarly, for the test data and validation data, we divide the training images in the required ratio using train test split from sklearn model selection.
- The images are then resized with the following factors, (width, height, channels). This is equal to (125, 125, 3) respectively.
- Here the total channels taken is 3, as the images are of RGB format.
- Using ImageDataGenerator, we preprocess the images by rotating, shuffling, shearing and flipping by various degrees.
- Subsequent steps include normalization of the images by a factor of 255.

Then the order of Images being fed is randomized. All the labels are then changed to using keras.utils folder to acquire the data in categorical classification. We have chosen some dominating images of the Parasitized and unparasitized blood cell images which helps us in differentiating among the two types of RBC [10]. The default CNN model helps us in detecting the simpler features in initial layers and the complex and intricate features and intricacies in deeper layers. Some observations are shown by the cell images:

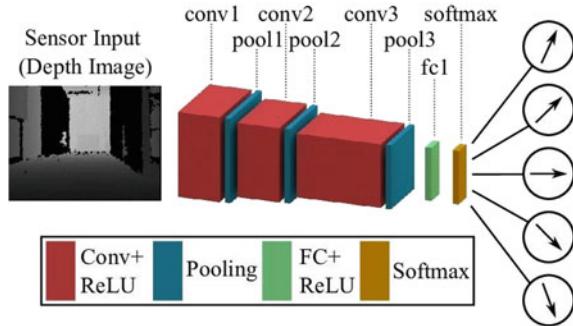


Fig. 5 A typical CNN architecture [22]

1. Loss of normal disk-like shape of the RBC.
2. Increased rigidity of the membrane.
3. Improved porousness to a large variety of ionic and different species.
4. Abrasions and damage to the RBC.
5. Destruction of both contaminated and uninfected red cells since membrane alterations take place.

3.3 Convolutional Neural Network

Convolutional neural network (CNN) is a branch studied underneath the class of deep learning, delivers great scalable characteristics extraction and categorization. In this model, we have created a base CNN deep learning model, without augmentation, as a characteristic extractor for the classification of Parasite and NON-Parasite cells for improved screening. We also will compare the performance to other pre-trained models (Fig. 5).

3.4 VGG19

VGGNet was initially created by the Visual Geometry Group from the University of Oxford. VGGNet is the parent model for other models like VGG19. VGG19 is a pre-trained model consisting of 19 layers. It has been trained in an extremely large number of images and can classify a huge number of images correctly. The VGG19 model uses very reliable weights such as Imagenet weights allowing for the model to perform admirably. In most cases, a fine-tuned VGG19 model may provide better test accuracy than custom CNN models (Fig. 6).

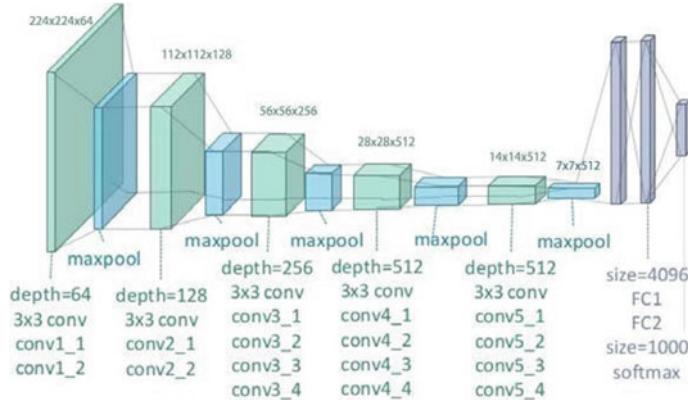


Fig. 6 Sample VGG19 model [23]

3.5 ResNet50

ResNet is a pre-trained convolutional neural network that has been trained in an extremely large number of images and can classify a huge number of images correctly. ResNet also employs the use of ImageNet weights allowing for higher test accuracy. ResNet50 [20] is a convolutional neural network that consists of 50 layers. The presence of many such layers provides a very accurate model but also carries the risk of overfitting. Hence, the implementation of drop layers is imperative to a great model. Various important key functions and constraints applied:

1. **Convolution Layer:** When programming a CNN, the input may be in shape (total images) \times (height in dimension) \times (width in dimension) \times (image depth). Once this image is allowed through the layer of filters. Then a feature map gets created which gives the image the identity and shape of (total images) \times (input height for layer) \times (input width for layer) \times (channels, for RGB = 3). These individual Layers might be called Convolutional Layers. There is however much more to what a Convolutional layer is (Fig. 7).
2. **Strides:** When we need to move a certain number of pixels or convolve a certain number of pixels in the input matrix, we tend to use stride. The amount we provide as an input to stride is the amount of pixels that move or convolve in the input matrix. For example, when the stride is 4, 4 pixels in the input matrix move or gets convolved.
3. **Pooling:** A very large input for the convolutional neural network would require large spatial storage and very high computational power. Both of these requirements are not quite always feasible. Due to this Pooling is used. Pooling reduces the size of the input, layer after layer, and helps ensure that not a large amount of computational power is required.

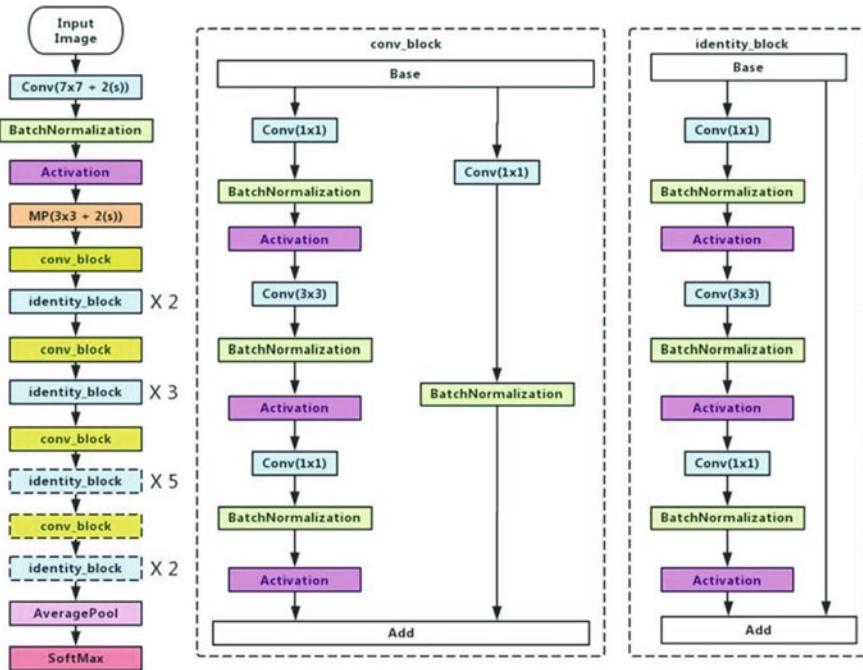
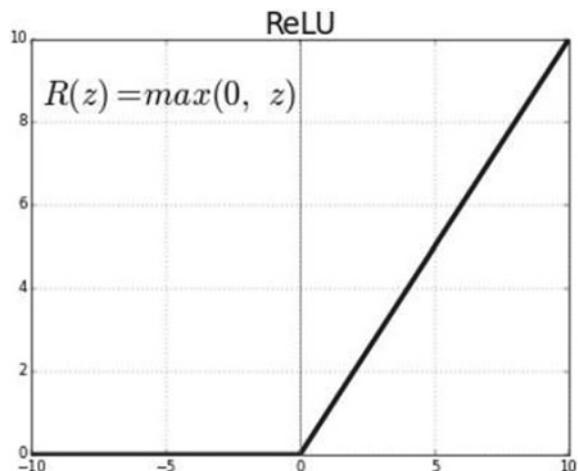


Fig. 7 A sample ResNet50 model [24]

- ReLU Activation function: ReLU also known as rectified linear unit, is an activation functions. ReLU gives the output as the number itself if the number is positive. If number entered is negative, it gives its output as 0 (Fig. 8).

Fig. 8 Sample VGG19 model [23]



5. Softmax Activation Function: This uses a logistic function to give an output in the form of vectors which adhere to a probability distribution whose sum equals 1.
6. Sigmoid Activation Function: Sigmoid activation functions is non-linear function. It is logarithmic. It allows us to stack more layers with better compatibility. It also allows for non-binary activation. In the case of the Sigmoid function, it causes a steep change in the y-axis for a small change in the x-axis. This makes it great for prediction models.
7. The fully connected Layer: The fully connected layer can be said to be a type of forward feed network. The input to the fully connected layer is the flattened output of the last convolution layer.
8. Flatten: The output of the final convolution layer is always in the form of an array. To convert this array into the form of 1-D vectors we use the flatten function (Figs. 9 and 10).

Fig. 9 Graph showing softmax activation function
[25]

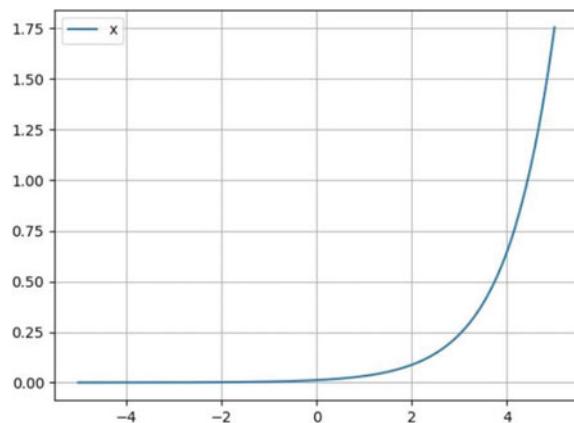
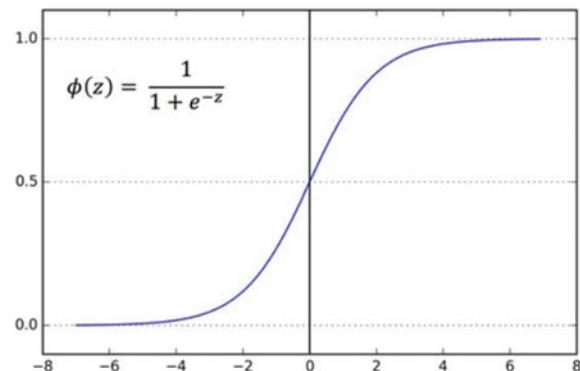


Fig. 10 Graph showing sigmoid activation function
[26]



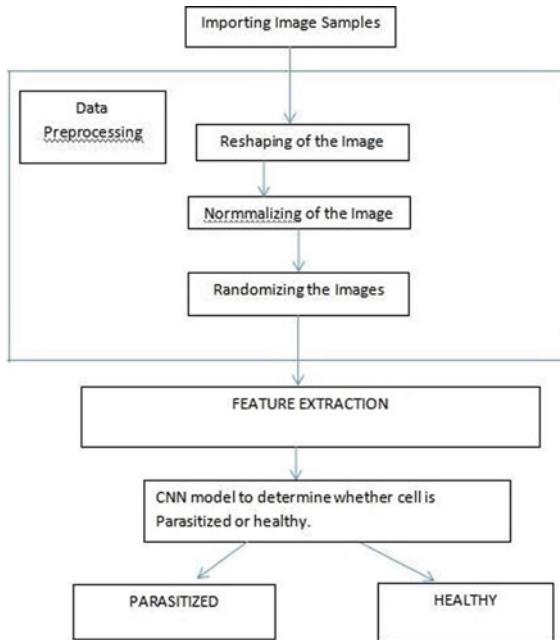


Fig. 11 Process flowchart of implementation

3.6 System Implemented

The flowchart given above may provide a better understanding of the steps taken to create the model (Fig. 11).

4 Result and Analysis

There have been previously many researches and many models have been built to provide the prognosis with great accuracy. However, it needs to be said that the purpose of this model is to enable the prognosis of malaria even in remote far-flung places. This is the reason why we need machine learning models with simpler programs, requiring less computational hardware. They must also not place a large load on the battery. This is why we created a base CNN with no augmentations, but also providing outputs with high accuracy. The Dataset obtained by us consists of 27,558 images for both types of Images, namely plasmodium-infected and healthy well. Out of these 13,779 images, each is Unhealthy and Healthy respectively. This allows us to achieve a high training accuracy. Here, we have put in place a model which predicts the probabilities of the sample being Parasitized and Healthy both, respectively, with 1 being a 100% assurance and 0 referring to a 0% guarantee.

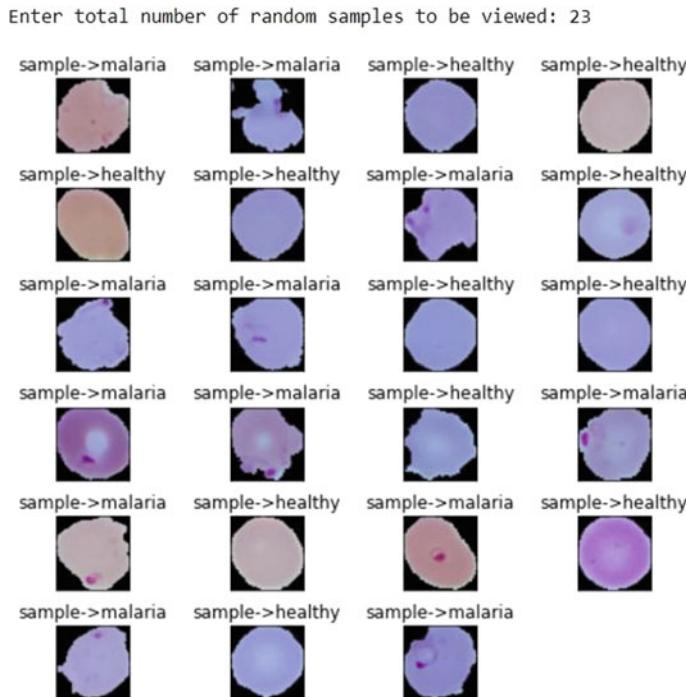


Fig. 12 Multiple random samples

The main concern with a simple CNN model is the accuracy it provides. However, with a larger dataset, efficient data pre-processing, fewer hidden layers, etc., the accuracy of the model can be improved without sacrificing its Learning efficiency. Also upon, analysis, it can be seen that neither the Training accuracy nor the Validation Accuracy is approaching 100%, this concludes the absence of overfitting, which makes the model even more efficient (Fig. 12).

Given below are the graphs for the various models built and trained. For custom CNN model (Figs. 13, 14, and 15).

The Final statistics for each model was recorded as follows (Fig. 16):

The prediction confusion matrix for the custom CNN model is as follows (Fig. 17):

The Model Classification Report of custom CNN Model is as follows (Fig. 18):

The model created for custom CNN is as follows (Fig. 19):

This allows us to understand that the efficiency of the model, though not the highest, is also not the worst. The situations where a person was infected with malaria, yet the model predicted a healthy cell were a minority case (225 cases). This is one of the shortcomings that may be easily overcome with a larger training dataset and better pooling, lesser hidden layers, etc.

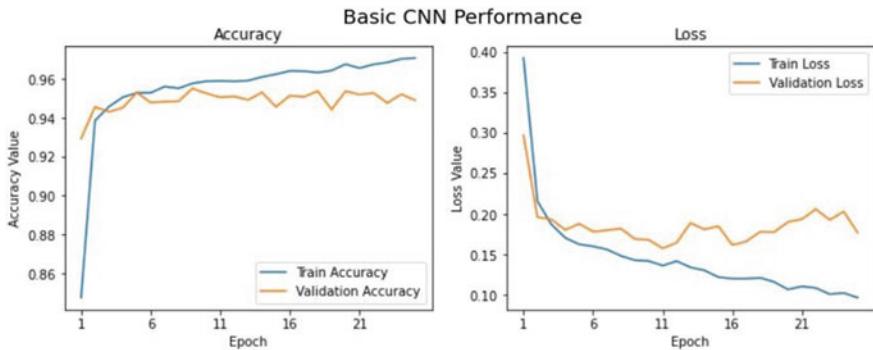


Fig. 13 Accuracy and losses for various epochs (For custom basic CNN model)

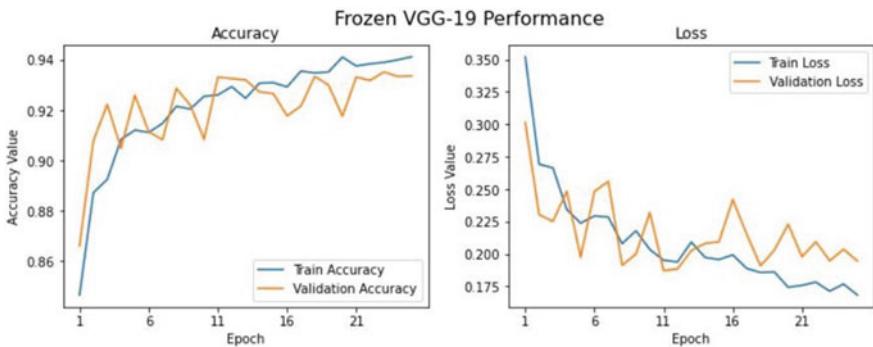


Fig. 14 Accuracy and losses for various epochs (For frozen VGG19 model)

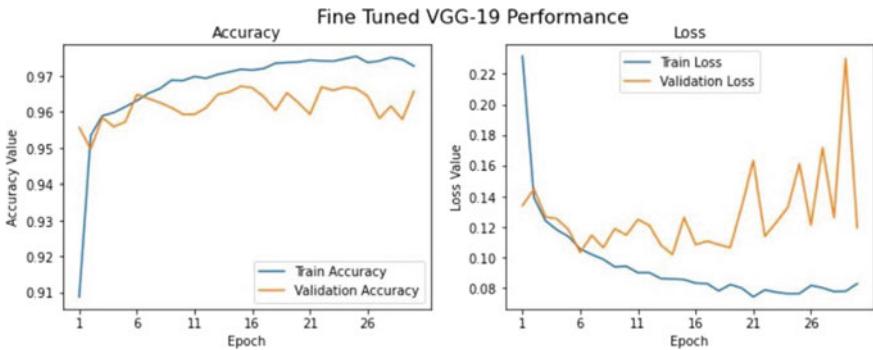


Fig. 15 Accuracy and losses for various epochs (For fine-tuned VGG19 model)

Fig. 16 Final evaluation of each model

	ACCURACY	F-1 SCORE	PRECISION	RECALL
Custom CNN	0.9501	0.9501	0.9502	0.9501
Frozen VGG19	0.9361	0.9361	0.9367	0.9361
Fine-Tuned VGG19	0.9614	0.9614	0.9618	0.9614
ResNet50	0.9755	0.9755	0.9760	0.9755

Fig. 17 Prediction confusion matrix of custom CNN

```
Prediction Confusion Matrix:
-----
Predicted:
          healthy malaria
Actual: healthy      3884    191
        malaria       225   3968
```

Model Performance metrics:

```
-----
Accuracy: 0.9501
Precision: 0.9502
Recall: 0.9501
F1 Score: 0.9501
```

Model Classification report:

```
-----
precision    recall  f1-score   support
malaria      0.96    0.94    0.95    2731
healthy      0.94    0.96    0.95    2781
accuracy           0.95    0.95    0.95    5512
macro avg     0.95    0.95    0.95    5512
weighted avg  0.95    0.95    0.95    5512
```

Fig. 18 Model classification report

5 Conclusion

Upon completion of the model and understanding the model completely, we may refer to the graphs which record all the accuracies and losses for each epoch, and the table which will indicate the accuracy and precision for different models. We can see that there are still losses and inaccuracies. The reasoning behind these losses is not confirmed. Upon hypothesizing, we can only conclude that traits among some healthy cells and some infected cells may be common. Due to these common traits, the machine may have concluded these to be healthy cells or infected cells when they were not. Furthermore, it can be argued that the use of models like ResNet-50, DenseNet 121, AlexNet, and VGG-16, may allow for better performance. While this

Model: "model"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[None, 125, 125, 3]	0
conv2d (Conv2D)	(None, 125, 125, 32)	896
max_pooling2d (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_1 (Conv2D)	(None, 62, 62, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 31, 31, 64)	0
conv2d_2 (Conv2D)	(None, 31, 31, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 15, 15, 128)	0
flatten (Flatten)	(None, 28800)	0
dense (Dense)	(None, 512)	14746112
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513

Total params: 15,102,529
Trainable params: 15,102,529
Non-trainable params: 0

Fig. 19 Custom CNN model and its parameters

may be true. This model was created simple and concise such that even in the most remote areas, people may have access to reliable prognosis. The use of heavy-duty models defeats this purpose. Furthermore, the difference between accuracies is not large to the extent where it may not be bridged when employing better techniques and more pre-processed training dataset. Given the rate of development in the field of automated diagnosis, Computer Vision, and Deep Learning, we believe that there will be much simpler yet efficient models in the future for the benefit of all people.

References

1. Dong Y, Jiang Z, Shen H, David Pan W, Williams LA, Reddy VVB, Benjamin WH, Bryan AW (2017) Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In: 2017 IEEE EMBS international conference on biomedical health informatics (BHI). IEEE, pp 101–104
2. Chaity AZ (2017) Bangladeshis flock to Indian, Thai hospitals in huge numbers. Dhaka Tribune, Retrieved from URL <https://www.dhakatribune.com/feature/health-wellness/2017/11/30/doc-tor-trust-bangladesh>

3. World malaria report 2019. Geneva: World Health Organization; 2019. License: CC BY-NC-SA 3.0 IGO
4. Global Burden of Disease Collaborative Network. Global Burden of Disease Study 2017 (GBD 2017) Results. Seattle, United States: Institute for Health Metrics and Evaluation (IHME), 2018
5. Wang H, Naghavi M, Allen C, Barber RM, Bhutta ZA, Carter A, Casey DC et al (2016) Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the global burden of disease study 2015. Lancet 388(10053):1459–1544
6. Liang Z, Powell A, Ersoy I, Poostchi M, Silamut K, Palaniappan K, Guo P et al (2016) CNN-based image analysis for malaria diagnosis. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp 493–496
7. “The Complete Beginners Guide to Deep Learning” published by Anne Bonner
8. Bibin D, Nair MS, Punitha P (2017) Malaria parasite detection from peripheral blood smear images using deep belief networks. IEEE Access 5, 9099–9108
9. Poostchi M, Silamut K, Thoma G, Image analysis and machine learning for detecting malaria
10. Nayak S, Kumar S, Jangid M (2019) Malaria detection using multiple deep learning approaches. In: 2019 2nd international conference on intelligent communication and computational techniques (ICCT), Jaipur, India, pp 292–297. <https://doi.org/10.1109/ICCT46177.2019.8969046>
11. <http://www.codeheroku.com/static/workshop/datasets/> malaria detection/train.csv
12. <http://www.codeheroku.com/static/workshop/datasets/malaria> detection/test.csv
13. Sathpathi S et al (2014) Comparing Leishman and Giemsa staining for the assessment of peripheral blood smear preparations in a malaria-endemic region in India. Malaria J 13(1):512–516
14. Tek FB, Dempster AG, Kale (2006) Parasite detection and identification for automated thin blood film malaria diagnosis. Comput Vis Image Understand 114(1):21–32
15. Zhang Z et al (2016) Image classification of unlabeled malaria parasites in red blood cells. In: 2016 38th annual international conference of the IEEE engineering in medicine and Biology Society (EMBC), Orlando, FL. IEEE, pp 3981–3984. <https://doi.org/10.1109/EMBC.2016.7591599.Medicine>
16. Liang Z (2016) CNN-based image analysis for malaria diagnosis. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 493–496
17. Boray TF, Dempster AG, Kale I (2009) Computer vision for microscopy diagnosis of malaria. Malaria J 8(1):153
18. Ross (2006) Automated imassssr the diagnosis and classification of malaria on thin blood smears. Med Biol Eng Comput 44(5):427–436
19. Hung J, Carpenter A (2017) Applying faster R-CNN for object detection on malaria images. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 56–61
20. Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR (2018) Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. Peer J 6:e4568. <https://doi.org/10.7717/peerj.4568>. PMID: 29682411; PMCID: PMC5907772
21. Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
22. Tai L, Liu M (2016) Deep-learning in mobile robotics—from perception to control systems: a survey on why and why not
23. Zheng Y, Yang C, Merkulov A (2018) Breast cancer screening using convolutional neural network and follow-up digital mammography 4. <https://doi.org/10.1117/12.2304564>
24. Ji QH, He J, Sun W, Yankui (2019) Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. Algorithms 12:51. <https://doi.org/10.3390/a12030051>

25. <https://www.machinecurve.com/index.php/2020/01/08/how-does-the-softmax-activation-function-work/>
26. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
27. Wilson ML (2012) Malaria rapid diagnostic tests. Clin Infect Dis 54(11):1637–1641

Automating Live Cricket Commentary Using Supervised Learning



Aniket S. Hegde, Kaustubh Jha, S. Suganthi, and Prasad B. Honnavalli

Abstract Cricket is one of the most popular sports in the world. It is a dynamic game involving complex rules and strategies. There are many websites like Cricbuzz and Cricinfo that provide live text commentary using journalists who report from the ground or use live telecast of the match. This process of using journalists for typing out live text commentary is both labor and cost intensive. This research presents an approach that uses dynamic web scraping to scrape live scores and associated parameters like run rate, outcome of each ball, etc. which are then fed to a supervised learning algorithm that uses all these parameters and generates a commentary for the current ball event by selecting an appropriate commentary template from a pool of predefined commentary templates. We also compare results between the different supervised learning algorithms, i.e., Neural Network and Random Forest, in terms of accuracy metrics and prove that Random Forest performs better by having an accuracy of 92.7% in generating the appropriate commentary.

Keywords Dynamic web scrapping · Supervised learning · Neural network · Multilayer perceptron · Random forest · Cricket · Commentary · Classification

1 Introduction

Cricket is the most followed sport in India. There are mainly three types of matches in cricket, namely tests, One-Day and twenty-twenty (T20), which differ mainly in duration. The type of match provides the contexts for events within match. The context of each ball can drastically change depending on the type of a match, which bowler is being given the current over, runs scored, and strike rate of the batsmen

A. S. Hegde (✉) · K. Jha · S. Suganthi · P. B. Honnavalli

Department of Computer Science, PES University, Bengaluru, Karnataka 560085, India

S. Suganthi

e-mail: suganthis@pes.edu

P. B. Honnavalli

e-mail: prasadbh@pes.edu

batting. Over is a collection of six consecutive legal balls bowled by a particular bowler. Strike rate is a number that represents the average number of runs a batsman scores for every 100 balls that they face.

Websites like Cricbuzz and Cricinfo hire journalists who use their cricket expertise to type out a textual description of the events that happen each ball. We have noticed this to be cost and time inefficient as hiring multiple journalists to cover several matches will be costly and there is significant lag of around 30–40s between when an event (fifties/hundreds hit, wickets taken, etc.) actually occurs in the match and when Cricbuzz displays the entire commentary for the event. One of the solutions to above mentioned problems is automating the said process of commentary generation. There are many ways in which we can automate this process, one such way is using supervised learning algorithms like Neural Network or Random Forest. The main purpose of this research is to replace the process of manually typing out the text-based commentary with an automatic commentary generator that uses a web scraper along with a supervised learning algorithm. We limit the scope of our research to the T20 format of cricket as it is the most “dynamic” format of cricket, i.e., it has a high number of events taking place within a short duration of time and it will serve as the basis for our supervised learning algorithm and commentary generation.

2 Literature Survey

There are two types of learning algorithms namely Supervised and Unsupervised learning. In Supervised learning, the machine is trained using data that is well labeled, i.e., the output is already known for that data [1]. We have seen very little work carried out so far, that uses supervised learning and gives a real time output on live sports. The commentary templates that we use in our study to generate the commentary are strings which are categorical data, some supervised learning algorithms like Neural networks cannot directly work on categorical data and hence it needs to be converted to numerical data, this can be done by encoding. The encoding technique we will be using in our case is One-Hot encoding [2]. The two supervised learning algorithms we'll be using are random forest and neural networks. A Neural Network is a set of neurons or nodes which are usually divided into layers [3]. The initial layer is called the input layer and final layer is called the output layer. There can be any number of hidden layers present between the input and output layers. There are many types of neural networks like Multilayer Perceptron (MLP), Radial Basis Function (RBF) networks, etc. [4]. The one which we will be using in our study is a MLP. A MLP is a class of feedforward artificial neural networks that consists of three-layer nodes: input layer, hidden layer, and output layer (Fig. 1). Its multiple layers and non-linear activation function distinguish MLP from a linear perceptron, i.e., it can distinguish data that is not linearly separable [5]. Activation function is what decides the output for a particular node given the set of the inputs. To put it simply it calculates the weighted sum of the input and adds a bias (1). This output may be given to the next layer as the input. The activation function we will be using

in our study through all the layers is Sigmoid Function (2).

$$\text{Output} = \Sigma_{\text{weight} * \text{input} + \text{bias}} \quad (1)$$

$$f(x) = \frac{1}{e^{-x} + 1} = 1 - f(-x) \quad (2)$$

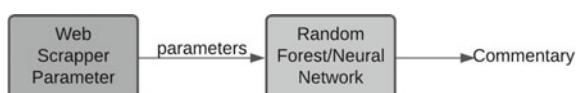
MLP can be used as a classifier, the results for these have been both consistent and useful. Ruck [5] showed a network analysis technique that showed the responsiveness of the network's output and how MLP can learn to ignore the useless features.

Random Forest (Random decision forest) is a supervised machine learning algorithm. It is dependent upon ensemble learning for classification. Here the “forest” is a collection or the ensemble of decision trees. Ensemble methods use multiple learning algorithms to obtain better predictive performance [6]. Creating a wide range of decision trees at the time of training and then performing classification results in better performance in comparison to other classifiers also [7]. A decision tree is a tree-like branched out model which acts like a tool that takes a path based on the attributes provided, which may result in different possible consequences [6]. In our study, the input to both of the supervised learning algorithms are score parameters that is obtained from the web scrapper. Web Scraping is the method by which we extract the data from a website. This information can be collected, exported, or filtered in a way such that it is more useful to the end user or it can be further processed into such a form (spreadsheet, Application Programmable Interface) that may be useful for some other tasks [8]. Dynamic Web Scraping is used to scrape those websites which are constantly changing. As there is too much information to be processed, we can use some scraping tools that run as a thread in background which makes this job easier. One such tool is Selenium Web Driver, which we will be using in our study [9].

3 Research Methodology

The architecture of the proposed system is based upon two independent services that communicate together, these services are the web-scrapper service and supervised learning service (Random Forest or MLP). The supervised learning service acts as a black box service, i.e., we do not need to know the internal details and the working of the service. The reason behind using this methodology was to make the task abstract so that both the services can function on their own, i.e., they have low-coupling and low dependency. Both services, combined, produce the cricket commentary (Fig. 1).

Fig. 1 Cricket live text commentary generator system design



The web scrapper takes a Uniform Resource Locator (URL) of the live coverage of any match in Cricbuzz as input from the user, which it parses, then depending on whether the match is Live or not it scrapes out the necessary parameters (batsman score, run rate, etc.) every ball which will be fed into the supervised learning service. Based on those parameters the trained supervised learning service selects the most appropriate commentary template from a pool of predefined templates.

3.1 Web Scrapper

Selenium Web Driver. The Selenium web driver is a tool that will help us to constantly scrape the webpage at the given URL, this tool will work in real time while increasing the efficiency of the web-scraping process in general. The Web driver fetches the contents from the URL of the live match coverage by Cricbuzz which is input by the user, that will be used for further processing.

Current Match. Once the Selenium web driver is injected, we can scrape the webpage for obtaining the current match. This will require us to search the webpage for a specific tag. We found that Cricbuzz has a unique value for classes in a HyperText Markup Language (HTML) tag, so if we wanted to search the webpage for the current match, we just need to look for a particular class. One way of doing this was to search all the header tags for a string “vs” in them which can be done using regex (regular expression).

Score parameters. It is used to extract out all the score parameters (Score parameter extractor). Generally, the score for a match where a team which is batting will be displayed in format “team runs/wicket (overs) CRR REQ”, here we have to note that the CRR stands for Current Run Rate (average number of runs scored in an over) and REQ stands for Required Run Rate (average number of runs required per over to get to the target number of runs set by team batting first). If the match is currently on the first innings REQ parameter won’t be present, hence we can also determine if it’s a chase or not. (A chase is when a team batting is second, trying to reach target number of runs set by team batting first).

Score parameter extractor: Extracting score parameters

Step 1	:	score_text <- scrape for the score string
Step 2	:	TEMP <- split score_text on “ ” (space) delimiter
Step 3	:	TEMP2 <- split score_text on “\xa0” (tab) delimiter
Step 4	:	for each word in TEMP: if team_batting is not empty and word is not empty: team_batting <- word if word has ‘(’ and ‘)’: overs <- word if word has ‘/’: runs_by_wickets <- word
Step 5	:	for each word in TEMP2: if word has “CRR”: CRR <- word if word has “REQ”: REQ <- word

State based machines. State based machines are the machines that changes the state depending on the inputs they receive. We try to incorporate this methodology when do we want particular pieces of the information to be fetched. So, in our scenario, we should fetch the baller and batsmen details only when there are some changes in overs (no runs), or there is some change in the runs (no balls) or both together. A no ball is an illegal ball by bowler on which batsman can score runs but the ball is not counted.

Batsmen and Bowlers. These parameters determine which two batsmen are currently batting, bowler bowling the current over and bowler that bowled the previous over. One way to search for them is to locate all the HTML div tags with the particular class attribute having a value of “batsmen” and “bowler” for batsmen and ballers respectively.

Outcome. Outcome [10] of the ball is determined by the end result of the ball. This can be some runs, no balls, wides, wickets, or byes. The way the outcome is determined is by string matching. When the ball is bowled, the outcome is produced instantly on the webpage. Generally, the commentary for the ball is in the format of the following example “17.6, Dale Steyn to Virat Kohli, FOUR”. After scraping all the commentaries, we can use string matching to return the outcome (in above example outcome is FOUR). The outcome is in the form of text and as the classifier does not understand what a string variable is, we convert the outcome to an integer, for example, no run is 0, four is 4, wicket is 8 (or 9 in case of run out), etc.

3.2 Supervised Learning

Dataset Generation. In our study, we have applied various logic to generate a well distributed dataset. The Dataset covers all the aspects from a fast century to a slow fifty and from 1st wicket to the 10th wicket.

The first case we have covered is for centuries. It can be a fast/slow century, fast/slow century while chasing. Generally, we have observed that a century is considered fast when the strike rate of the batsmen is above 165, i.e., he/she takes less than 61 balls to reach 100 runs. The maximum strike rate is set as 330 because the fastest 100 in history of cricket was scored in 30 balls at strike rate of 330. We also observed that a century can be considered slow or a normal paced century when it takes 61 or more balls, i.e., strike rate less than 165. The same aforementioned logic is used for fast/slow century while chasing with the only difference being that the team is batting second which is indicated by the variable chase being set to 1 (Century dataset generator).

Century dataset generator: Dataset generation for fast century slow century, fast chase and slow chase.

```

Step 1      : counter = 1
Step 2      : for i in range 0 to 800:
Step 3      :   batsmen_score <- random integer between 94 to 99
Step 4      :   wickets <- random integer
              between 0 to 9
Step 5      :   chase <- random integer between
              0 or 1
Step 6      :   if i is even:
                  counter <- counter XOR 1
                  bowler_economy <- random
                  number between 0 to 36
                  bowler_wicket <-random integer between 0 to 6
                  if counter is 0:
                      strike_rate <- random number between 165 to 330
                  else:
                      strike_rate <- random number between 135 to 165
                  else:
                      bowler_economy <- random number between 0 to 18
                      bowler_wicket <- random integer between 0 to 6
                  if counter is 0:
                      strike_rate <- random number between 135 to 165
                  else:
                      strike_rate <- random number between 165 to 330

```

Step 7	:	outcome <- 6
Step 8	:	<p>if chase is 1:</p> <p>if strike_rate is greater than or equal to 165:</p> <p style="padding-left: 20px;">template <- unique template number from created templates</p> <p>else:</p> <p style="padding-left: 20px;">template <- a different unique template number from created templates</p> <p>else:</p> <p>if strike_rate is greater than or equal to 165:</p> <p style="padding-left: 20px;">template <- unique template number from created templates</p> <p>else:</p> <p style="padding-left: 20px;">template <- a different unique template number from created templates</p>
Step 9	:	data <- [batsmen_score, strike_rate, wickets, bowler_economy, bowler_wicket, outcome, template]
Step 10	:	Insert data in database or csv file

(Century dataset generator) generates 200 rows for each case, the reason for some specific numbers given in algorithm is both logical and observational. The counter variable is added to ensure that there is no correlation between bowler statistics like economy or number of wickets and the batsman's strike rate so that the dataset is truly random. The upper limit for a number of wickets for a particular bowler is set to 6 as the record for highest number of wickets picked up by a bowler in T20 cricket stands at 6. During an over by a bowler, his/her economy (average number of runs given per over) may increase but it is very rare to see the economy rise above 18. Finally depending on strike rate at which the century was scored we assign the appropriate commentary template from a pool of predefined commentary templates.

The second case we have considered is for fast/slow fifties and fast/slow fifties while chasing. These cases have been handled using an algorithm similar to the one used in case of century with the only difference being the threshold strike rate values which separate fast and slow fifty. We observed that a fifty by a batsman can generally be considered fast if strike rate is greater than 150, i.e., takes 32 or fewer balls to hit 50 runs. Consequently, if a batsman takes more than 32 balls to hit 50 runs, i.e., strike rate lesser than 150, the fifty can be considered slow or normal paced.

Third case is for wickets, there are various different types of wicket, for example, if a well-set batsman (batsman who has scored 50 or more runs) in any innings gets out, i.e., his wicket is taken, it's a crucial wicket. When a batsman with a very low strike rate or who had just come into bat gets out, that wicket doesn't hold the same importance as that of a well-set batsman. The wicket of a batsman who is not a well-set batsman but has a very high strike is also an important wicket but in a different way to that of a well-set batsman. We divided the types of wicket in such a way that it covers many such possibilities and have assigned appropriate commentary templates to each of them (Wicket dataset generator).

Wicket dataset generator: Wicket dataset Generation

Step 1	:	for i in range 0 to 200:
Step 2	:	batsmen_score <- select a random integer less than 100
Step 3	:	wickets <- random integer between 0-6
Step 4	:	chase <- random integer between 0 or 1
Step 5	:	bowler_economy <- random number between 0 to 18 strike_rate <- random number 0 and 330
Step 6	:	outcome <- out
Step 7	:	template <- template number (one of 2-37) depending on preceding parameters
Step 8	:	data <- [batsmen_score, strike_rate, wickets, bowler_economy, bowler_wicket, outcome, template]
Step 9	:	Insert data in database or csv file

(Wicket dataset generator) generates different types of wickets, when the values of the bowler_economy, bowler_wicket, strike_rate, or batsman_score variables are changed. Subsequently, an appropriate commentary template is assigned to each type of wicket.

A very important point is that all the values for every case mentioned above are heuristically decided taking many T20 matches into consideration.

Classification. There are 36 manually defined templates for commentary in our template pool. There are 2 types of model we can use to classify the current ball event as one of the commentary templates from our template pool depending on the parameters we obtain from the web-scraping service, these are MLP and random forest. When using MLP to classify current ball event, min-max normalization is performed on the dataset. Min-max normalization is rescaling of a feature column such that all values in column fall in range [0,1]. This is achieved by dividing each value in feature column by length of the range (minimum value in a range subtracted from maximum value in the range) of the feature column. One-Hot-Encoding is performed on the template column because the 36 templates are represented as numbers from 2 to 37 and without encoding the MLP assumes a relation between the template numbers. The number of layers is determined as 3, one input layer (7 nodes each corresponding to a single feature), one output layer (36 nodes, each corresponding to a commentary template), and a hidden layer (21 nodes, heuristically selected to increase accuracy). Activation functions are defined between adjacent layers (sigmoid).

When using random forest for classifying current ball events, no data preprocessing was required as they can directly work on categorical data.

For a clear comparison between the two models (Random Forest and MLP) in terms of accuracy metrics, we perform cross validation (with $n = 5$, i.e., dataset is split into 5 parts with one part acting as testing data and 4 others as training data in an iteration).

4 Results and Discussion

Accuracy metrics for MLP and Random Forest (Fig. 4) is computed with the help of external python libraries. The accuracy metrics used are precision [11], recall [11] and F-score [12]. The performance of the entire system is measured using delay (6).

(1) **Precision** Precision (3) in classification is the fraction of relevant instances among the retrieved instances [11].

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (3)$$

(2) **Recall** Recall (4) is the fraction of relevant instances that were retrieved [11].

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (4)$$

(3) **F-score:** F-score (5) is a measure of a test's accuracy. It is calculated from the Precision and Recall of the test as follows [12].

$$F - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (5)$$

(4) **Delay** Delay of the entire architecture can be defined using the (6).

$$\text{Delay} = \Delta_{\text{web scrapper service delay}} + \Delta_{\text{supervised learning service delay}} \quad (6)$$

From Figs. 2 and 3, we see that a major part of the delay of architecture (6) stems from the web-scrapper service (3–7 s of delay on an average per match) whereas the supervised learning service delay is almost negligible (0.042–0.065 s of delay on an average per match). Adding the 2 delays, average total delay per match falls in the range of 3.042–7.26 s (average across 5 matches being 5.134 s).

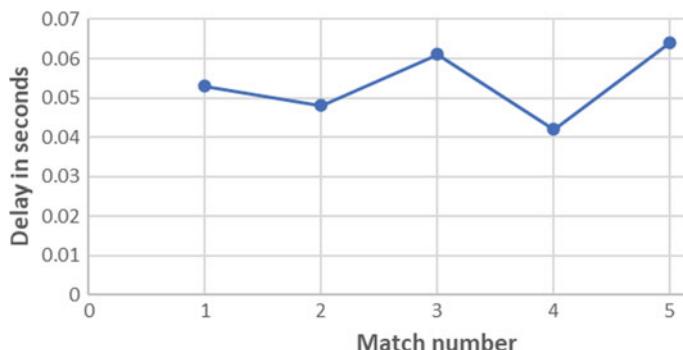
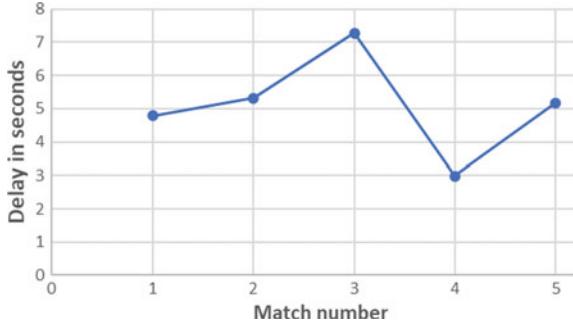


Fig. 2 Average delay of supervised learning service per match across 5 matches (in seconds)

Fig. 3 Average delay of web-scrapper service per match across 5 matches (in seconds)



Figures 4 describes the class wise accuracy metrics for Random Forest and MLP respectively. Here class numbers from 2 to 37 represent the 36 commentary templates. We can observe that Random forest slightly outperforms MLP in all the accuracy metrics.

From Figs. 5 and 6, We can observe that 6.2 s after the occurrence of the event in the live match, only the outcome of the ball has been output by cricbuzz and the descriptive commentary is yet to be generated whereas in case of proposed system both outcomes of ball and descriptive commentary has already been generated.

5 Conclusion and Future Work

Websites like Cricbuzz and Cricinfo that provide live text commentary for cricket matches use journalists to do so, this process is both labor and cost intensive. This research presents an approach to automate live text commentary for cricket matches using two loosely coupled, independent services namely a web scraper service and a supervised learning service (MLP or Random forest). The web scraper service is used to scrape live scores and associated parameters like run rate, outcome of each ball, etc. from cricbuzz, which are then fed to the supervised learning service which uses all these parameters and generates a commentary for the current ball event by selecting an appropriate commentary template from a pool of predefined commentary templates. This study also compares MLP and random forest classifiers in terms of accuracy metrics and found that the random forest classifier is a better option for this use case because all the accuracy metrics are higher for Random Forest compared to MLP. This project also calculates the delay for the entire system and we found out that real time output is not possible. Even though the output of the system is not real time, delay for the system is significantly lower than that of Cricbuzz, which had an average delay per match of 14.7 s to generate descriptive commentary across 5 matches whereas the proposed system had an average delay per match of 5.134 s to generate descriptive commentary across 5 matches. Findings in the study have shown that the major portion of the entire system's delay stems from the web-scrapper service, which is also very resource intensive, to solve this problem

	Random forest			Class number	MLP		
	Precision	Recall	F-score		Precision	Recall	F-score
	1.000	1.000	1.000	2	1.000	0.995	0.997
	1.000	1.000	1.000	3	0.991	0.991	0.991
	1.000	1.000	1.000	4	0.995	1.000	0.997
	1.000	1.000	1.000	5	0.991	0.991	0.991
	1.000	1.000	1.000	6	1.000	0.995	0.997
	1.000	1.000	1.000	7	0.995	1.000	0.997
	1.000	1.000	1.000	8	1.000	0.995	0.997
	1.000	1.000	1.000	9	0.990	0.995	0.992
	1.000	1.000	1.000	10	0.986	1.000	0.993
	1.000	1.000	1.000	11	1.000	0.991	0.995
	1.000	1.000	1.000	12	0.985	0.985	0.985
	1.000	1.000	1.000	13	0.995	0.984	0.989
	0.995	1.000	0.997	14	0.984	0.985	0.984
	1.000	1.000	1.000	15	0.995	0.984	0.989
	1.000	1.000	1.000	16	0.990	1.000	0.995
	1.000	1.000	1.000	17	0.986	0.959	0.972
	1.000	1.000	1.000	18	0.980	0.976	0.978
	1.000	1.000	1.000	19	0.972	0.986	0.979
	0.643	0.632	0.637	20	0.626	0.613	0.619
	0.918	0.950	0.934	21	0.873	0.860	0.866
	0.678	0.692	0.685	22	0.645	0.623	0.634
	0.875	0.980	0.925	23	0.812	0.945	0.873
	0.937	0.890	0.913	24	0.879	0.870	0.874
	0.907	0.975	0.940	25	0.850	0.965	0.904
	0.949	0.930	0.939	26	0.878	0.870	0.874
	0.661	0.662	0.661	27	0.643	0.693	0.667
	0.659	0.627	0.643	28	0.648	0.627	0.637
	0.865	0.960	0.910	29	0.813	0.955	0.878
	1.000	1.000	1.000	30	0.985	0.955	0.970
	1.000	0.995	0.997	31	0.984	1.000	0.992
	1.000	1.000	1.000	32	0.985	0.943	0.964
	1.000	1.000	1.000	33	0.995	1.000	0.997
	1.000	1.000	1.000	34	1.000	1.000	1.000
	1.000	1.000	1.000	35	0.995	1.000	0.997
	1.000	1.000	1.000	36	0.985	0.943	0.964
	1.000	1.000	1.000	37	0.995	1.000	0.997
Weighted average	0.921	0.927	0.924		0.903	0.910	0.906

Fig. 4 Class wise statistics for random forest and MLP (left and right respectively)

8.5 Rahul Chahar to Shubman Gill, out Caught by Pollard!! Shubman Gill c Pollard b Rahul Chahar 33(24) [4s-5 6s-1], Rahul Chahar picks up his first wicket, Shubman Gill falls just when he was looking set, ready to accelerate.

Fig. 5 Commentary generated by proposed system for 8th over 5th ball for a particular match, 6.2 s after the actual event took place in live match

8.5 Rahul Chahar to Shubman Gill, **out Caught by Pollard!! Shubman Gill c Pollard b**
Rahul Chahar 33(24) [4s-5 6s-1]

Fig. 6 Cricbuzz commentary for 8th over 5th ball for same aforementioned match, 6.2 s after the actual event took place in live match

we can introduce threading or a better solution would be to replace the current web scraping service with an application of computer vision [13].

References

1. Cunningham P (2008) Supervised learning. Springer, Berlin, Heidelberg
2. Staff PE (2019) How to handle categorical data for machine learning algorithms. (Packt) Retrieved from <https://hub.packtpub.com/how-to-handle-categorical-data-for-machine-learning-algorithms/>
3. Wang S-C (2003) Artificial neural network. In: Interdisciplinary computing in java programming. The springer international series in engineering and computer science, vol 743. Springer, Boston, 81–100
4. Ghoshal A, Types of neural networks. (EDUCBA) Retrieved from educba: <https://www.educba.com/types-of-neural-networks/>
5. Ruck DW (1990) Feature selection using a multilayer perceptron. J Neural Netw Comput 2(2):40–48
6. Opitz D (1999) Popular ensemble methods: an empirical study. J Artif Intell Res
7. Rokach L (2010) Data mining and knowledge discovery handbook. Springer Science+Business Media, LLC
8. Perez M (2019) What is web scraping and what is it used for. (Parsehub) Retrieved from <https://www.parsehub.com/blog/what-is-web-scraping/>
9. Nayak R (2019) Web scraping using beautiful soup and selenium for dynamic page. (Medium) Retrieved from <https://medium.com/ymedialabs-innovation/web-scraping-using-beautiful-soup-and-selenium-for-dynamic-page-2f8ad15efe25>
10. Kumar R, Santhadevi D, Barnabas J (2019) Outcome classification in cricket using deep learning. In: 2019 IEEE international conference on cloud computing in emerging markets (CCEM), pp 55–58. <https://doi.org/10.1109/CCEM4844.2019.00012>
11. Powers DM (2019) Evaluation: from precision, recall and F-factor to ROC, Informedness, Markedness & Correlation
12. Wood T, What is the F-score? (deepai) Retrieved from deepai.org: <https://deepai.org/machine-learning-glossary-and-terms/f-score>
13. Bhalla A, Ahuja A, Pant P, Mittal A (2019) A multimodal approach for automatic cricket video summarization. In: 2019 6th international conference on signal processing and integrated networks (SPIN), pp. 146–150. <https://doi.org/10.1109/SPIN.2019.8711625>

Real-Time Object Detection and Distance Approximation



Rohit Beniwal and Ashish Singh

Abstract Ordinary diurnal tasks can be very strenuous for people with visual defects. Over the years many contemporary technologies have been developed to help visually impaired persons. However, these technologies are only able to narrate the contents on a mobile screen and do not help in describing the real-world objects around a visually impaired person. The purpose of this research work is, therefore, to provide a system that can detect an object and predict its distance and direction from an individual in real-time. The proposed system is a combination of an object detection model and a novel algorithm to approximate the distance and direction of objects called distance approximation algorithm. The detection and localization of objects are carried out by MobileNet and Single Shot Detector which are deep neural networks and are pretrained on the COCO dataset. The detection model highlights the identified objects by means of labeled bounding boxes. The coordinates of these bounding boxes are then used by the distance approximation algorithm to predict an object's distance and direction. The system is tested using different images and live video feed from a camera, however in order to determine the efficiency of the system images of a single object taken from various distances are used. Findings indicate that the system achieves an average accuracy of 96% in predicting the distance and thus, would be able to be effective in aiding visually impaired or blind persons.

Keywords Deep neural networks · MobileNet · Single shot detector · Object detection · Object recognition · Object localization · Distance approximation

1 Introduction

Image processing refers to the process where some operations are performed on an image, which makes it fit for extracting useful information. This useful information is mainly features or characteristics of the image provided as an input to the image processor. The output of the image processor is further used to detect various

R. Beniwal · A. Singh (✉)

Department of Computer Science and Engineering, Delhi Technological University, Delhi 110042, India

objects in the image using a computer vision technique known as object detection. Object detection algorithms rely on machine learning to identify various patterns from feature extraction of an image and based on them, they distinguish between different objects. Therefore, we also use the application of image processing and machine learning to provide a model for detecting an object in the blind person's surroundings and predicting how far it is from the concerned person and what is its related direction. The solution to this model can help the people, who are suffering from blindness, glaucoma, diabetic retinopathy, etc. by sensing the surrounding environment and thus assisting them to know the things around them with an approximate distance.

As per the work of Bourne et al. in Lancet Global Health [1], an estimated 217 million people suffer from moderate to severe visual impairment and 36 million are blind. Functional presbyopia affects an estimate of 1094.7 million people, out of which 666.7 million people are aged 50 years and above. The rise in the number of elderly will increase the percentage of the population who are at risk of visual impairment [1]. Further, these visual defects are not limited to just the elderly, children aged below 15 who are in the prime of their lives are also suffering from these defects. Moreover, according to the World Health Organization's (WHO) data, globally around 2.2 billion people suffer from visual impairment. Therefore, these people are in dire need of aid to enhance their vision and impede the progression of their disability to sense the world better [2].

Amidst this, there are 15 million people, who are suffering from blindness, thus making India, home to the biggest blind population on the globe [3]. Globally, it is estimated that overall, 40–45 million people are blind and cannot walk without any assistance [4]. There are few apps, which help blind or visually impaired people in navigating through public places using assistance from volunteers of these apps. However, these apps have a limitation that they are always dependent on the assistance of volunteers. However, in this research work, we provide a system, which will work independently of any volunteer, i.e., without any other person's assistance. In the intended system, we use MobileNet and Single Shot Detector (SSD) for detecting objects from an image or a video frame. The video frames are captured from a live video stream using python's OpenCV library. After detecting an object, its approximate distance is calculated using the distance approximation algorithm, which is discussed later in the paper.

The rest of the paper is organized as follows: Sect. 2 discusses the related works; Sect. 3 explains the research approach followed by Sect. 4, which describes the implementation of the research approach along with the discussion on results and analysis; Finally, Sect. 5 concludes the research paper and provides the direction for future work.

2 Related Work

Although many researchers have worked on object detection and image processing, however, to the best of our knowledge, no work has been found so far to detect an object in real-time and predict its distance using neural networks and image processing techniques. Although there are various tools, from a simple cane to the complex system using software and hardware, to assist a blind person. However, their feasibility varies from indoor to outdoor along with dynamic surroundings. In the 1990s, Golledge et al. were the first ones to propose a conceptual model that intended the use of Geographic Information System (GIS), sonic sensor components, Global positioning system (GPS), and speech for helping the visually impaired [5]. An example of an implemented model is a system called MOBIC, which is based on GPS for the aid of the less visually privileged. It also uses a voice command to dictate the path and direction to the blind user [6]. Similarly, Drishti, which is a wireless pedestrian navigation system, devises a path for the user that incorporates various technologies such as mobile computers, wireless networks, voice recognition, GIS, and GPS. Drishti System imbibes all the surrounding information and then evaluates an optimized path for the destination of the concerned blind user [7]. In addition, technologies such as RFID chips use a lot of hardware infrastructure, being highly static in nature, require advanced positioning of chips, and are only limited to indoor systems [8]. Another example of a device that can sense its environment and can walk avoiding obstacles is a robot known as Lola, which is a human-sized biped robot that uses onboard sensing and 3D point cloud processing techniques [9]. With the outbreak of coronavirus disease, several researchers have worked on measuring the social distance between the two objects [10]. However, their work is still lagging in providing any assistance to a visually impaired person. Therefore, in this research work, we provide a system that can fulfill the gaps of the earlier works where a system has been designed and implemented, which works independently without requiring any person's assistance. Moreover, the system detects the objects in real-time and predicts their approximate distance from the blind person.

3 Research Approach

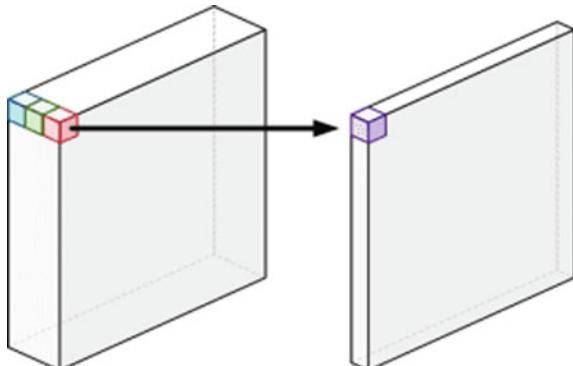
The research approach for real-time object detection and distance approximation is divided into two phases namely object detection and distance approximation, which are as follows:

3.1 Object Detection

An amalgamation of MobileNet and SSD, which is trained on the COCO dataset, allows us to detect, recognize and localize multiple objects in an image. This combination is much more computationally economical and doesn't sacrifice much accuracy. For the classification of objects, we will use MobileNet Architecture. MobileNet is a lightweight neural network, which uses depth-wise separable convolutions. The MobileNet uses the standard convolution in which all the input channel values are combined by the convolution operation. The standard convolution is only used as the first layer where a single channel per pixel image is obtained as output by running a single convolution kernel across an image that has 3 input channels. The rest of the layers do "depth-wise separable" convolution. Two different convolution operations are combined here, a point-wise convolution and a depth-wise convolution. In a regular convolution, the input channels are combined which is not the case with depth-wise convolution where convolution is carried out on each channel explicitly. A depth-wise convolution creates an output image with 3 channels for a 3-channel input image. A unique set of weights is assigned to each channel. Tasks like filtering the input channels, edge detection, color filtering, etc. are performed by the depth-wise convolution. A point-wise convolution is succeeded by the depth-wise convolution [11, 12]. A 1×1 kernel is what makes it different from a regular convolution as shown in Fig. 1 [11].

In other words, the channels are all simply summed up as a weighted sum. In a regular convolution, many of these point-wise kernels are usually stacked up together with many channels to create an output image. The purpose of this point-wise convolution is to create new features by combining the output channels of the depth-wise convolution. The result is known as a depth-wise separable convolution made by putting together these two convolutions, namely a point-wise convolution succeeded by a depth-wise convolution. The task of filtering and combining is done in a single step in a regular convolution, however, these are performed at different steps with a depth-wise separable convolution [11, 12].

Fig. 1 A point-wise convolution



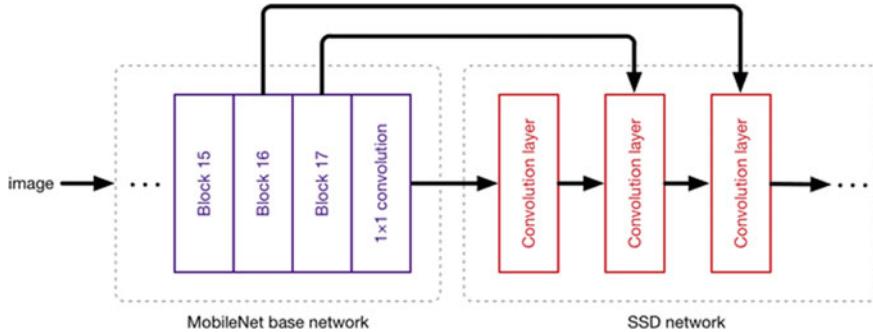


Fig. 2 MobileNet SSD architecture

This process will help us bring down the computation cost of the process from " $D_1 \cdot D_1 \cdot X \cdot Y \cdot D_2 \cdot D_2$ " to " $D_1 \cdot D_1 \cdot X \cdot D_2 \cdot D_2$ " where Y and X are the number of output and input channels respectively. Here, $D_1 \times D_1$ is the kernel size and $D_2 \times D_2$ is the feature map size [11].

SSD is used to localize the objects in an image. SSD is supposed to be free from the bottom network in a way that it can simply run above everything. Here, we will combine it with MobileNet [12]. MobileNet + SSD as shown in Fig. 2 [13] shows that instead of regular convolutions, depth-wise separable layers are used for the network's object detection part. With the SSD sandwiched above MobileNet, we will get real-time results. The conversion of pixels from the input image into features is carried out by the MobileNet layers. These features describe the objects within the image and the next layer receives them as a forward. MobileNet will be employed here as a feature extractor for the SSD [13].

The need to feed forward the low-level features of MobileNet to SSD convolution layers is because we do not want only the classification of objects, but also the object's location in each image. Therefore, for this, we would not only connect the high-level features of the MobileNet Network but would also feed forward the low-level features to localize the object. Figure 3 depicts the proposed control and data flow of the object detection process.

3.2 Distance Approximation

First, we would need the objects to be detected from the image or videos. TensorFlow Object Detection API will be used for object detection, which is going to discover different objects from the images. It can detect 79 different classes of objects. For videos, OpenCV will be used to feed the video frames to the object detection model. After detecting the object and classifying its class, the coordinates of the bounding box are to be evaluated, which would be further used to estimate the object's distance from the camera. For the purpose of distance prediction, we need to pre-store one

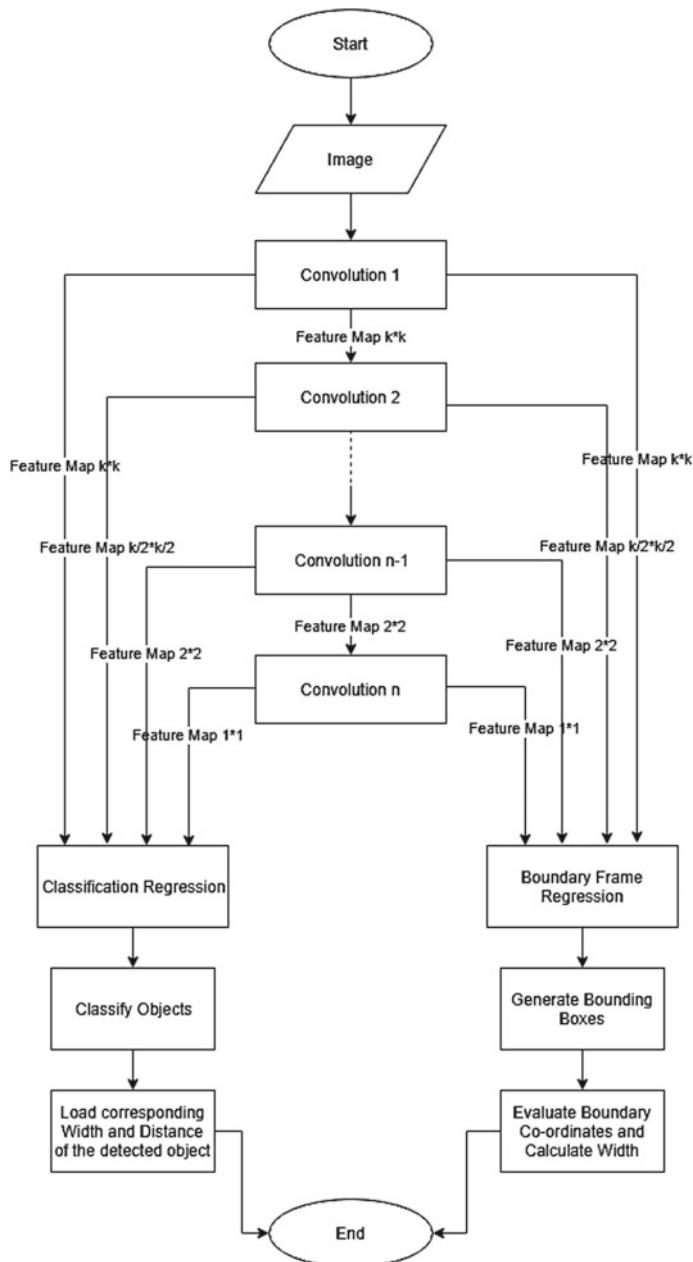


Fig. 3 Object detection process

instance of the object belonging to every class from a distance and the original size of the object must also be known. By using this instance, a formula is defined and that will be used to predict the distance for other instances. Now, for the direction of the object, coordinates of the bounding box are to be used. By using the coordinates, the quadrant can be defined in which the center of the object lies.

This method of distance estimation will be implemented on images as well as on videos. For distance estimation, we considered that the distance and size of the object are inversely proportional to each other. Assume that we place the object at a distance ' d ' inches from the camera and the size of the object in the image is ' S ' units. Now, we store these values once and for all, when this object is detected in some other image, the relation would be $S * d = s * D$, where ' s ' is the new size of the image and ' D ' is the new distance. Hence, the formulae for the distance will be $D = \frac{(S*d)}{s}$. The algorithm for distance approximation is as follows:

1. Detect an object and its class using object detection API (TensorFlow).
2. Find out the coordinates for the bounding box which are the coordinates of the principal diagonal of the rectangular box (x_1, y_1) and (x_2, y_2) .
3. Now, get the values of ' S ' and ' d ', previously stored for the object, which are training values.
4. Calculate new size of object by the Eq. 1: $s = x_2 - x_1$.
5. Distance $D = \frac{S*d}{s}$
6. To find out direction of object
 - Calculate the center of the object that is $\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}$.
 - Now, by examining where this point lies in the coordinate system of the camera screen or the window that is used, we can tell its direction.

4 Implementation, Results, and Analysis

To implement the research approach, there are two alternatives available. One is to build and train the model from scratch for object detection and then implement the distance approximation algorithm on top of the new model. The other alternative is transfer learning, where an already trained model is pulled to achieve the task. However, we chose the second alternative since it is a more efficient method as compared to the first one.

To proceed with the second alternative, we used the TensorFlow object detection API to pull “ssd_mobilenet_v1_coco_2017_11_17,” which is an already trained model on the COCO dataset. Now, with the model in hand, we used OpenCV to capture live frames via camera and then fed them to the model for classification. After classification is done on a frame, bounding boxes are drawn around every recognized object and the coordinates of those objects are then fed to the distance approximation algorithm, which ultimately gives us the distance and the direction of the object.

Through this module, we were able to identify and localize the prominent objects in the image. Along with that, we were also able to extract the coordinates of the bounding box, which with respect to the camera or user helps in finding the relative direction and distance of the classified object to aid the visually impaired.

4.1 Object Detection

The results of the implementation of the object detection approach are represented by the following figures. For, e.g., Fig. 4 shows the image of two street dogs and their successful detection as an output of the model with 93% confidence along with their coordinates as shown in Fig. 5.

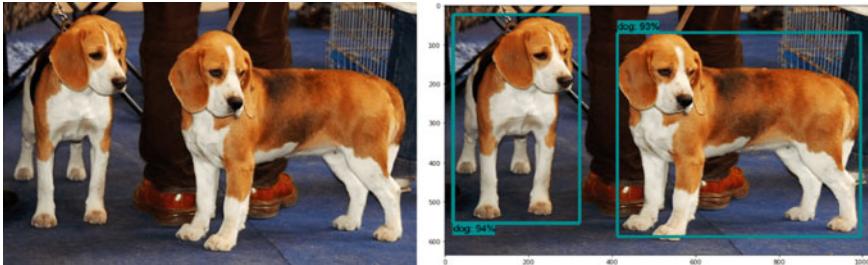


Fig. 4 Description of images from left to right: (1) image of 2 street dogs, (2) two street dogs predicted from the image

```

dog
dog
24.85745394229889 19.676193237304688 554.6577959060669 323.35205078125
69.65154719352722 412.503662109375 588.0749065876007 996.4010009765625

```

Fig. 5 Predicted dogs coordinate in image

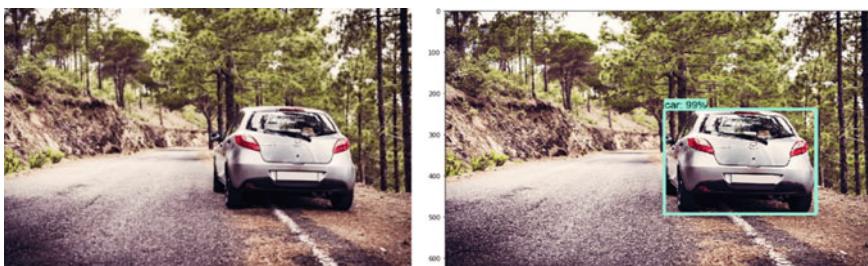


Fig. 6 Description of images from left to right: (1) car on a road, (2) predicted car from the image

```
car
239.68597412109375 491.24759674072266 493.0321502685547 832.8451538085938
```

Fig. 7 Predicted coordinates

Likewise, Fig. 6 shows an image of a car on the road, and successful detection as an output of the model with 99% confidence along with its coordinates as represented by Fig. 7.

After analyzing the above results with a live webcam, it can be said that they are satisfactory and this module can recognize various objects in an image, which can assist a visually impaired person. Since this module is computationally economical and light, it can provide real-time assistance to the user by using a live feed camera and voice commands.

4.2 Distance Approximation

We implement and test this approach by using images of a bicycle taken from different distances. The results in this module are obtained using the formulae as defined in Sect. 3.2. According to the formulae, first, we pre-stored one instance of the bicycle image as shown in Fig. 8a. For this image we already knew: Distance $d = 60$ inches and Size of the image from 60 inches, $S = 3300$.

Accordingly, these pre-stored are provided as a base for all other images. Distance prediction is then carried out based on these two parameters ‘ S ’ and ‘ d ’. The first case that is considered is for an image where the bicycle distance from the camera is 92 inches as shown in Fig. 8b.

The distance of the bicycle from the camera will be, $D = \frac{(S \cdot d)}{s}$.



Fig. 8 a Bicycle at a distance of 60 inches, (b) bicycle at distance 92 inches

Table 1 Comparison of predicted and original distances of the object

Original distance (inches)	Predicted distance (inches)	Percentage error
52	54	3.8462
68	65	4.412
76	73	3.947
84	79	5.952
92	89	3.261
100	97	3
108	101	6.481
82	79	3.659
125	120	4
Average percentage error		4

where, ‘ D ’ is the new Distance of the bicycle from the camera, ‘ S ’ is the size of the bicycle from a predefined distance, ‘ d ’ is the distance of object from a predefined distance and ‘ s ’ is the new size of the object.

‘ S ’ and ‘ d ’ are pre-stored whereas ‘ s ’ is calculated using the object coordinates as obtained in Sect. 4.1. The formulae for the calculation of ‘ s ’ is as discussed in Eq. 1 of Sect. 3.2. Here $s = 3129.74 - 913.75 = 2215.99$.

Consequently, $D = (3300 * 60) / 2215.99 = 89.35$ inches.

Similarly, distance approximation is carried out for various bicycle images taken from different distances and a table is compiled for their analysis. Table 1 shows a comparison of predicted and original distances of objects in inches along with their percentage error.

After analyzing the above table, it is found that the average error margin is 4% and thus resulting in accuracy of 96% in predicting the distances as compared to the original ones. Therefore, it can be said that this model can help a blind person to sense and be aware of his surroundings.

4.3 Real-Time Object Detection Using Webcam

Result as depicted by Fig. 9 shows the real-time feasibility and proficiency of the proposed system in detecting multiple objects of the same as well as of different types. This system has also been tested using a webcam and is perfectly detecting as well as predicting the distance of objects in live streaming. Also, the direction of the objects has been correctly determined by the system as depicted by Fig. 10.



Fig. 9 Object detection using webcam

Fig. 10 Direction of objects

```
Location of cell phone is Right
FOUND..... cell phone at
Distance of cell phone is 179.81865238699862
Location of cell phone is Right
50
FOUND..... person at
Distance of person is 26.463277887138606
Location of person is Right
51
FOUND..... cell phone at
Distance of cell phone is 29.42487469687771
Location of cell phone is Right
```

5 Conclusion and Future Scope

The purpose of doing this research work was to develop modules that can be integrated together to construct a more sophisticated system of object detection and distance approximation because such systems can be used to solve many real-world problems. Therefore, in this research work, we proposed a system for detecting the objects as well as finding out their approximate distances. To the best of our knowledge, this is the first attempt that combines detection and distance approximation of objects. MobileNet and SSD are used to detect and localize objects from input sources. The localized objects have bounding boxes drawn around them by the detection model. The coordination of these bounding boxes is then used by distance approximation algorithm to estimate the distance and direction of objects. We then tested this system

on some real-life objects as obtained from images and live streaming videos. As a result, the system is able to detect, recognize, and find the localized position of the objects in a given image or in live streaming videos. Furthermore, the system is capable of approximating an object's distance with an average accuracy of 96%. Also, the system is able to find the object's relative direction from the camera. This unique integration of the techniques as used in the proposed system will assist people with visual impairment in their day-to-day life activities. Moreover, as part of future work, the system can be implemented using voice prompts, which will assist disabled persons.

References

1. Bourne R, Flaxman S, Braithwaite T et al (2017) Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. Lancet Glob Health 5:e888–e897. [https://doi.org/10.1016/s2214-109x\(17\)30293-0](https://doi.org/10.1016/s2214-109x(17)30293-0)
2. World Health Organization (WHO) <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
3. India has the largest blind population (Kounteya Sinha, Oct 11, 2007) <https://timesofindia.indiatimes.com/india/India-has-largest-blind-population/articleshow/2447603.cms>
4. World Health Organization (WHO) <https://www.who.int/blindness/Vision2020%20-report.pdf>
5. Golledge R, Loomis J, Klatzky R et al (1991) Designing a personal guidance system to aid navigation without sight: progress on the GIS component. Int J Geograph Inf Syst 5:373–395. <https://doi.org/10.1080/02693799108927864>
6. Petrie H, Johnson V, Strothotte T et al (1996) MOBIC: designing a travel aid for blind and elderly people. J Navig 49:45–52. <https://doi.org/10.1017/s037346300013084>
7. Ran L, Helal S, Moore S (2004) Drishti: an integrated indoor/outdoor blind navigation system and service. In: Proceedings of the second IEEE annual conference on pervasive computing and communications, pp 23–30. <https://doi.org/10.1109/PERCOM.2004.1276842>
8. Chun kamon S, Tuvaphanthaphiphat P, Keeratiwintakorn P (2008) A blind navigation system using RFID for indoor environments. In: 2008 5th international conference on electrical engineering/electronics, computer, telecommunications and information technology, pp 765–768. <https://doi.org/10.1109/ECTICON.2008.4600543>
9. Wahrmann D, Hildebrandt A, Wittmann R, Sygulla F, Rixen D, Buschmann T (2016) Fast object approximation for real-time 3D obstacle avoidance with biped robots. In: IEEE international conference on advanced intelligent mechatronics (AIM) 38–45. <https://doi.org/10.1109/AIM.2016.7576740>
10. Ahmed I, Ahmad M, Rodrigues JJPC, Jeon G, Din S (2021) A deep learning-based social distance monitoring framework for COVID-19. Sustain Cities Soc 65:102571. ISSN 2210-6707. <https://doi.org/10.1016/j.scs.2020.102571>
11. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)
12. Google's MobileNets on the iPhone (14 JUNE 2017). <http://machinethink.net/blog/googles-mobile-net-architecture-on-iphone/>
13. MobileNet version 2 (2018) <http://machinethink.net/blog/mobilenet-v2/>

Spectral Efficiency Analysis of Massive MIMO



Shubham Mittal, Anuj Singal, Kuldeep Singh, and Manisha Jangra

Abstract A M-MIMO is an optimistic method to magnify spectral efficiency (SE) of cell support system, by sending antenna patterns with many active components at base stations and performing logical transceiver developing. A standard concept of thumb is that information systems ought to be a commission of magnitude many antennas F , then regular users U , because the user's channels are believed to be nearly perpendicular when $F = U > 10$. But, it doesn't demonstrate this concept of thumb really increases the SE. In the study of this research paper, we look at how much the optimal range of daily users, U Depending on the configuration of F and other information systems. End of latest SE expressions can be determined that enable efficient information system regularly investigation with power order, absolute pilot reuse and irregular user position. The precious value U in broad-F organization is derivative in not open form, while dissimulations are utilized to sign what go over at limited F , in various interference scenes, as well as various pilot reuse parameters and to various development strategy. Pilots can account for up to half of the continuity block and the ideal $F = U$ is less than 10 at different points in time that are realistically relevant. Absorbingly Depends potent at processing interact and so it is below the belt to similitude various plot the same U .

Keywords CSI—Channel state information · BS—Base stations · RF—Radio frequency · AP—Access points · SNR—Signal to noise ratio · F —No. of antennas · U —No. of users · M-MIMO—Massive multiple input-multiple output · SU-MIMO- point-to-point MIMO

1 Introduction

Need for wireless turnout, all mobile and fixed, will constantly arise. One we know that, in several years, increasing the actuality mobile in big areas will desire to transfer and accept video slightly regularly, say 100 mb/s/user in every point. Wireless links

S. Mittal (✉) · A. Singal · K. Singh · M. Jangra

Department of Electronics and Communication Engineering, GJUS&T, Hisar, Haryana 125001, India

are used to communicate the data from source to objective over a wide area with no physical link. The data is transferred via wireless channels whenever and anywhere [1].

The significant applications of wireless links are smartphones, radio receivers, networking with wireless standards and telecasts/broadcasts. During the transfer of the Data in air, noise is computed to the data. Because of the computing noise, the data strength is shortened, which results in loss of data. The most significant goal of the wireless communication role is to reduce the error and to overwork high data rate and system limit.

The day-to-day usability of higher data rates for wireless applications in the market is a fundamental issue. The easily manageable solution for wireless signals to achieve the optimistic data rate may be obtained through MIMO system architecture. The basic system architecture is depicted in Fig. 1. MIMO technology is unit of wireless advancement and it is used to transmit and receive high data rate signals for next generation communication [2].

M-MIMO innovation has the potential to meet these ever-increasing demands. The base stations (BS) or access points (AP) in the M-MIMO systems have many more antennas as compared with 4G systems. The M-MIMO is a key element of the 5G standard. The 5G systems and future generations of systems will have the capacity to serve various user simultaneously in the exact identical time-frequency resource.

The wireless channel is time-variant and the channel matrix are always varying by the channel characteristics, there is a necessity to analyze the channel parameters and design an efficient channel estimation model to diminish the mean square error compared to the existing optimized intelligent channel estimation methods [3, 4].

Some of the advantages of an M-MIMO system:

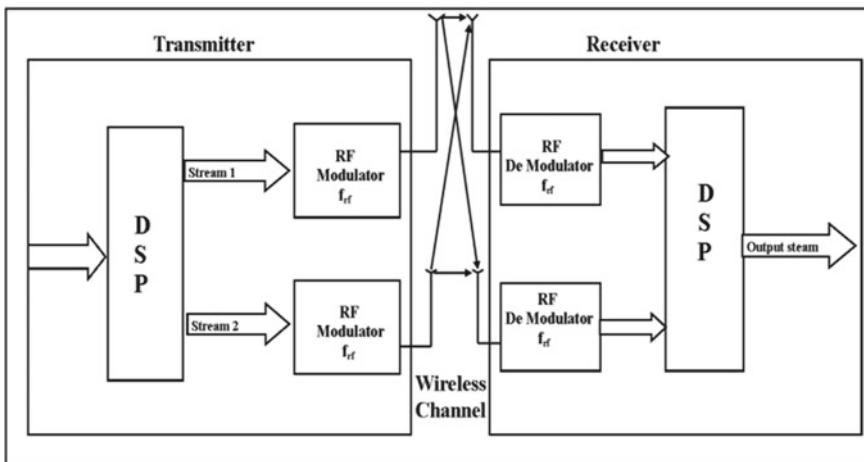


Fig. 1 M-MIMO system architecture

- a. **Capacity increase:** Forceful spatial multiplexing.
- b. **Accretion data rate:** More than free data streams.
- c. **Enhanced reliability:** Limited channels are coded.
- d. **Improved energy efficiency:** BS can focus on its radiated energy into ordered directions.
- e. Lower transmitted power.
- f. Overall power allotment depends upon number of sending Antennas.

1.1 Classification of MIMO

The MIMO technology is sensibly ordered among three primary classes like:

- a. SU-MIMO
- b. Multiuser MIMO (MU-MIMO)
- c. M-MIMO

SU- MIMO

SU- MIMO is the least complicated form of MIMO. Figure 2, depicts a SU- MIMO where AP that has been equipped with an antenna array functions as an MS or an STA. Usually, these are called terminals and they are fitted out with an antenna array. Varying terminals are multiplexed orthogonally. For instance, it can be done by combining TDM and FDM. It is unscalable since the necessary pilot overhead and on the grounds that LoS propagation yields channels of not up to the position. If the transmitter obtains a CSI, there can be an improvement in the performance [3–6].

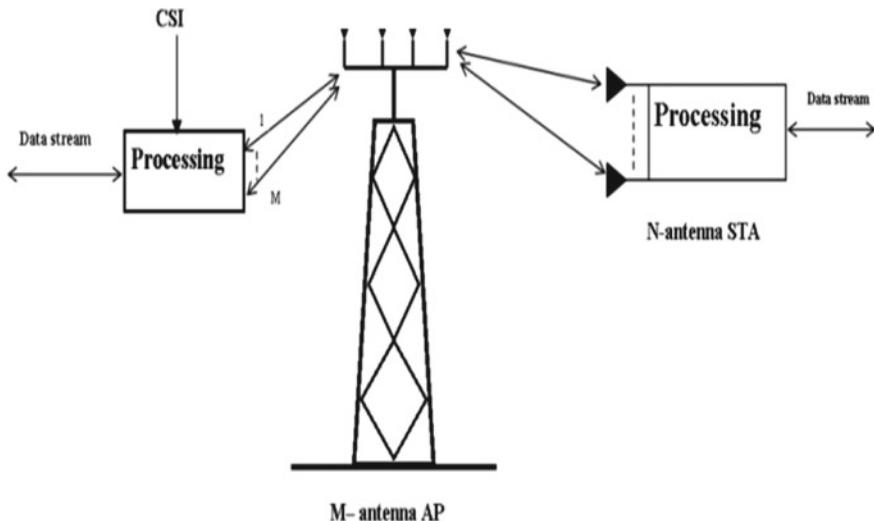


Fig. 2 SU MIMO

Multuser MIMO (MU-MIMO)

The concept behind MU-MIMO is that single BS serves multiple MSs or STAs utilizing identical time-frequency resources as can be seen in Fig. 3. The MU-MIMO in effect is derived towards the SU-MIMO arrangement where the U antenna terminal is broken up into many autonomous terminals. Terminals are spatially multiplexed. Every terminal might be one antenna. The necessities on engendering channel are significantly relaxed as contrasted with the SU-MIMO. In any case, promising signal processing muddled and exact two-way CSI is needed, which in requests that high number assets be dedicated to pilots [7–10].

Practically speaking, there are three restrictions on the functionality of the SU-MIMO namely, (a) the equipment of the terminal is complicated and requires independent RF chains or antennas along with digital processing that is advanced to isolate the information streams, (b) An environment for propagation needs to be able to support a minimum of (M, K) independent streams and (c) As a result of high path loss, the SNR is low close to the near end of the cell structure.

M-MIMO

It is the advanced version of MU-MIMO. The M-MIMO addresses a total separation from the customary Multiuser MIMO. The M-MIMO, a definitive type of Multiuser MIMO. Figure 4 stands apart from the customary Multiuser MIMO severally. In the first instance, CSI is the basic requisition for base station. Channel solidification helps in channel assessment of terminals requirements. The use of TDD mode with reciprocally of channel helps in resources calculation for pilots U . This delivers the M-MIMO completely versatile as for the quantity of base station antennas, F . At this point When F is large, direct handling at the base station is almost ideal [11–14].

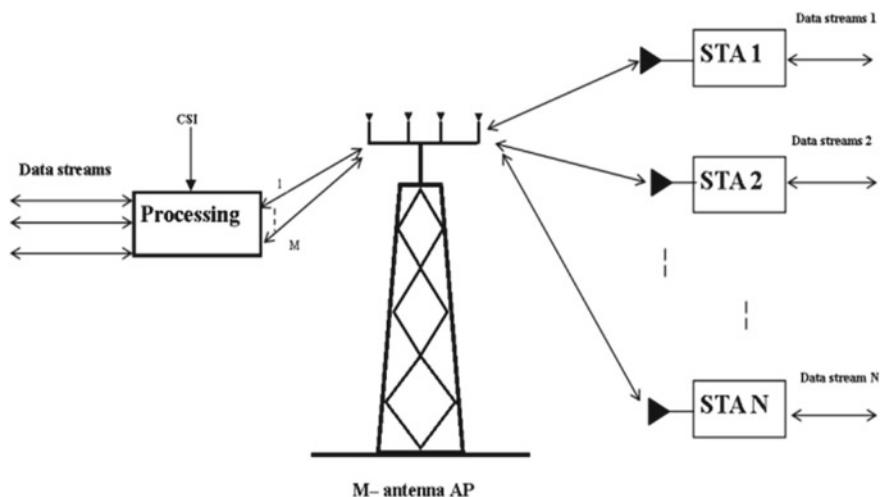


Fig. 3 Multiuser MIMO (MU-MIMO)

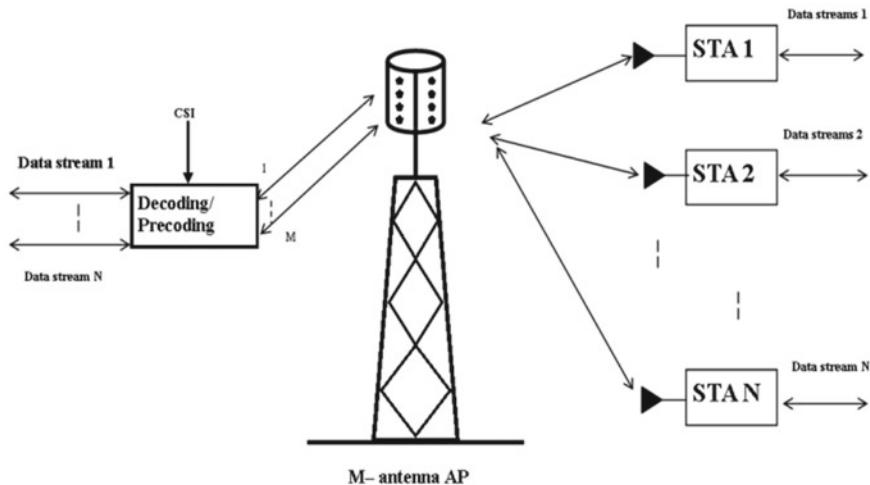


Fig. 4 M- MIMO

1.2 M-MIMO: A Versatile Technology

M-MIMO overcomes the versatility barrier by avoiding the absolute Shannon limit and perplexingly, by increasing the scale of the set-up. It thrice departs from Shannon-hypothetical practise. To begin with, data from the downlink channel was only received by the base station. The time it takes to get CSI in a TDD device is independent of the number of base station antennas. Second, the quantity of base station antennas is regularly expanded to many times the number of users. Finally, the downlink a simple precoding multiplexing is used, along with decoding de-multiplexing on the uplink. Precoding and decoding execution will shift closer to the Shannon limit as the number of base station antennas grows [15].

Problem Statement

Today, the research on improving spectral efficiency is increasing day by day in the academic M-MIMO system. The reason for broadband communication is to give dependable and higher information rates over a wide area. Because of finite spectrum is complex and costly to build data transmission. So, MIMO is proposed for taking out impedance and to improve connect reliability without extra bandwidth efficiencies. The principal commitment of this paper is summarized below:

1. To Analyze & Maximize the spectral Efficiency.
2. Increase in the users and Pilots should be allocated in the different areas.
3. Power control.

Power Control

As shown in a previous area, power control, the pair of uplink and downlink, involves spread the first arrangement of U QAM by power control coefficients,

$$\text{Downlink : } q_U \rightarrow \sqrt{n_U} q_U = 1 \quad U = 1 \dots U, \sum_{U=1}^U n_U \leq 1$$

$$\text{Uplink : } q_U \rightarrow \sqrt{n_U} q_U, n_U \leq 1, \quad U = 1, \dots U.$$

Out of pleasant properties of M-MIMO is many antennas makes the beam forming gains essentially consistent over frequency and besides, relative only large-scale fading coefficients, which is free of bandwidth and index of antenna. Consequently, the coefficients of power control may be separated from bandwidth and its impact on the information rate accomplished by a specific user might be calculated without considering the pilots' momentary channel estimates. The power control functions join the articulation in the numerical expressions for the valid SINRs of precoded/decoded signals in such a way that gap specifications of SINRs are comparable linear imbalance conditions on power control functions.

So, lots of linear equations for the power control functions are solved to get peak power control, which gives everyone in the cell equal throughput. Various systems are possible. Different systems are conceivable. For, e.g. one could determine wanted data for subset users and subject is imperatives utilize linear programming identify the power control coefficients gives the leftover users max-min throughput.

Approach: M-MIMO System with Antenna preference

In this paragraph, a sender and recipient schematic representation of M-MIMO system with antenna preference utilizing nT send and nR get receiving antenna is appeared in Figs. 2 and 3 separately. In the transmitter part, first and for most information is prepared by the M-MIMO system to utilize the T_x antenna preference strategies. In send and get antenna preference part, RF exchanging unit assists with choosing the quantity of RF chains. At the end, M-MIMO processing is utilized to retake the send information as appeared in Fig. 2. As we realize that the MIMO is utilized to build limited system, these different radio frequency (RF) chains build the system intricacy, cost and power used. Along these lines, antenna preference is utilized to beat this issue at sender and beneficiary side by choosing the subset of best send and get receiving antenna among the accessible antenna [16] (Figs. 5 and 6).

Case Work: Thick Macro-cell M- MIMO

We'll look at a cluster situation in which every cell, contains a normal 10 randomly founded users, are served as a base station by 64-cell cluster. The cell orbit are 1000 m, the transporter bandwidth is 1.91 GHz and the spectral frequency is 24 MHz. The 64-cell cluster, containing $\frac{1}{2}$ frequency fix antennas organized in cylindrical arrangement, approx. Diameter is 41 and 56 cm high, as seen in Fig. 1. A constant

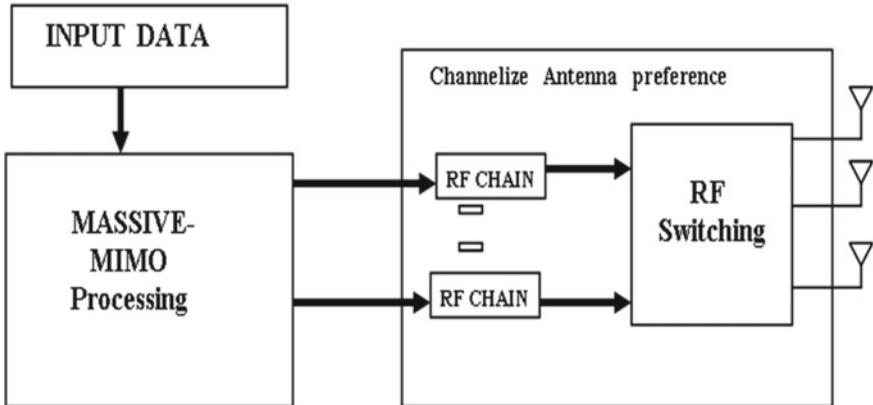


Fig. 5 M- MIMO transmitter with antenna preference

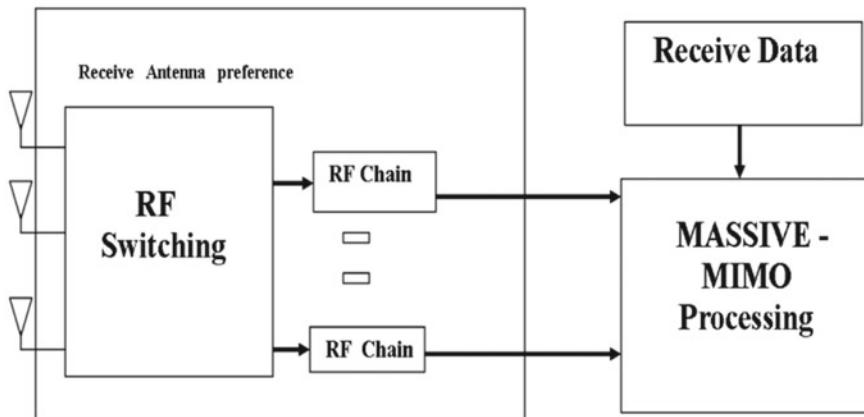


Fig. 6 M-MIMO receiver with antenna preference

myth easily discredited 12Vis that M-MIMO isn't down to earth at anything other than millimetre wavelengths in view of the supposedly actual size of the exhibit. The maximum absolute downlink emanates done watt power of base station and the most extreme uplink transmitted power of every point 200 mW.

Preparing, Pilot Contamination and Space Structure

TDD opening span is 1.82 ms, allowing 7.1×10^4 m/h versatility and indicated by the OFDM specification examine depart of M-MIMO channel assessment, the term of space is $T \frac{1}{4} 392$. Grouping of Pilot length 18 would enough guaranteed symmetry inside every cell, except reutilization of similar 18 pilots starting with one cell to another leads to pilot exposure [17, 18]. Inferring the pilot-inferred estimation only for channel between base station and Channels between users and base station in

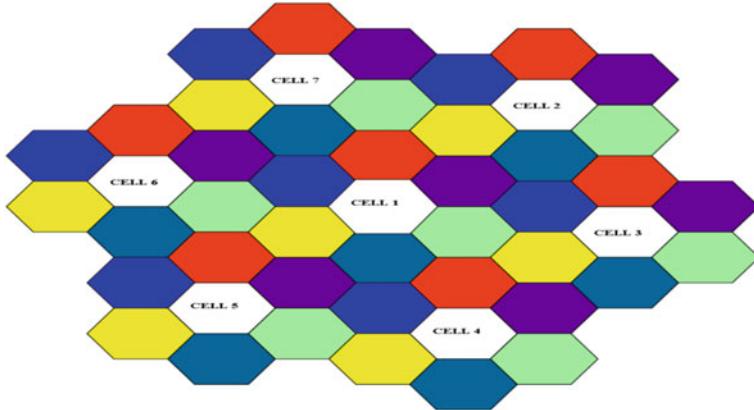


Fig. 7 Pilot re-use seven cells

separate cells is called pilot reutilization who is in the identical pilot groups debase one of its users. The base station sends coherent impedance to users in separate cells on the downlink. This disability doesn't improve when more antennas are added.

Mostly on uplink, an identical impact occurs. Using pilot sequences with a length of $7 \times 18 = 126$, which start a group of seven cells and provide mutually orthogonal pilot protocols to all users, is an easy way to reduce pilot interference. Cell is made up of 2 concentric rings of non-contaminating cells, as shown in Fig. 7. Six cells in the middle ring cause pilot pollution, but the coherent interference is reduced by around 40 dB [19–21]. $\frac{126}{392} = 0.32$ is the percentage of time spent on training. The remainder of the available uplink and downlink data transmission slots is equally distributed.

1.3 Results and Discussions

This paper investigates the efficiency of an uplink-downlink M-MIMO method using MR, P-ZF and ZF MMSE precoding/decoding schemes for channel modelling at Fig. 8. It's also worth noting that the SE's empirical lower bound closely matches the simulation performance as in Figs. 9, 10 and 11: Respectively with Mat lab Tool. When comparing the five scheduling Kappa's, kappa = 3.5 performs the best the others.

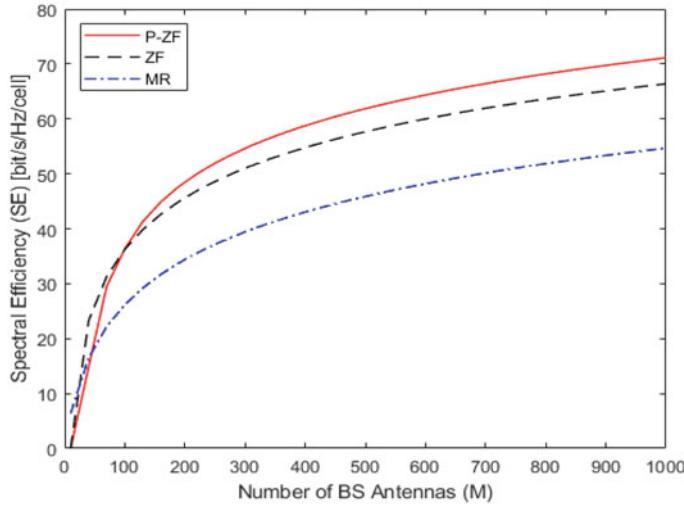


Fig. 8 PZF, ZF, MR values per-cell SEU = 10

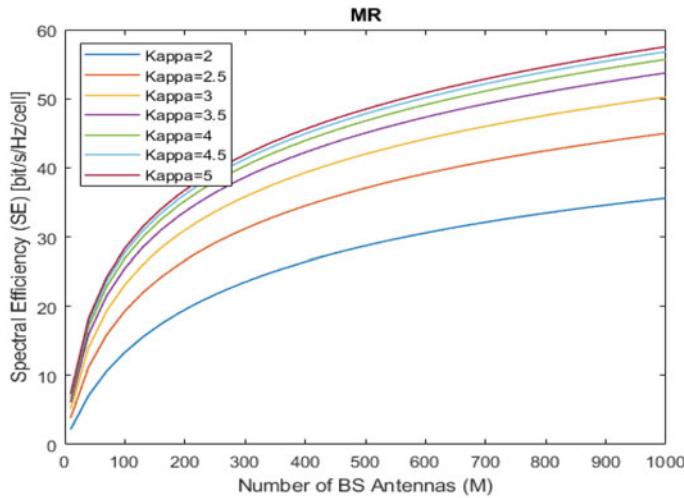


Fig. 9 MR values at different kappa per-cell SE U = 10

2 Conclusion

The efficiency of an uplink-downlink M-MIMO system with MR, P-ZF and ZF MMSE precoding/decoding schemes of channel modelling is examined in this paper. With perfect CSI, some asymptotic uplink-downlink expressions for the SE are simulated and analyzed. We also note that the SE's analytical lower bound matches the

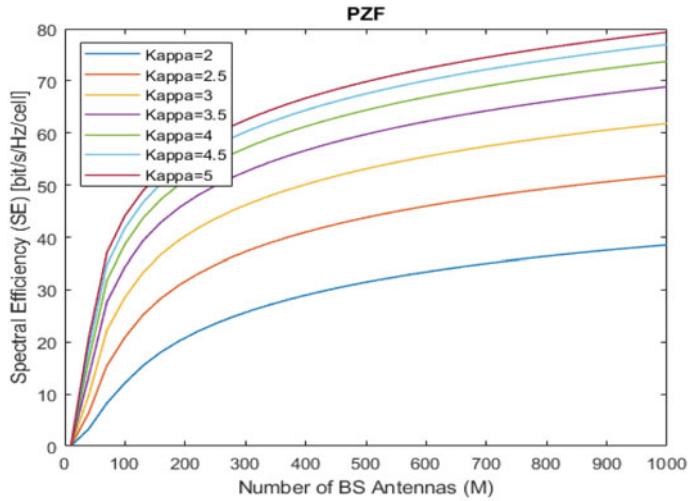


Fig. 10 P-ZF values at different kappa per-cell SE $U = 10$

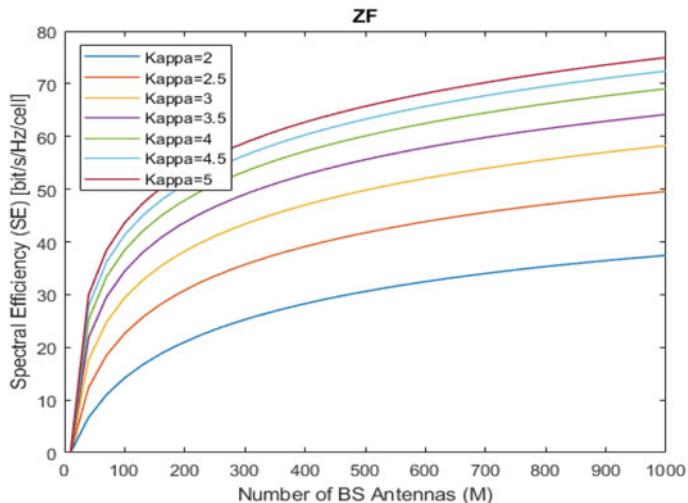


Fig. 11 ZF values at different kappa per-cell SE $U = 10$

simulation results very closely. Among the five scheduling Kappa tested, kappa = 3.5 consistently performs far superior to the others.

Acknowledgements I would like to thank Mr. Kuldeep Singh, Mr. Anuj Singal and Mrs. Manisha Jangra from GJUS&T, Hisar (125001) for her generous help in improving this paper.

References

1. Foschini G, Gans M (1998) On limits of wireless communications in a fading environment when using multiple antennas. *Wirel Pers Commun* 6:311–335. <https://doi.org/10.1023/A:100889222784>
2. Larsson EG, Edfors O, Tufvesson F, Marzetta TL (2014) Massive MIMO for next generation wireless systems. *IEEE Commun Mag* 52(2):186–195. <https://doi.org/10.1109/MCOM.2014.6736761>
3. Raleigh GG, Cioffi JM (1998) Spatio-temporal coding for wireless communication. *IEEE Trans Commun* 46(3):357–366. <https://doi.org/10.1109/26.662641>
4. Paulraj A, Kailath T (1994) Increasing capacity in wireless broadcast systems using distributed transmission/directional reception (DTDR). U.S. Patent 5 345 599, Sep 6, 1994
5. Gesbert D, Kountouris M, Heath Jr RW, Chae C, Chae T (2007) Shifting the MIMO paradigm. *IEEE Sig Process Mag* 24(5):36–46
6. Caire G, Shamai S (2003) On the achievable throughput of a multi antenna Gaussian broadcast channel. *IEEE Trans Inf Theory* 49:1691–1706. <https://doi.org/10.1109/TIT.2003.813523>
7. Viswanath P, Tse DNC (2003) Sum capacity of the vector Gaussian broadcast channel and uplink–downlink duality. *IEEE Trans Inf Theory* 49(8):1912–1921. <https://doi.org/10.1109/TIT.2003.814483>
8. Vishwanath S, Jindal N, Goldsmith A (2003) Duality, achievable rates, and sum-rate capacity of Gaussian MIMO broadcast channels. *IEEE Trans Inf Theory* 49(10):2658–2668. <https://doi.org/10.1109/TIT.2003.817421>
9. Marzetta TL (2006) How much training is required for multiuser Mimo? In: 2006 Fortieth asilomar conference on signals, systems and computers, pp 359–363. <https://doi.org/10.1109/ACSSC.2006.354768>
10. Rusek F et al (2013) Scaling up MIMO: opportunities and challenges with very large arrays. *IEEE Signal Process Mag* 30(1):40–60. <https://doi.org/10.1109/MSP.2011.2178495>
11. Telatar E (1999) Capacity of multi-antenna Gaussian channels. *Eur Trans Telecomm* 10:585–595. <https://doi.org/10.1002/ett.4460100604>
12. Marzetta TL (2010) Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Trans Wirel Commun* 9(1):3590–3600
13. Li Y, Nam Y-H, Ng BL, Zhang J (2012) A non-asymptotic throughput for massive MIMO cellular uplink with pilot reuse. In: IEEE global communications conference (GLOBECOM), 4500–4504. <https://doi.org/10.1109/GLOCOM.2012.6503827>
14. Müller RR, Cottatellucci L, Vehkaperä M (2014) Blind pilot decontamination. *IEEE J Sel Top Signal Process* 8(5):773–786. <https://doi.org/10.1109/JSTSP.2014.2310053>
15. Yin H, Gesbert D, Filippou M, Liu Y (2013) A coordinated approach to channel estimation in large-scale multiple-antenna systems. *IEEE J Sel Areas Commun* 31(2):264–273. <https://doi.org/10.1109/JSAC.2013.130214>
16. Li M, Jin S, Gao X (2013) Spatial orthogonality-based pilot reuse for multi-cell massive MIMO transmission. In: International conference on wireless communications and signal processing, 1–6. <https://doi.org/10.1109/WCSP.2013.6677139>
17. Karlsson M, Larsson EG (2014) On the operation of massive MIMO with and without transmitter CSI. In: 2014 IEEE 15th international workshop on signal processing advances in wireless communications (SPAWC), pp 1–5. <https://doi.org/10.1109/SPAWC.2014.6941305>
18. Gao X, Edfors O, Rusek F, Tufvesson F (2015) Massive MIMO performance evaluation based on measured propagation data. *IEEE Trans Wirel Commun* 14(7):3899–3911. <https://doi.org/10.1109/TWC.2015.2414413>
19. Guo K, Guo Y, Fodor G, Ascheid G (2014) Uplink power control with MMSE receiver in multi-cell MU-massive-MIMO systems. In: IEEE international conference on communications (ICC), 5184–5190. <https://doi.org/10.1109/ICC.2014.6884144>

20. Yang H, Marzetta TL (2014) A macro cellular wireless network with uniformly high user throughputs. In: 2014 IEEE 80th vehicular technology conference (VTC2014-Fall), pp 1–5. <https://doi.org/10.1109/VTCFall.2014.6965818>
21. Biguesh M, Gershman AB (2004) Downlink channel estimation in cellular systems with antenna arrays at base stations using channel probing with feedback. EURASIP J Adv Signal Process 963649. <https://doi.org/10.1155/S1110865704403023>

Recommendations for DDOS Threats Using Tableau



Sagar Pande, Aditya Kamparia, and Deepak Gupta

Abstract The massive increase in network-based traffic reveals enterprise networks to a vast range of threats. Disruptive traffic is interfering with the daily activity of the network by wasting organizational energy and time. Efficiency is boosted by effective methods for detecting, defending, and minimizing disruptive events. IDS is one of the most important characteristics of network and host protection since it is deployed in the network and at the client hardware level to track suspicious traffic in the network and on specific computers. The information obtained from IDS includes the threats and normal activities which will help in improving the performance of the IDS. The patterns and crucial threats can be identified by analyzing the obtained information. The data is analyzed using the tableau application.

Keywords Intrusion detection system · Visualization · Tableau · Threats · NSL-KDD

1 Introduction

Cybersecurity risks are growing exponentially and in complexity as network technology and information technology develop at a breakneck pace. For example, according to a study conducted by McAfee on the financial consequences of cyber threats released in 2018, fraudulent practices are incredible, about 80 billion fraudulent checks are conducted every day. Besides, according to the 2018 Cybersecurity Contravention Report, 43 percent of high-profile companies have experienced cybersecurity attacks in the previous 12 months globally [1]. Besides that, as per the yearly cybersecurity survey-2017, economical damages from cybersecurity operations are expected to cost about \$6 trillion a year by 2021 [2]. Because of the massive price,

S. Pande (✉) · A. Kamparia

School of Computer Science and Engineering, Lovely Professional University, Punjab, India

D. Gupta

Department of Computer Science Engineering, Maharaja Agrasen Institute of Technology, New Delhi, India

e-mail: deepakgupta@mait.ac.in

there is a pressing requirement to implement modern cyberattack security mechanisms and technologies. Despite the availability of antivirus tools, firewalls, and intrusion detection systems to identify and defend IT architectures from a range of established cyber threats, malicious hackers have grown more expert at creating new sophisticated and more nuanced strategies to obtain accessibility to and destroy sensitive IT architectures. According to the Security Report made by Cisco in the year 2018, the usage of machine learning methodologies would prepare the path for the creation of cyber protection strategies that can automatically identify any abnormal latest trends in network traffic [3]. Firewalls, for example, have conventionally been contemplated the first line of protection at odds with cyberattacks in certain corporate organizations, yet they struggle to detect threats on authorized resources. Antivirus and intrusion detection systems are needed as a second line of security protection in those kinds of circumstances. Antivirus program, on the other hand, is restricted to protecting the network only against threats whose identifiers are kept in a server. Identifiers are often updated regularly. As a result, in the interval among upgrades, the network is vulnerable to vindictive activity. IDS plays a vital role in the collection of information about malicious threats which will be useful in the identification of various signatures associated with various malicious threats. Analyzing or visualizing the obtained information from IDS will help to understand the patterns that exist in various threats that are followed by various hackers to intrude into the network or server to steal the essential information of various organizations.

The major contributions of the proposed framework deal with

- Analyzing the various datasets such as KDD and NSL-KDD.
- Identifying the importance of NSL-KDD when compared with another mentioned dataset.
- Analyzing the NSL-KDD, thereby generating the recommendations using tableau which will be helpful for further research.

This article was organized into various sections. Section 2 mainly deals with the discussion of related literature information based on IDS and threats. Section 3 mainly deals with the discussion of the dataset, related methodological discussion of the proposed framework. Section 4 mainly discusses the visualization patterns as part of results obtained from the tableau platform that exists in the dataset. Section 5 mainly deals with the discussion of the obtained conclusion from the obtained results.

2 Literature Review

With devastating results, innovative advancements in network technology have sparked the development of sophisticated cyberattacks that outperform conventional security defenses. Intrusion Detection Systems (IDS) have progressively been viewed as a critical component of network security architectures for achieving a strong line of defense against cyber threats. Binbusayyis et al. [5] presented the goal of the

study to suggest the most suitable attribute evaluation measure for creating an effective IDS. In this context, 4 filter-based attribute evaluation measures derived from various concepts such as consistency, correlation, knowledge, and proximity are examined for their possible impacts in improving the IDS framework's identification capacity for various types of threats. Besides, the impact of chosen attributes on IDS framework classification accuracy is investigated using 4 various types of classification algorithms: KNN, RF, SVM, and DBN. Eventually, the study outcomes are subjected to a two-step statistical importance test to ascertain which function assessment measure reflects statistically important differences in IDS efficiency.

Aljawarneh et al. [6] focused on the network information agreement optimal attributes that were made accessible for training, constructs the latest hybrid framework that can be utilized to evaluate the invasion range threshold level. The hybrid framework had an important influence on reducing the numerical and time complexities intricated in evaluating the function interaction effective size, according to the research observations [7]. For the binary categorical class and multi-class NSL-KDD datasets, the suggested framework's accuracy was 99.81 percent and 98.56%, accordingly. Nevertheless, achieving high false-positive and low false-negative rates is problematic. To fix these problems, a hybrid solution with 2 key components is suggested [8]. To continue, details must be processed utilizing the Vote methodology with Knowledge Gain, which integrates the probability distributions of these base learners to pick the essential attributes that have a positive effect on the suggested framework's accuracy. Kasongo et al. [9] mentioned the wireless networks have developed over time to become one of the most widely used networking mediums. At any considered time, these systems are capable of transmitting vast amounts of data. This has resulted in a slew of protection and privacy issues. This article introduces a wireless IDS classifier based on deep LSTM. We equate the DLSTM IDS to current approaches like Deep FFNN, SVM, KNN, RF, and NB using the NSL-KDD dataset. The DLSTM IDS exceeds current methods, according to the findings of the experiments. Zhu et al. [10] proposed a mechanism of evaluating whether network activity has deviated from natural known as network oddity identification. As networks increase in size and complexity, identifying irregular activities in broad energetic networks has become progressively important. However, detecting network anomalies quickly and accurately is difficult. Because of its strong attribute modeling capabilities, deep learning is a promising approach for detecting network anomalies. The FNN and CNN models are used in this article to introduce the latest oddity detection approach based on deep learning frameworks [11]. Many other studies with the famous NSL-KDD dataset are used to test the frameworks' efficiency. The FNN and CNN frameworks have not only a good modeling potential for network outlier identification, but also have high precision, according to the research observations [12]. Deep learning frameworks can enhance both identification accuracy and the potential to distinguish various forms of anomalies. Mahfouz et al. [13] proposed a comparative study based on the identification of various intrusions with the aid of machine learning methodologies. Latest threats emerge as a result of the exponential development of network-based technologies, and various protection protocols require increased effort to increase time and efficiency. Despite the introduction of

various latest security technologies, the exponential growth of fraudulent operations remains a major concern, and the developing assaults pose severe challenges to the security of the network [14]. IDSs are extensively used by network operators to track certain network disruptive behaviors. One of the most popular technologies for intrusion identification is ML, which involves learning frameworks from information to distinguish between suspicious and normal traffic. Despite the widespread utilization of ML methodologies, a thorough examination of ML methodologies in the field of intrusion identification is deficient [15]. The current study is a thorough review of several current ML classifiers for detecting network-based traffic intrusions. This paper examines classification methodologies on many levels, including function selection, vulnerability to hyper-parameter choices and their optimizations, and class disparity issues that are common in intrusion detection. Various classification methodologies are tested utilizing the NSL-KDD dataset, and their usefulness is summarized utilizing a comprehensive investigational assessment. Dey et al. [16] analyzed the performance of the Software-defined networking architecture-based identification of intrusions-based frameworks with the selection of features technologies. This framework includes various ML methodologies such as Naïve Bayes, Decision Trees, Random Forest, Random Basial Function Net, Bayes Network, PART, and J48. The performance of the proposed framework was evaluated through various evaluation metrics such as accuracy, precision, recall, F1-score, false alarm rate, and Mathews correlation coefficient. The massive increase in Internet-based traffic introduces enterprise networks to a vast range of threats [17]. Disruptive traffic is interfering with the daily activity of the network by wasting organizational time and resources. Efficiency is boosted by effective methods for detecting, defending, and minimizing disruptive events [18]. IDS is one of the most important characteristics of network and host protection since it is deployed in the network and at the client operation level to track suspicious traffic in the network and on specific computers. Outlier traffic analysis methods that aren't monitored are getting better all the time. Yihunie et al. [19] attempt to study with 5 ML techniques to discover an effective classification methodology that identifies outlier traffic from the NSL-KDD dataset with good accuracy and low fault rate. To generate the output, 5 binary classification methodologies are assessed: SGD, RF, LR, SVM, and Sequential Frameworks. The results show that the RF classification methodologies outperform the other 4 classification methodologies both with and without the dataset being normalized.

3 Methodology

Researchers also implemented many network traffic datasets to aid in the evaluation of various intrusion Identification techniques. There are three types of datasets. They are public datasets, private datasets, and network replicating datasets. The majority of such databases are created utilizing a variety of methodologies that assisted in the capture of traffic, the initiation of various forms of threats, and the surveillance of

traffic designs. We utilize the NSL-KDD dataset in this article, which is among the most widely used specification datasets in the intrusion identification field.

The NSL-KDD dataset [20] is an improved variant of the completely familiar KDDcup99 dataset that can be used outdoors. Most of the researchers have conducted various kinds of analyses on the NSL-KDD and developed successful IDSs using various methodologies and techniques [21]. There are 41 features in the NSL-KDD dataset along with 1 class feature. Dhanabal and Shantharajah [22] includes a detailed description of these characteristics. We provide a numerical overview of the NSL-KDD dataset in the format described above. The whole dataset consists of 4 classes and they are namely DoS, Probe, U2R, R2L. There are again sub-classes for each of the classes. DoS consists of 11 sub-classes, probe consists of 6 sub-classes, U2R consists of 7 sub-classes, and R2L consists of 15 sub-classes. This dataset also consists of one more class called Normal. The dataset distribution can be represented as mentioned in Table 1. The comparison between the NSL-KDD dataset and the KDD dataset is represented in Table 2.

The NSL-KDD dataset will be having the training dataset and testing dataset by default. Training and testing datasets are combined to form a total dataset. Total instances or records in the dataset are 1,48, and 517. The total instances are distributed according to the classes and the percentage of instances or records under each of the classes are normal—51.88%, DoS—35.95%, Probe—9.48%, U2R—0.17%, and R2L—2.46%.

As per the definition of Intrusion provided by the National Institute of Standards and Technology as an attempt of breaking the three essentials of the network such as confidentiality, integrity, and availability. Intrusion Detection Systems are viewed as a vital tool for protecting IT networks from attacks and improving network security. These essentials are usually referred to as the CIA triad [4]. As a result, for about

Table 1 Distribution of various classes in the NSL-KDD dataset

Dataset	Total number of records as per the sections					
	Total	Normal	DOS threats	Probe threats	U2R threats	R2L threats
Training dataset	125,973	67,343	45,927	11,656	52	995
Testing dataset	22,544	9711	7458	2421	200	2654
Total dataset	148,517	77,054 (51.88%)	53,385 (35.95%)	14,077 (9.48%)	252 (0.17%)	3649 (2.46%)

Table 2 Comparison between NSL-KDD and KDD datasets

Dataset	No. of instances/records		
	Total	Normal	DDOS
KDD	488,735	97,277	391,458
NSL-KDD	113,270	67,343	45,927

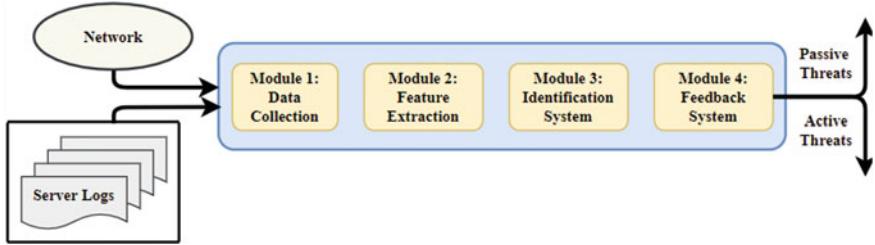


Fig. 1 Generalized architecture of intrusion detection system

all organizations, Intrusion identification is the practice of continuously tracking and observing incidents in a software device or network for evidence of interference.

As a consequence, the Intrusion Detection System is security surveilling system that identifies suspicious behaviors inside the digital devices or server infrastructure and flags those that breach the CIA's information security standards. They can identify malicious activity from both adversaries and network infrastructure participants. Data source, selection of attributes, Identification system, and Feedback are the 4 main constituents of an IDS. These 4 constituents work together to detect threats and outline the outcomes in a necessary form. Data source, selection of attributes, Identification system, and Feedback are the 4 main constituents of an Information Detection System. These 4 constituents work together to detect threats and outline the outcomes in a necessary form. The generalized structure of the Intrusion Detection System can be represented as mentioned in Fig. 1.

Data Collection: This phase is in charge of obtaining intrusion evidential information from appropriate sources and distributing it to the entire system in a detailed manner. Trying to collect this data is costly, and the most difficult aspect is gathering the appropriate data.

Selection of Features: This phase must hold only the most useful attributes for threat classification while discarding the unnecessary features. Eventually, it constructs a function vector from a subclass of functions. This project will present the latest methodology for the extraction of various features to improve the identification efficiency of the system.

Identification System: This phase is the most important part of an Intrusion Detection System and is in charge of processing information to identify intruder behavior. The capacity of this part to identify all forms of threats also decides the ultimate power of the Intrusion Detection System.

Feedback System: When the identification system detects a threat, it is in charge of deciding how to adjust and controlling the feedback approach. This part chooses between passive threats feedback which merely activates an alarm without taking any cause against the origin and active threats feedback which includes disabling the origin for a specified period. The type of feedback action to be taken is determined by the security policies of an organization.

4 Results and Discussion

The NSL-KDD dataset was analyzed utilizing the Tableau application. First, as seen in Fig. 2, I attempted to evaluate the scattering of different classes in the NSL-KDD dataset. The Neptune class, with a percentage of 64.53%, is the most common class in the dataset, followed by regular class, smurf class, satan class, portsweep class, saint class, teardrop class, and so on, with corresponding percentages of 15.19%, 9.63%, 8.29%, 0.80%, 0.68%, 0.44%, and 0.44%. The remaining groups are essentially unimportant. As seen in Fig. 3, the next study is to evaluate the scattering of different

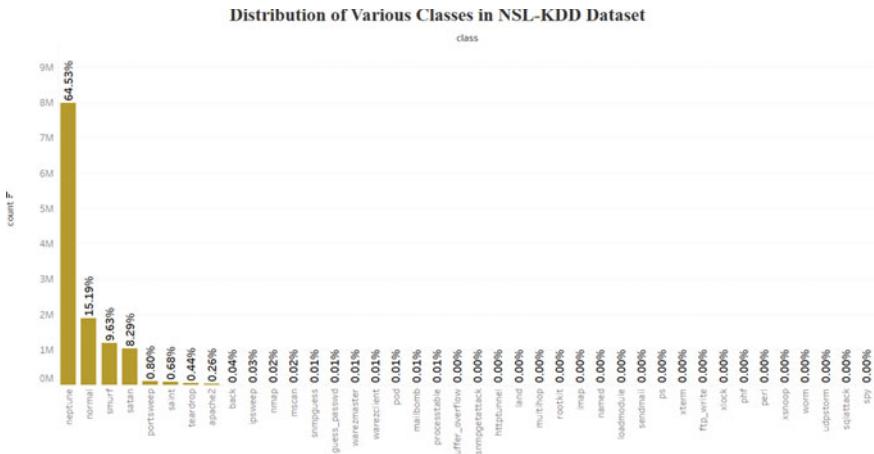


Fig. 2 Representation of scattering of various classes in NSL-KDD

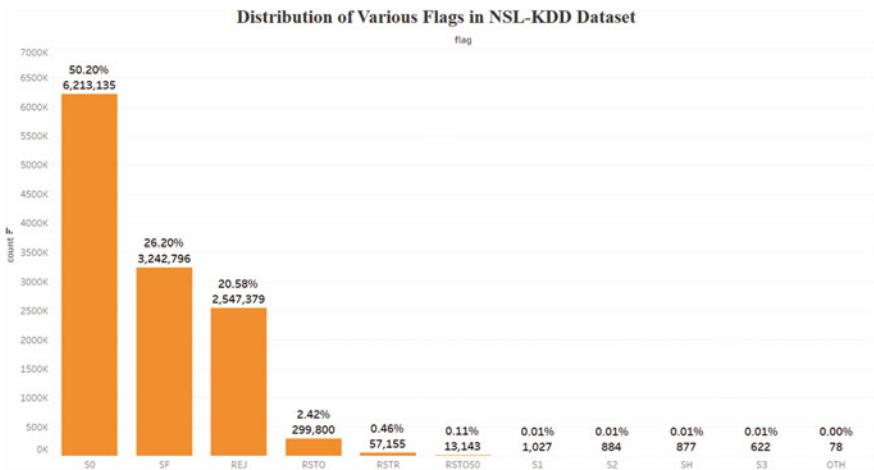
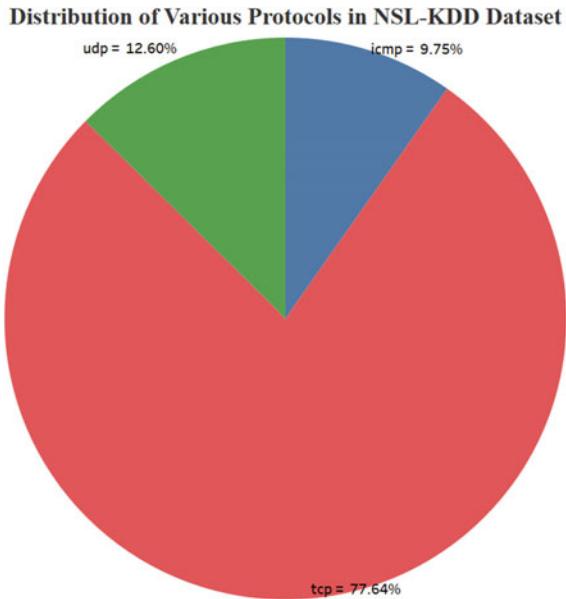


Fig. 3 Representation of scattering of various flags in NSL-KDD

Fig. 4 Representation of scattering of various protocols in NSL-KDD



flags in the NSL-KDD dataset. S0, SF, REJ, and RSTO make up the majority of the flags in the dataset, contributing 50.20%, 26.20%, 20.58%, and 2.42%, respectively. The remaining flags have a negligible impact on the dataset in use. The NSL-KDD dataset was then analyzed further to determine the distribution of different protocols, as seen in Fig. 4. The tcp protocol makes a significant contribution to the used dataset, as can be seen in this graph. To the used dataset, tcp contributes about 77.64%, icmp contributes about 9.75%, and udp contributes about 12.60%. Then, as seen in Fig. 5, the study of the NSL-KDD dataset was generalized to describe the delivery of different resources. The important resources in the NSL-KDD are private, ecr_i, domain u, other, and http, as seen in this graph. The private server, the ecr_i service, the domain u service, the other service, and the http service all add 30.00%, 9.65%, 8.67%, 6.91%, and 4.93% to the dataset, respectively. In the dataset, the remaining resources are irrelevant.

Then, as seen in Fig. 6, the study of the NSL-KDD dataset was generalized to define the distribution of different groups surrounding a protocol in the NSL-KDD dataset. The Neptune class is the most important class in the tcp protocol, accounting for 64.53% of the total. The icmp protocol's most important class, with a contribution of 9.63%, is the smurf class. The standard class is the most important class in the udp protocol, accounting for 11.40% of the total. Then, as seen in Fig. 7, the study of the NSL-KDD dataset was generalized to describe the distribution of different facilities surrounding a protocol in the NSL-KDD dataset. The private service, which contributes 27.28% to the tcp protocol, is the most relevant service. The ecr_i service, which contributes 9.65% to the icmp protocol, is the most important service. The domain u service, which contributes 8.67% to the udp protocol, is the most relevant

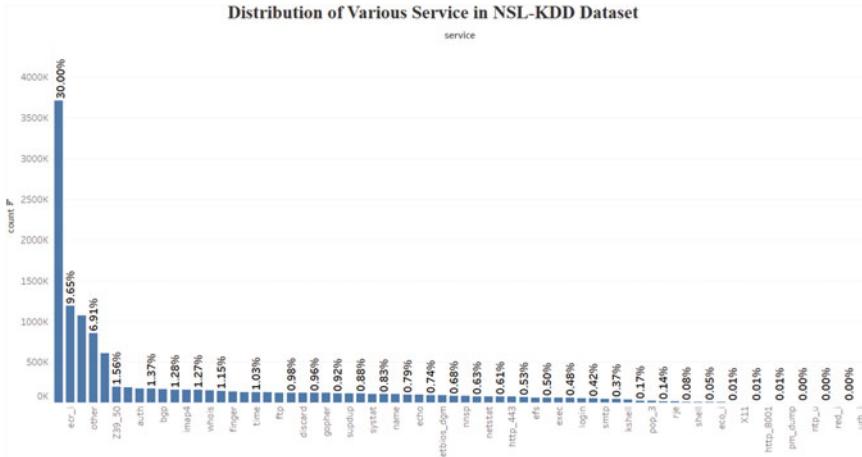


Fig. 5 Representation of scattering of various service in NSL-KDD

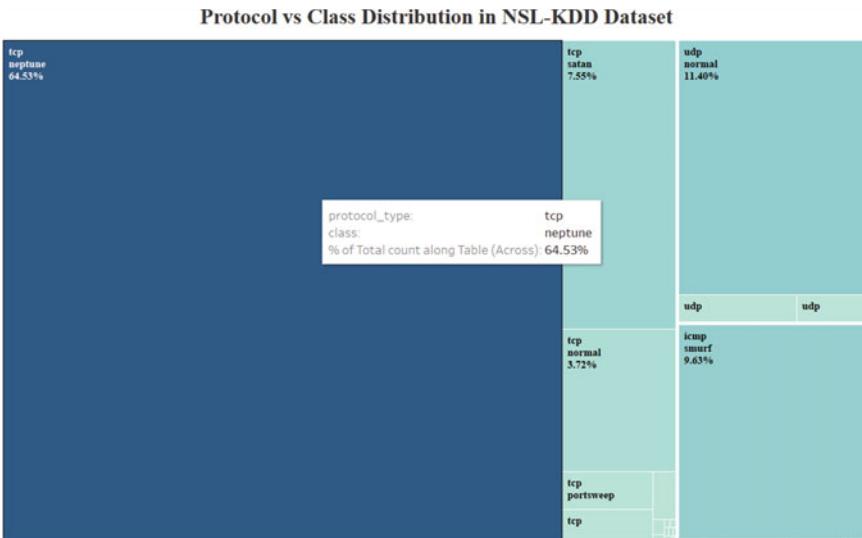


Fig. 6 Representation of scattering of various classes concerning protocol in the NSL-KDD

service. Then, as seen in Fig. 8, the study of the NSL-KDD dataset was generalized to classify the distribution of different resources surrounding a class in the NSL-KDD dataset. The private service, which contributes 24.45% to the Neptune class protocol, is the most relevant service. The domain u service, which adds 8.67% to the regular class, is the most appropriate service. The ecr_i service, which contributes 9.63% to the smurf class, is the most appropriate service. The other service, which contributes 5.28% to the satan class, is the most important service.

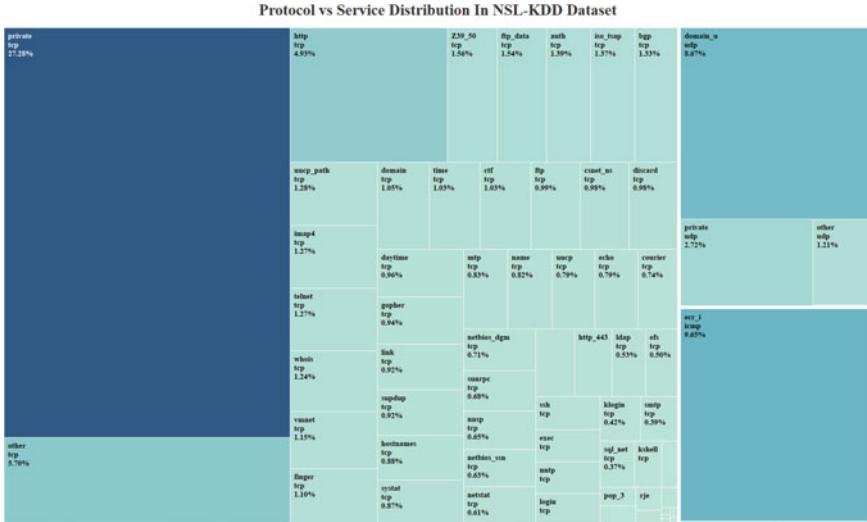


Fig. 7 Representation of scattering of various services concerning protocol in the NSL-KDD dataset

Class vs Service Distribution in NSL-KDD Dataset

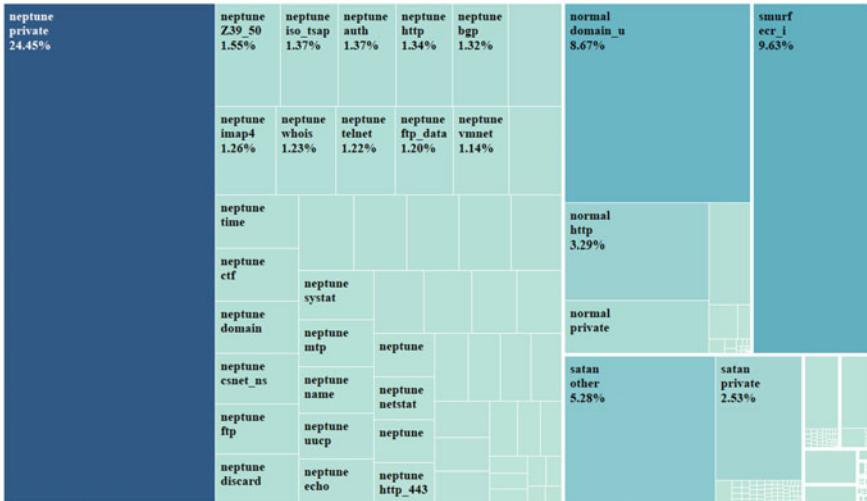


Fig. 8 Representation of scattering of various services concerning class in the NSL-KDD

5 Conclusion

The information obtained from the intrusion detection system will provide information about the various threats and normal activities that influence the network or

server or the physical systems of the organizations. Obtaining the information is not just enough but also essential for identifying the hidden patterns inside the information. The NSL-KDD is the popular dataset that provides such information. The present research is able to study those patterns that are hidden within the information. Identifying those patterns and the popular threats helpful in the generation of a more appropriate Intrusion Detection System would help in identifying the threats in an appropriate duration. The analysis of the information obtained from IDS will also help in improving the performance of IDS.

References

1. Vaidya R (2018) Cyber security breaches survey 2018: statistical release, technical report. Department for Digital, Culture, Media and Sport, London
2. Morgan S (2017) Cybercrime report, technical report, cybersecurity ventures
3. Annual Cybersecurity Report, Executive Summary, Cisco (2018)
4. Bace R, Mell P (2001) NIST special publication on intrusion detection systems, technical Report. Booz-Allen and Hamilton Inc., McLean, VA
5. Binbusayyi A, Vaiyapuri T (2020) Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection. *Heliyon* 6(7):e04262
6. Aljawarneh S, Aldwairi M, Yassein MB (2018) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J Comput Sci* 25:152–160
7. Lian W, Nie G, Jia B, Shi D, Fan Q, Liang Y (2020) An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning. *Math Prob Eng* 2020
8. Varghese JE, Muniyal B (2018) A comparative analysis of different soft computing techniques for the intrusion detection system. In: International symposium on security in computing and communication. Springer, Singapore, pp 563–577
9. Kasongo SM, Sun Y (2020) A deep long short-term memory based classifier for the wireless intrusion detection system. *ICT Express* 6(2):98–103
10. Zhu M, Ye K, Xu C-Z (2018) Network anomaly detection and identification based on deep learning methods. In: International conference on cloud computing. Springer, Cham, pp 219–234
11. Li Y, Xu Y, Liu Z, Hou H, Zheng Y, Xin Y, Zhao Y, Cui L (2020) Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Measurement* 154:107450
12. Boutaba R, Salahuddin MA, Limam N, Ayoubi S, Shahriar N, Estrada-Solano F, Caicedo OM (2018) A comprehensive survey on machine learning for networking: evolution, applications, and research opportunities. *J Internet Serv Appl* 9(1):1–99
13. Mahfouz AM, Venugopal D, Shiva SG (2020) Comparative analysis of ML classifiers for network intrusion detection. In: Fourth international congress on information and communication technology. Springer, Singapore, pp 193–207
14. Zhao H, Li M, Zhao H (2020) Artificial intelligence-based ensemble approach for intrusion detection systems. *J Vis Commun Image Representation* 71:102736
15. Paulauskas N, Baskys A (2019) Application of histogram-based outlier scores to detect computer network anomalies. *Electronics* 8(11):1251
16. Dey SK, Raihan Uddin M, Mahbubur Rahman M (2020) Performance analysis of SDN-based intrusion detection model with feature selection approach. In: Proceedings of the international joint conference on computational intelligence. Springer, Singapore, pp 483–494
17. Wu, Peilun, Hui Guo, and Richard Buckland. “A transfer learning approach for network intrusion detection.” In 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA), pp. 281–285. IEEE, 2019.

18. Mulay SA, Devale PR, Garje GV (2010) Intrusion detection system using support vector machine and decision tree. *Int J Comput Appl* 3(3):40–43
19. Yihunie F, Abdelfattah E, Regmi A (2019) Applying machine learning to anomaly-based intrusion detection systems. In: 2019 IEEE Long Island Systems, applications, and technology conference (LISAT). IEEE, pp 1–5
20. NSL-KDD dataset [online] is available: <http://www.unb.ca/cic/datasets/nsl.html>. Accessed on 21 Oct 2018
21. Ingre B, Yadav A (2015) Performance analysis of NSL-KDD dataset using ANN. In: International conference on signal processing and communication engineering systems (SPACES). IEEE
22. Dhanabal L, Shanthalrajah SP (2015) A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *Int J Adv Res Comput Commun Eng* 4(6):446–452

Sinkhole Attack Detection in Wireless Sensor Networks



Aina Mehta, Jasminder Kaur Sandhu, Meena Pundir, Rajwinder Kaur, and Luxmi Sapra

Abstract Wireless Sensor Networks (WSNs) is a group of spatially deployed multi-functional sensor nodes which communicates data in a short range of network so that sensed data can be shared with other nearby nodes. These networks possess limited resources such as less memory, limited energy, and low communication resources due to which security becomes the critical problem. Moreover, they are deployed in a dynamic environment because of which they get susceptible to many Denial of Service (DoS) attacks such as blackhole, wormhole, and sinkhole attacks. A sinkhole attack is the most threatening routing attack of the network layer which sends fake information assuming it is the shortest path to the base station so that entire network traffic gets attracted toward it. The sinkhole node which is created possesses many issues in the network such as tampering with the data, modifying, altering, and damaging the entire structure. So, it becomes crucial to understand existing detection approaches against this threatening attack. Further, an overview of the rule-based, anomaly-based techniques, statistical and hybrid is presented. Moreover, it also talks about the detailed examples of an anomaly-based approach that includes message digest, zone-based, game theory, emotional ants' algorithms. Lastly, it highlights challenges and provides a future perspective in detecting sinkhole attacks.

Keywords Wireless sensor networks · Denial of Service (DoS) attack · Sinkhole attack · Cryptographic approach · Rule-based approach

1 Introduction

Wireless Sensor Networks (WSNs) have gained wide popularity due to their research results and emerging real-world applications. Some of these applications include

A. Mehta (✉) · J. K. Sandhu · M. Pundir · R. Kaur
Chitkara University Institute of Engineering & Technology, Punjab, India
e-mail: aina.mehta@chitkara.edu.in

L. Sapra
Dev Bhoomi Institute of Technology, Dehradun, Uttarakhand, India

home automation, vehicle tracking, traffic monitoring, and military surveillance [1]. WSN consists of thousands of tiny, low-cost, multifunctional sensor motes which are spatially distributed and deployed in a particular area [2]. Along with these sensor nodes, these networks also have embedded sensors, processors, and radio with less power consumption which is mainly used for wireless communication with the base station. The basic operation of the sink is to perform activation of sensor nodes, gathering the information and processing, and connecting with the other networks [3]. Sensor nodes in these networks gather the required information from physical surroundings such as humidity, pollution level, and sound. Moreover, this collected data transferred to the sink or base station for processing. After processing the data, it gets transmitted to users through the Internet [4] as shown in Fig. 1. So as discussed, due to its wide applications and as information is transmitted in WSN from sender to receiver it becomes very important to ensure security [5].

Security in WSN is a very challenging task as almost all the sensor nodes are deployed in hostile environments and sometimes it becomes very difficult to keep an eye on all the actuator nodes every time. But as sensitive data is generated in these sensor networks it is important to prevent any kind of attacker from hindering the transmission of information [6]. Moreover, each sensor node has limited resources such as bandwidth, energy, memory, and limited computational power, etc. So, due to these constraints, these networks become susceptible to so many attacks especially DoS attacks [7]. There are various kinds of DoS attacks such as Hello flooding, Jamming, Collision, and Sinkhole attack. Here we will discuss in detail sinkhole attacks. Section 2 discusses a detailed explanation of Sinkhole attacks and an overview of certain detection approaches Sect. 3 explains the anomaly-based approach in detail and Sect. 4 gives major design obstacles in detecting Sinkhole Attacks. Section 5 concludes the paper.

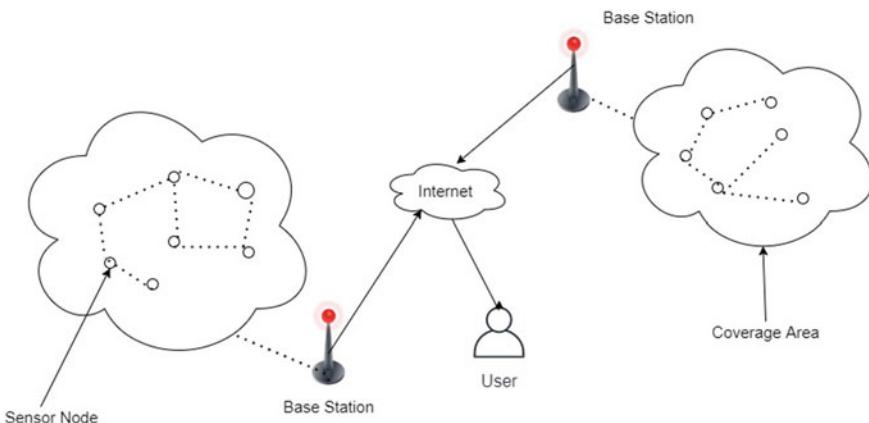


Fig. 1 Wireless sensor network

2 Sinkhole Attack Detection Approaches

Sinkhole attack is the most common network layer attack of WSN in the Transmission Control Protocol/Internet Protocol (TCP/IP) model. In this attack, the attacker introduces a false node that will create a link with its nearby real nodes. Moreover, this compromised node informs neighborhood motes as a false shortest path to the base station to attract the trap the traffic [8]. This adversary node will communicate fake information and prevent the real ones to send the essential data to a base station and deciding if it is the best path based on certain routing matrices [9]. WSNs are susceptible to this attack because transmission happens like many nodes connected to specific base nodes [10]. So, this particular compromised node will disturb the network traffic, hinders the network latency and creates the problem with other network and tries to damage the entire network as much as possible as shown in Fig. 2.

Moreover, in this attack as discussed entire traffic is attracted by a malicious node. So, this node will draw the attention of all the nearby nodes by transmitting fake data by using data transfer capacity. The neighborhood nodes which are been tricked will be sending messages through that particular hostile node which further results in packet loss by that adversary node. Many attacks such as eavesdropping, selective forwarding, blackhole, a wormhole can be empowered by this sinkhole attack.

Due to the increase in demand for various applications in this sensitive domain, certain researchers are still working to provide security mechanisms various approaches have been used to detect sinkhole attacks. These approaches are classified as Anomaly-based, statistical-based, rule-based, cryptographic, and hybrid approaches. The subsequent subsections give a detailed explanation of different approaches with each example.

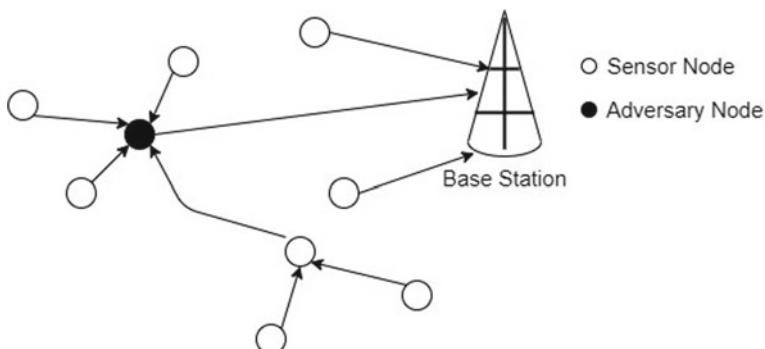


Fig. 2 Sinkhole attack

2.1 Anomaly-Based Approach

In this method, the normal behavior of nodes is defined and studied. Along with, any kind of intrusion detection system searches anything to figure out the anomaly in the network. Moreover, in this technique intrusion is considered anomalous because it is different from normal behavior. Rule-based and statistical-based approaches are a subset of the Anomaly-based approach. Section 3 gives a detailed explanation of this approach with an example.

2.1.1 Rule-Based Approach

In this particular approach, certain rules are defined which are dependent on techniques that create sinkhole attacks. These predefined rules are applied to any kind of intrusion detection system. Further, these rules are embedded into the message transmission from these sensor nodes. So, if any malicious node will be isolated if it did not follow these particular rules [11]. A fuzzy rule method is a common example that has been used under this method. In this technique, IDS is being presented which further makes use of WSNs and mobile robots. These sensor nodes use fuzzy adaptive resonance theory (ART) to detect intruders. After detection, these robots travel to a location at which this attacker is investigated. This ART is then modified to detect the attack rapidly.

2.1.2 Statistical Approach

This approach is based on the data which is associated with the activities of nodes in the sensor network that are deeply studied and recorded. This is done basically to measure the threshold point, for instance, to check transmission of packets between nodes or to monitor the depletion of resources such as CPU usage. Along with this, a false node is detected by comparing its actual situation with the threshold value. If any node is found to exceed the threshold value, then it is detected as a malicious node [12]. Dynamic Trust Management System (DART) comes under this category. Here, each node will measure trust value with all its nearby nodes based on previous knowledge and interaction. After measuring, it will send the data to a base station and now BS decides which particular node is sinkhole after receiving trust values from other nodes. So, if a particular node trust value falls beyond the normal value, then it is taken as a sinkhole attack [13].

2.2 *Cryptographic Approach*

To achieve integrity and confidentiality of data that is communicated from sender to receiver this approach is used. In this method encryption and decryption keys are used to protect information from an attacker. Any kind of packet transmitted in the network is added with another message so that anyone if wants to access the data, requires a key. Along with, modification in messages can easily be detected and these keys help these sensor nodes to check if data is coming from the sink. Two protocols have been used in this category which increases resilience to sinkhole attacks. Both these protocols prevent a malicious node from lying about their fake distances to a base station. RESIST-0 provides high resilience compared to RESIST-1 for sinkhole attacks [14].

2.3 *Hybrid Approach*

This is the combination of basically anomaly-based and cryptographic techniques. This hybrid method ensures that data is protected and helps to detect the adversary node. A hybrid Intrusion detection system is categorized under this approach. This system uses certain detection agent which will identify sinkhole node. Moreover, sensor nodes are attached to this (HIDS) and are further asked to share the resources of that particular node. So, after analyzing the data which is collected from neighbors these adversary nodes are embedded into a blacklist based on anomalous behavior. After that, a central agent will receive the list and a final decision will be made based on the feature of detecting attack patterns [15] (Table 1).

3 Anomaly-Based Approach for Sinkhole Attack Detection

3.1 *Message Digest Method*

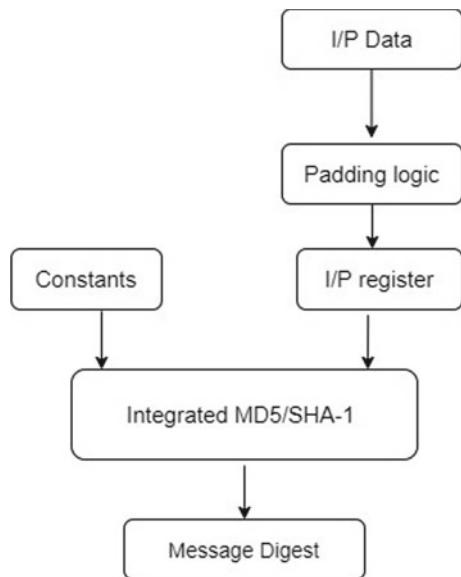
In this paper, a sinkhole attack is detected using the digest approach. As discussed above, it becomes essential to detect the exact malicious node so that message is protected between sender and receiver in the forward routing mechanism. There are certain cryptographic hash functions such as SHA-1, MD5, and RIPEMD which are used for message digest algorithms. So, here we will discuss the extended Double-Davis Mayer scheme which is the combination of the SHA-1 and MD5 message digest algorithm [21]. The main advantage of using this particular combination is that it improves the collision resistance between sensor nodes in the WSN network. The existing MD5 algorithm contains four 32-bit words, which will be further modified so that one more 32-bit word can be included. The result will contain a total of 160 bits. The MD5 hash algorithm consists of 4 rounds each having 16 steps. But in this

Table 1 Summary of certain existing approaches

Type of detection approach	Algorithm	Tool used	Performance parameters
Signature based	Leach [16]	TETCOS NETSIM	Average network lifetime, average network throughput, average energy consumption
Rule-based	RMHSD algorithm [17]	MATLAB	Detection rate, false positive rate
Anomaly-based	Cumulative summation algorithm [18], Markov Chain [19]	CASTALIA, GloMoSim	Detection rate, false alarm rate and communication overhead, MTFA (Mean time false alarm rate), detection rate
Statistical approach	Intrusion detection algorithm [20], dynamic trust management system	MATLAB, NS-2	Detection time, false positive rate, effectiveness, detection rate, packet delivery ratio
Hybrid	Intrusion detection system	INSPIRE TESTBED	Detection rate

proposed approach MD5 is used in collaboration with SHA-1 which will have 20 steps for each particular round. The schematic framework of this model is shown in Fig. 3. In this method, the nodes which exist in the network are allocated with the position of a nearby node that will lie in the path of the sink. So, as soon as there is an incoming of a new node it will advertise itself for the shortest path to sink, therefore it becomes mandatory to find whether that node is trustworthy or sinkhole node. Initially, the message is sent to the real path and the advertised route [22]. At some point in time, both of these paths will meet at a common node. Now, if that advertised route is a sinkhole, then the data would have been modified that has been communicated through it. Thus, the message would be different from the message obtained from trusted nodes. So, by analyzing the information at a common meeting point, predicting the behavior of a node, i.e., trustable or false can be done. This method will now create the digest as soon as the node advertises using the MD5 U SHA-1 algorithm and transmits to the original path and at the same time to the advertised route. This advertised route will keep the message if it will be the trusted node or change the data if it will be a sinkhole.

Fig. 3 Schematic framework



3.2 Zone-Based Intrusion Detection System

In this approach entire network is divided into non-overlapping zones. In the case of inter-zone communication, each gateway zone will be further connected to all other nodes placed at different zones. Along with that, each node will have an IDS agent which will be using the Markov chain. Markov chain is also a kind of anomaly detection algorithm. In this particular algorithm data which is routed to nearby nodes will be monitored and alerts will be generated. These alerts indicate possible attacks. These attacks are generated in the local area then they are transmitted inside a particular zone. The main role of these gateway zones is to provide data aggregation among networks [23].

3.3 Game Theory Approach

This approach is mainly used to analyze anomalies in the network. Host-based IDS and attackers will interact with each other. Further, there are two types of games, i.e., cooperative and non-cooperative which will provide security issues in WSNs. The main objective of the attacker in the game is to forward the infected message from any random node so that it can strike the target node. So, when this infected data reaches the desired machine without meeting the host-based IDS then intrusion is successful.

3.4 Emotional Ants Approach

Ant-colony theory is based on the intrusion detection system. The main principle of this algorithm is to find the route in which there is a minimal activity of sensor nodes and which is also disturbed by certain kinds of intrusion. The agents which are involved in this particular task maintain the record of the nodes which has been visited. Each visit to a certain path has an effect to make it less appealing for other agents so that they are encouraged to visit the paths that are not yet visited [24].

4 Major Design Challenges in Detecting Sinkhole Attacks

- 1 **Communication Pattern:** The messages that are communicated in the network are being forwarded to only the base station that creates a chance for the sinkhole to propel attack. Sinkhole attacks arise when adversaries must communicate false routing information to all nearby nodes to trap traffic congestion. So, this communication pattern will allow a particular intruder to advertise only those nodes which come in the path of the base station instead of transmitting to other nodes in the network and this created the opportunity for an attack [25].
- 2 **Detection Rate and False Alarm:** There are many kinds of DOS attacks in WSN such as Grayhole, Blackhole, Sybil, and Wormhole. So, calculating correct packets transmitted from the number of total packets transmitted is the main challenge because the accurate count is needed as the alarming rate is dependent on it.
- 3 **Unpredictable Behavior of Sinkhole Attacks:** Wireless Sensor Networks make use of some routing metrics for packet transmission which are further used by different routing protocols. So, the node which is affected uses a particular routing metric that will launch a sinkhole attack due to which the entire data flows through the adversary node. Therefore, techniques of sinkhole attacks changed due to metrics of different protocols [13].
- 4 **Limited Resources:** Resource constraints in wireless sensor networks such as limited memory, computational power, and less communication range. So, to design any routing protocols these constraints should be taken into consideration. These constraints destroy the implementation of certain routing protocols [26].
- 5 **Physical Attack:** The sensor motes are deployed in hostile environments due to which most of the nodes are easily accessible. Due to this reason, any attacker from outside can attack the sensor node and get all the required information [27].

5 Conclusion and Future Scope

Wireless Sensor Networks have emerged as the fastest-growing research area nowadays. It consists of thousands of nodes possessing certain resource constraints such

as low cost, limited computational, and communication ability due to which they get vulnerable to many DoS attacks. This paper deals with a discussion of one of the most threatening DoS attacks, i.e., sinkhole attack. This attack threatens the security of almost every layer of a network model. The other DoS attacks such as black-hole, wormhole, selective forwarding get empowered by this attack. Moreover, it prevents the Base station from receiving correct sensed data. Therefore, this article discusses various detection strategies used by certain researchers such as rule-based, anomaly-based, statistical, and cryptographic approaches against this attack. This paper focuses on detailed examples of Anomaly-based approaches which include Message digest, Zone-based, Game theory, and Emotional Ants algorithms. Here, it also discusses major challenges in detecting sinkhole attacks. In the future, a proposal of more efficient detection techniques, and enhancing certain performance parameters such as packet drop, throughput, and Packet Delivery Ratio (PDR) can be improved. Moreover, the techniques for the removal of this attack can be proposed.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. *Comput Netw* 38(4):393–422
2. Kumar V, Jain A, Barwal PN (2014) Wireless sensor networks: security issues, challenges, and solutions. *Int J Inf Comput Technol (IJICT)* 4(8):859–868
3. Li CT (2010) Security of wireless sensor networks: current status and key issues. *Smart Wirel Sens Netw*, 299–313
4. Gong X, Long H, Dong F, Yao Q (2016) Cooperative security communications design with imperfect channel state information in wireless sensor networks. *IET Wirel Sens Syst* 6(2):35–41
5. Sharma R, Grover J (2015) Mitigation of byzantine attack using enhanced cooperative bait detection and prevention scheme (ECBDPS). In: 2015 4th International conference on reliability, infocom technologies and optimization (ICRITO) (Trends and Future Directions). IEEE, pp 1–6
6. Pandey A, Tripathi RC (2010) A survey on wireless sensor networks security. *Int J Comput Appl IJCA* 3:43–49
7. Gothane S, Sarode MV, Raju KS, Study of wireless sensor networks its security issue, challenges and security management
8. Rehman AU, Rehman SU, Raheem H (2019) Sinkhole attacks in wireless sensor networks: a survey. *Wirel Pers Commun* 106(4):2291–2313
9. Chaudhry JA, Tariq U, Amin MA, Rittenhouse RG (2013) Dealing with sinkhole attacks in wireless sensor networks. *Adv Sci Technol Lett* 29(2):7–12
10. Ahlawat J, Chawla M, Sharma K (2012) Attacks and countermeasures in wireless sensor networks. *Int J Comput Sci Commun Eng, Spec Issue Emerg Trends Eng*
11. Tumrongwittayapak C, Varakulsiripunth (2009) Detecting Sinkhole attacks in wireless sensor networks. In: 2009 ICCAS-SICE. IEEE, pp 1966–1971
12. Sharmila S, Umamaheswari G (2011) Detection of sinkhole attack in wireless sensor networks using message digest algorithms. In: 2011 International conference on process automation, control and computing. IEEE, pp. 1–6
13. Roy SD, Singh SA, Choudhury S, Debnath NC (2008) Countering sinkhole and black hole attacks on sensor networks using dynamic trust management. In: 2008 IEEE symposium on computers and communications. IEEE, pp 537–542

14. Papadimitriou A, Le Fessant F, Viana AC, Sengul C (2009) Cryptographic protocols to fight sinkhole attacks on tree-based routing in wireless sensor networks. In: 2009 5th IEEE workshop on secure network protocols. IEEE, pp 43–48
15. Coppolino L, D'Antonio S, Romano L, Spagnuolo G (2010) An intrusion detection system for critical information infrastructures using wireless sensor network technologies. In: 2010 5th international conference on critical infrastructure (CRIS). IEEE, pp 1–8
16. Sundararajan RK, Arumugam U (2015) Intrusion detection algorithm for mitigating sinkhole attack on LEACH protocol in wireless sensor networks. J Sens
17. Zhang Z, Liu S, Bai Y, Zheng Y (2019) M optimal routes hop strategy: detecting sinkhole attacks in wireless sensor networks. Clust Comput 22(3):7677–7685
18. Shang F, Zhou D, Li C, Ye H, Zhao Y (2019) Research on the intrusion detection model based on improved cumulative summation and evidence theory for wireless sensor network. Photon Netw Commun 37(2):212–223
19. Sun B, Wu K, Pooch UW (2006) Zone-based intrusion detection for mobile ad hoc networks. Ad Hoc Sens Wirel Netw 2(3):297–324.c
20. Chen C, Song M, Hsieh G (2010) Intrusion detection of sinkhole attacks in large-scale wireless sensor networks. In: 2010 IEEE International conference on wireless communications, networking and information security. IEEE, pp 711–716
21. Mirvaziri H, Jumari K, Ismail M, Hanapi ZM (2007) A new hash function based on combination of existing digest algorithms. In: 2007 5th Student conference on research and development. IEEE, pp 1–6
22. Mathew, A., & Terence, J. S. (2017, April). A survey on various detection techniques of sinkhole attacks in WSN. In *2017 International Conference on Communication and Signal Processing (ICCPSP)* (pp. 1115–1119). IEEE.
23. Krishnan RS, Julie EG, Robinson YH, Kumar R, Tuan TA, Long HV (2020) Modified zone-based intrusion detection system for security enhancement in mobile ad hoc networks. Wirel Netw 26(2):1275–1289
24. Banerjee S, Grosan C, Abraham A (2005) IDEAS: intrusion detection based on emotional ants for sensors. In: 5th international conference on intelligent systems design and applications (ISDA'05). IEEE, pp 344–349
25. Rassam MA, Zainal A, Maarof MA, Al-Shaboti M (2012) A sinkhole attack detection scheme in mintroute wireless sensor networks. In: 2012 international symposium on telecommunication technologies. IEEE, pp 71–75
26. Ali M, Nadeem M, Siddique A, Ahmad S, Ijaz A, Addressing Sinkhole attacks in wireless sensor networks—a review
27. Zaminkar M, Fotohi R (2020) SoS-RPL: securing internet of things against sinkhole attack using RPL protocol-based node rating and ranking mechanism. Wirel Perso Commun 114:1287–1312

HPGAB3C: A Novel Hybridized Optimization Approach



Rattan Deep Aneja, Amit Kumar Bindal, and Shakti Kumar

Abstract This paper proposes a novel soft computing based hybrid approach named HPGAB3C for global optimization. The proposed approach is based upon the nature inspired hybridized approaches used for optimization in various domains. In the proposed approach, a combination of genetic algorithms (GA) and big bang big crunch (BBBC) approach is used where parallel/multiple populations will be passed to genetic algorithm and the best population evaluated by genetic algorithm will be passed on to big bang big crunch approach to compute the optimized solution. We implemented the approach in MATLAB. Its performance for optimal route evaluation was tested on WMNs with node sizes varying from 1000 to 10,000 nodes. Performance of the proposed approach is compared with 9 other approaches. We observed that the proposed approach outperforms all the other approaches.

Keywords WMN · Routing · HPGAB3C · GA · BBBC · Soft computing · Hybrid approach

1 Introduction

In the recent past, soft computing based approaches have found extensive use in finding optimized solutions for the class of problems belonging to NP-hard or NP-complete problems [6, 10]. Soft computing is used where the best possible solution can be replaced with a good enough solution. Soft computing based optimization approaches perform fairly well individually however it has been observed that hybridized approaches have better performance than the individual approaches [4, 7, 13, 19, 21, 30, 32]. Multiple or parallel populations provides a vast range of inputs and improves the probability of finding the optimized solution quickly [9, 16, 29, 31].

R. D. Aneja (✉) · A. K. Bindal
Maharishi Markandeshwar (Deemed to be University), Mullana, India

S. Kumar
Panipat Institute of Engineering & Technology, Samalkha, India

Genetic algorithms (GAs) are very effective search and optimization algorithms employed for global optimization. Researchers are preferring to use GAs with little modifications in multiple domains and finding a better results as compared to other algorithms [1, 11, 14]. Big bang big crunch (BBCB) based optimization got attention in recent years and was found to provide optimized solutions for numerous non-deterministic polynomial problems [32, 33]. This paper proposes a novel hybridized optimization approach based upon GA and BBCB. Multiple/parallel populations are considered as input to the proposed approach for finding the optimized solution. The proposed approach was validated using optimal path evaluation in WMNs problem.

Wireless mesh networks (WMNs) have attracted the attention of research community because of their dynamic self-balancing nature. WMNs create a radio network by interconnection of different client nodes. WMNs can be classified [12] as (i) Mesh architecture based client WMNs (ii) Infrastructure mesh architecture based WMNs and (iii) Hybrid mesh WMNs. Routing is an important aspect to measure the efficiency of WMNs. A wide range of reactive routing protocols is available for communication networks such as DSR, AODV, LQSR, ANODR, ABR, FW-AODV, AODV-DF [15, 17, 18, 20, 24, 28], etc. but FW-AODV and AODV are the only reactive protocols that are frequently used as they provide better performance [28]. Proactive routing protocols were observed even more efficient as compared to reactive protocols [2, 27].

Round trip time per hop [25], minimum hop count [5], Packet pair delay per hop, Integrated link cost [16], Expected transmission count [3], etc. are some of the performance measures to evaluate performance of routing algorithms/approaches. Integrated link cost (ILC) [16] is a fuzzy based measure used for performance evaluation in this paper. ILC is a function of 3 major parameters, i.e. delay, jitter and throughput.

This paper is divided into 5 sections. Introduction is covered in Sect. 1. Section 2 presents the related work done in literature. Proposed approach/algorithm is written in Sect. 3 of the paper. Implementation of proposed approach and performance comparison is discussed in Sect. 4. Conclusion is drawn in Sect. 5.

2 Related Work

Soft computing approach is the significant thrust area for researchers trying to optimize the routing algorithms in Wireless Mesh Network. Search and optimization techniques are being used either individually or in combination making a new hybrid approach. Some of the observations in concern to hybrid approaches and routing in WMNs is being discussed over here.

Teaching learning based optimization is integrated with Adaptive neural fuzzy inference system and Taguchi method and proposed a hybrid approach [4] to reduce the acoustic streaming energy and was found to perform better than other existing approaches.

Another hybrid approach was proposed by combining Sine Cosine Algorithm and Grey Wolf Optimizer [30]. The proposed approach was tested on one sine dataset, five biomedical datasets and twenty-two benchmark tests. Solutions provided by proposed hybrid GWOSCA were compared with Particle Swarm Optimization, Sine Cosine Algorithm, Ant Lion Optimizer, Grey Wolf Optimizer, Whale Optimization Algorithm and proved that the proposed approach was highly effective as compared to others.

AODV and two soft computing based routing approach ACO and BBBC were applied to Client WMNs of 10, 20, 30, 50 and 100 nodes and cost of paths evaluated using ILC. It was found by the authors that BBBC performed well [22]. BBBC routing approach was able to find optimal paths quickly. ACO and AODV could not perform better in these scenarios show the superiority of BBBC over the other two approaches. Limitation of this paper was that authors considered network of maximum 100 node client WMN.

FW-AODV routing approach is an extension of AODV proposed in [28] and compared the performance with BAT, AODV and ACO based routing approaches. Authors considered the scenario of 100, 500 and 1000 node client WMNs and cost evaluated using ILC. Authors observed that FW-AODV performed better than other three approaches.

Two soft computing based approaches BBO and BBBC were applied for finding shortest path in client WMNs with varying numbers of nodes such as 25, 64, 100 and 2500 nodes [23]. Observations and performance comparisons showed that BBBC produces minimal cost paths as compare to BBO. Authors also observed that BBBC is capable of finding shortest routes in less time as compare to BBO.

A multi-population/parallel soft computing approach named PBBBC was proposed in [16] and compared the performance for finding optimal paths in WMNs. Trials were made with 100, 500, 1000, 1500, 2000 and 2500 nodes client WMNs and results compared with other 7 routing approaches namely BBBC, FA (Firefly algorithm), DSR, ACO, BAT, AODV, BBO. It was observed that PBBBC outperforms all other 7 approaches and found optimal paths maximum times.

It has been observed that hybrid approaches are performing better than individual ones and soft computing based approaches are giving better results in minimal time. One more observation was that most of the authors considered client WMNs of maximum 2500 nodes. Taking these observations by different authors into consideration in this paper, we propose a novel soft computing based hybrid approach namely HPGAB3C and applied it for finding optimal routes in node client WMNs. Large sized (more than 1000 nodes with a maximum of 10,000 nodes) client WMNs are considered in this paper. Cost of route is calculated using ILC and compared with other 9 existing approaches.

3 Proposed Approach

Soft computing based approaches include a variety of search and optimization algorithms that include nature inspired, evolutionary computing, particle swarm optimization, non-swarm optimization, physics based, chemistry based algorithms [8, 26, 31, 33]. Taking the advantages of such soft computing approaches and improving the efficiency of the proposed approach, in this paper we combined Multi-population/parallel genetic algorithms (GA) with big bang big crunch (BBBC). The proposed hybridized approach is applied for finding the optimal paths in WMNs. We considered ‘3 populations of N individuals each’ as input and applied the GA for global optimization to find best ‘N individuals’ as output. Then applied the BBBC approach over ‘N best individuals’ for local optimization. The proposed approach has two phases where first phase will process multiple populations (3 populations of N individuals each) and the second phase will process single population of N individuals. The proposed HPGAB3C approach is given below:

3.1 Nomenclature

CN: Client nodes

N: No. of genes/paths

out_loop: Maximum number of iterations for outer loop

ga_in_loop: Maximum number of iterations for GA phase

bbbc_in_loop: Maximum number of iterations for BBBC phase

pop_mat: Matrix containing 3 populations each of N individuals

path_mat: Matrix of population with N individuals

elite: Best gene/shortest path.

3.2 HPGAB3C Approach

1. Randomly create an initial population of size ‘3 N’ and put it into pop_mat.
2. Evaluate the fitness of each gene in pop_mat and select the best gene as elite.
3. Repeat step 4 and step 5 for out_loop times
4. Repeat GA (Phase 1) steps for ga_in_loop times
 - (a) Select fitter individuals as parents
 - (b) Apply the Crossover
 - (c) Mutate the population
 - (d) Evaluate the new population and select the best gene
 - (e) If best gene is better than elite then replace the elite and update pop_mat
 - (f) Evaluate pop_mat and select ‘N’ best population.
 - (g) Put the ‘N’ best population into path_mat and pass to the BBBC phase

5. Repeat BBBC (Phase 2) steps for bbbc_in_loop times
 - (a) Evaluate the fitness of path_mat and select the best gene as elite
 - (b) Apply big bang phase (Create new population around the elite)
 - (c) Apply big crunch phase (Find the best individual)
 - (d) Select the best individual as elite
 - (e) Merge pop_mat with path_mat and select best ‘3 N’ populations of N individuals each to update pop_mat
 - (f) Pass the pop_mat to GA phase
6. Return elite as the best gene

3.3 Pseudocode for HPGAB3C Based Routing Approach in WMNs

1. Randomly create ‘3 N’ initial population (paths from source node to destination node) and put it into pop_mat.
2. Evaluate the fitness of each path in pop_mat and select the best path as elite.
3. Repeat step 4 and step 5 for out_loop times.
4. Repeat GA (Phase 1) steps for ga_in_loop times.
 - (a) Select the paths from population.
 - (b) Apply the Crossover.
 - (c) Mutate the paths produced after Crossover.
 - (d) Evaluate new paths and select the best path.
 - (e) If best path is better than elite then replace the elite and update pop_mat.
 - (f) Evaluate pop_mat and select ‘N’ best paths.
 - (g) Put the ‘N’ best paths into path_mat and pass to the BBBC phase.
5. Repeat BBBC (Phase 2) steps for bbbc_in_loop times.
 - (a) Evaluate the fitness of path_mat and select the best path as elite.
 - (b) Apply big bang phase (Create new population around the elite)
 - (c) Apply big crunch phase (Find the best path among generated paths).
 - (d) Select the best path as elite.
 - (e) Merge pop_mat with path_mat and select best ‘3 N’ paths to update pop_mat.
 - (f) Pass pop_mat to GA phase.
6. Return elite as the best path.

We have implemented the proposed approach for finding optimal path in WMNs and compared the results with 9 other approaches. Results and performance comparison of HPGAB3C approach are shown in next section.

4 Simulation, Results and Discussion

To evaluate performance of the proposed HPGAB3C based routing approach and validate HPGAB3C approach we have considered 9 other routing approaches to compare the results. We have implemented the routing approaches in MATLAB by deploying WMN client nodes in a defined area and then finding routes from source node to destination node. The cost of routes is evaluated using ILC (Integrated Link Cost) while conducting trials on a computer system with specifications as Core i5 processor@1.6 GHz speed, 8 GB RAM for 1000 node client WMN to 10,000 node client WMN. Architectural details such as number of nodes in WMN, specified area, radio range of each node and timing constraints are given in Table 1.

Simulations for the routing between source node and destination node was done and the route cost calculated using ILC performance measure. 30 trials were taken for each of the architectural scenarios specified in Table 1. Simulation results for timing constraint 0.1 s, area of 1000 m², 250 m radio range and 1000 node client WMN is presented in Table 2.

Similar simulation results are observed for every architectural scenario. For each architectural scenario, 30 trials were conducted that comes out to a total of 900 trials. We presented here only one table of simulation results for the sake of brevity. Brief summary of all the simulation result tables has been prepared out of which the comparative analysis of 1000, 5000 and 9000 node client WMNs is presented here to keep the paper brief.

4.1 Performance Analysis of 1000 Node Client WMNs

Performance of 1000 node Client WMN is observed by deploying 1000 nodes in the area of 1000 m² and timing constraints of 0.1, 0.2 and 0.3 s are considered.

Table 1 Architecture scenario of client WMNs

No. of nodes	Area (In m ²)	Radio range	Timing constraints (In s)
1000	1000 × 1000	250	0.1, 0.2, 0.3
2000	1500 × 1500	250	0.4, 0.6, 0.7
3000	1500 × 1500	250	1.0, 1.2, 1.3
4000	1500 × 1500	250	2.1, 2.2, 2.3
5000	2000 × 2000	250	2.0, 2.5, 3.0
6000	2000 × 2000	250	4.0, 4.5, 5.0
7000	2500 × 2500	250	4.2, 4.5, 4.8
8000	3000 × 3000	250	5.5, 6.0, 6.5
9000	3000 × 3000	250	4.5, 5.0, 5.5
10,000	3500 × 3500	250	6.0, 7.0, 8.0

Table 2 Simulation results for 1000 node Client WMN with 0.1 s timing constraint

Routing approach	AODV	ACO	BAT	BBO	DSR	Firefly	BBBC	GA	PB3C	HPGAB3C	Min cost
Trial	Route cost										
1	Not found	Not found	2.6144	2.6144	Not found	2.6144	0.82445	0.79932	0.91521	0.98003	0.79932
2	Not found	Not found	2.9402	2.9402	Not found	2.9402	2.5622	1.5544	1.424	1.424	1.424
3	Not found	Not found	3.7798	3.7798	Not found	3.7798	2.3057	2.2131	1.5826	1.5826	1.5826
4	Not found	Not found	9.581	9.581	Not found	9.581	3.3452	1.4279	1.8158	1.8158	1.4279
5	Not found	Not found	3.8493	3.8493	Not found	3.8493	0.85994	0.74469	2.1354	1.3132	0.74469
6	Not found	Not found	3.4463	3.4463	Not found	3.4463	2.023	2.0736	2.5627	1.0644	1.0644
7	Not found	Not found	2.6335	2.6335	Not found	2.6335	2.6335	0.98039	2.2035	2.2035	0.98039
8	Not found	Not found	3.941	3.941	Not found	3.941	2.2031	2.329	2.2093	2.2093	2.2031
9	Not found	Not found	5.4154	5.4154	Not found	1.6545	2.4163	1.8269	1.5396	2.2324	1.5396
10	Not found	Not found	2.129	2.129	Not found	2.129	1.4292	1.347	2.154	2.154	1.347
11	Not found	Not found	6.7512	6.7512	Not found	6.7512	4.3561	2.2899	3.1158	2.0761	2.0761
12	Not found	Not found	3.82	3.82	Not found	3.82	2.055	1.958	1.5141	2.788	1.5141
13	Not found	Not found	3.843	3.843	Not found	3.843	3.1381	2.2171	1.8726	4.6543	1.8726
14	Not found	Not found	3.2866	3.2866	Not found	3.2866	3.2866	1.098	1.1856	4.3636	1.098
15	Not found	Not found	2.2696	2.2696	Not found	2.2696	2.0388	1.5876	0.75113	1.093	0.75113
16	Not found	Not found	1.6741	1.6741	Not found	1.6741	0.90435	1.2156	1.5774	1.3922	0.90435
17	Not found	Not found	2.6683	2.6683	Not found	2.6683	1.2066	1.9844	0.99589	2.0126	0.99589
18	Not found	Not found	1.5087	1.5087	Not found	1.0418	1.5087	1.4178	0.62466	0.77459	0.62466
19	Not found	Not found	2.8021	2.8021	Not found	2.8021	1.0225	0.7771	0.80268	1.2378	0.7771
20	Not found	Not found	1.2585	1.2585	Not found	1.2585	0.6911	0.68034	1.1947	0.68443	0.68034

(continued)

Table 2 (continued)

Routing approach	AODV	ACO	BAT	BBO	DSR	Firefly	BBBC	GA	PB3C	HGPAB3C	Min cost
21	Not found	Not found	0.52502	0.52502	Not found	0.525	0.4714	0.44306	0.71321	0.75949	0.44306
22	Not found	Not found	0.85452	0.85452	Not found	0.4331	0.69979	0.8263	0.68676	0.4701	0.43308
23	Not found	Not found	0.77852	0.77852	Not found	0.3954	0.50324	0.5052	0.63338	0.5057	0.39541
24	Not found	Not found	0.58319	0.58319	Not found	0.4427	0.33887	0.34324	0.78674	0.35278	0.33887
25	Not found	Not found	0.69417	0.69417	Not found	0.6942	0.48167	0.46293	0.50733	0.40467	0.40467
26	Not found	Not found	2.071	2.071	Not found	1.8346	0.49749	1.1454	0.8383	0.44497	0.44497
27	Not found	Not found	0.89472	0.89472	Not found	0.8947	0.28021	0.40985	0.40985	0.44811	0.28021
28	Not found	Not found	0.61706	0.61706	Not found	0.6171	0.34503	0.52105	0.52597	0.40848	0.34503
29	Not found	Not found	0.59423	0.59423	Not found	0.5942	0.41469	0.52644	0.63821	0.3878	0.3878
30	Not found	Not found	0.8903	0.8903	Not found	0.4366	0.31594	0.41824	0.65423	0.39467	0.31594
No. of times optimal path found	0	0	0	0	2	6	9	8	7		

Performance comparison is shown in Table 3 and comparative analysis is presented in Fig. 1. It has been observed that ACO and DSR were failed to find any path for 1000 node Client WMN. For 0.1 s timing constraint, AODV was failed to find any path. Further, BAT and BBO could not find any optimal path. GA found optimal path maximum 9 times followed by PB3C (6 times matchless and 2 times shared), HPGAB3C (5 times matchless and 2 times shared), BBBC found 6 times and FA found 2 times. For 0.2 s timing constraint HPGAB3C found optimal path maximum times (9 times individually and 4 times shared) along with BBBC (7 times individually and 6 times shared) followed by GA (4 times individually and 5 times shared), AODV (1 time individually and 8 times shared), FA (1 time individually and 3 times shared), PB3C, BAT and BBO (3 times each shared). For 0.3 s timing constraint, AODV was failed to find any path. Further, BBBC found optimal path maximum times (14 times individually and 2 times shared) followed by GA (6 times individually and 2 times shared), HPGAB3C found 7 times, FA, BAT and BBO (2 times each shared) and PB3C found 1 time.

4.2 Performance analysis of 5000 node Client WMNs

Performance of 5000 node Client WMN is observed by deploying 5000 nodes in the area of 2000 m^2 and timing constraints of 2, 2.5 and 3 s are considered. Performance comparison is shown in Table 4 and comparative analysis is presented in Fig. 2. It has been observed that AODV, ACO and DSR were failed to find any path for 5000 node Client WMN. Further, BBO and BAT could not find any optimal path. For 2 s timing constraint HPGAB3C found optimal path maximum 10 times followed by BBBC found 9 times, GA found 6 times, FA found 3 times and PB3C found 2 times. For 2.5 s timing constraint HPGAB3C found optimal path maximum times (10 times individually and 1 time shared) followed by GA found 9 times, BBBC found 5 times, PB3C found (3 times matchless and 1 time shared) and FA found 2 times. For 3 s timing constraint HPGAB3C found optimal path maximum times (9 times individually and 1 time shared) followed by BBBC found 9 times, GA found 7 times, PB3C found (3 times matchless and 1 time shared) and FA found 1 time.

4.3 Performance analysis of 9000 node Client WMNs

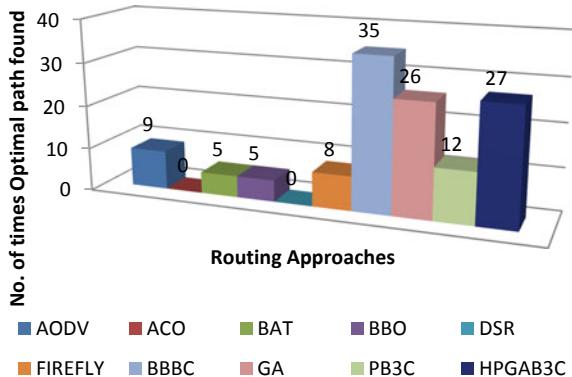
Performance of 9000 node Client WMN is observed by deploying 9000 nodes in the area of 3000 m^2 and timing constraints of 4.5, 5 and 5.5 s are considered. Performance comparison is shown in the Table 5 and comparative analysis is presented in Fig. 3. It has been observed that AODV, ACO and DSR were failed to find any path for 7000 node Client WMN. Further, BBO and BAT could not find any optimal path for 4.5 and 5 s timing constraints. For 4.5 s timing constraint, GA found optimal path maximum 10 times followed by HPGAB3C found 8 times, BBBC found 5 times,

Table 3 Performance comparison of 1000 node Client WMN

S. No.	Nodes	Area (Square Metres)	Timing constraint	Trials	AODV	ACO	BAT	BBO	DSR	Firefly	BBBC	GA	PB3C	HPGAB3C
1	1000	1000 × 1000	0.1	30	—	0	0	—	2	6	9	6 + A	5 + A	
2	1000	1000 × 1000	0.2	30	1 + F	—	B	B	—	1 + B	7 + E	4 + D	B	9 + C
3	1000	1000 × 1000	0.3	30	—	—	A	A	—	A	14 + A	6 + A	1	7
Total				90	9	—	5	5	—	8	35	26	12	27

(A = 2 times, B = 3 times, C = 4 times, D = 5 times, E = 6 times, F = 8 times) jointly winner, 0 Means no optimal path found, Means failed to find any path

Fig. 1 Comparative analysis of 1000 node Client WMN



PB3C found 4 times and FA found 3 times. For 5 s timing constraint HPGAB3C found optimal path maximum 8 times followed by BBBC, GA and PB3C found 7 times each and FA found 1 time. For 5.5 s timing constraint BBBC found optimal path maximum times (9 times matchless and 1 time shared) followed by HPGAB3C found 9 times, GA found (5 times matchless and 1 time shared), PB3C found 4 times, FA found (2 times individually and 1 time shared).

4.4 Overall performance analysis of all Client WMNs

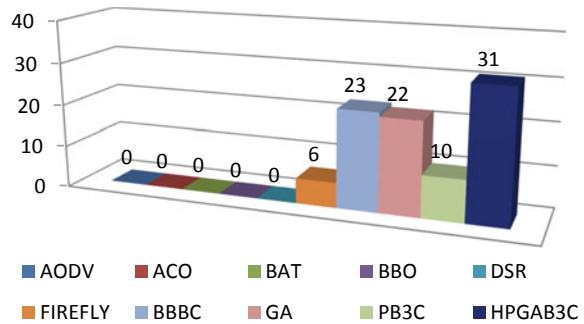
Overall performance comparison for all Client WMNs is shown in Table 6. Client WMNs of 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000 and 10,000 nodes are considered with the architecture scenario presented in Table 1. It has been observed that HPGAB3C found optimal path maximum 296 times followed by BBBC found 242 times, GA found 240 times, PB3C found 118 times, Firefly (FA) found 44 times, BAT and BBO found 13 times each and AODV found 12 times. This is to mention that ACO and DSR were failed to find any path in all the architecture scenarios. Overall comparative analysis is presented in Fig. 4.

Table 4 Performance comparison of 5000 node Client WMN

S. No.	Nodes	Area (m ²)	Timing constraint	Trials	AODV	ACO	BAT	BBO	DSR	Firefly	BBBC	GA	PB3C	HPGAB3C
1	5000	2000 × 2000	2	30	—	0	0	—	—	3	9	6	2	10
2	5000	2000 × 2000	2.5	30	—	0	0	—	—	2	5	9	3 + G	10 + G
3	5000	2000 × 2000	3	30	—	0	0	—	—	1	9	7	3 + G	9 + G
Total				90	—	0	0	—	—	6	23	22	10	31

G = 1 time jointly winner, 0 Means no optimal path found, Means failed to find any path

Fig. 2 Comparative analysis of 5000 node Client WMN



5 Conclusion

This paper proposes a novel global optimization approach HPGAB3C. HPGAB3C is a hybridized soft computing based optimization approach designed with the combination of parallel genetic algorithm and big bang big crunch approaches. For validating the proposed approach, HPGAB3C is applied to find optimal path in Client WMNs as per the scenario presented in Table 1 for 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000 and 10,000 nodes and compared the performance with 9 other routing approaches namely AODV, ACO, DSR, BAT, BBO, FA, GA, BBBC, PB3C. ILC performance measure is used to evaluate the route cost for all the routing approaches. It has been observed from Table 6 that HPGAB3C performed best as compare to other 9 routing approaches by finding optimal path maximum 296 times followed by BBBC found optimal path 242 times followed by GA found optimal path 240 times followed by PB3C found optimal path 118 times. Firefly (FA) routing approach was able to find the optimal path only 44 times. BAT and BBO found optimal path 13 times each and AODV found optimal path 12 times. ACO and DSR were failed to find any path in all the architecture scenarios. From Table 6 and Fig. 4, we can conclude that proposed approach HPGAB3C is superior to all other 9 routing approaches taken into consideration for finding the optimal path.

Table 5 Performance comparison of 9000 node Client WMN

S. No.	Nodes	Area (m ²)	Timing constraint	Trials	AODV	ACO	BAT	BBO	DSR	Firefly	BBBC	GA	PB3C	HPGAB3C
1	9000	3000 × 3000	4.5	30	—	0	0	—	—	3	5	10	4	8
2	9000	3000 × 3000	5	30	—	0	0	—	1	7	7	7	7	8
3	9000	3000 × 3000	5.5	30	—	G	G	—	2 + G	9 + G	5 + G	4	9	
Total				90	—	1	1	—	7	22	23	15	25	

G = 1 time jointly winner, 0 Means no optimal path found, Means failed to find any path

Fig. 3 Comparative analysis of 9000 node Client WMN

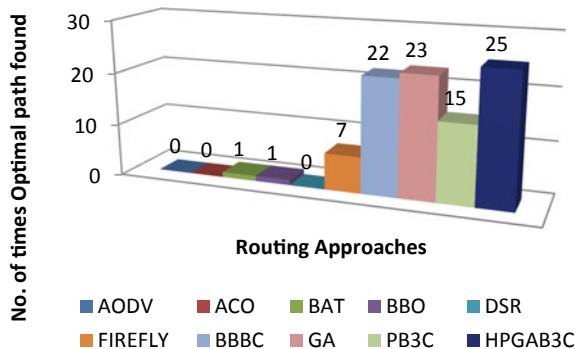
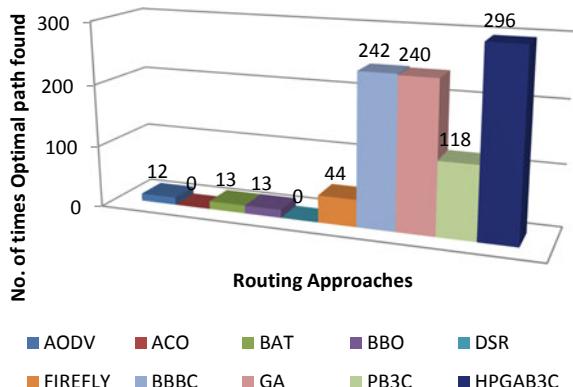


Table 6 Overall performance comparison of all Client WMNs

S. No.	Nodes	Trials	AODV	ACO	BAT	BBO	DSR	Firefly	BBBC	GA	PB3C	HPGAB3C
1	1000	90	9	0	5	5	0	8	35	26	12	27
2	2000	90	0	0	1	1	0	2	27	21	10	34
3	3000	90	0	0	0	0	0	1	25	23	10	33
4	4000	90	2	0	0	0	0	3	27	18	6	34
5	5000	90	0	0	0	0	0	6	23	22	10	31
6	6000	90	0	0	0	0	0	3	25	20	13	29
7	7000	90	1	0	0	0	0	2	20	24	9	34
8	8000	90	0	0	0	0	0	1	18	30	11	30
9	9000	90	0	0	1	1	0	7	22	23	15	25
10	10,000	90	0	0	6	6	0	11	20	33	22	19
Total no. of times optimal path found		900	12	0	13	13	0	44	242	240	118	296

Fig. 4 Overall comparative analysis of all Client WMNs



References

- Ali MZ, Awad NH, Suganthan PN, Shatnawi AM, Reynolds RG (2018) An improved class of real-coded Genetic Algorithms for numerical optimization. Neurocomputing 275, pp 155–166. Available from <https://doi.org/10.1016/j.neucom.2017.05.054>
- Aneja RD, Bindal AK, Kumar S (2019) WMN routing: state of the art survey. Int J Manage Technol Eng IX(Xi):111–115
- Boushaba M, Hafid A, Gendreau M (2016) Source-based routing in wireless mesh networks. IEEE Syst J 10(1):262–270
- Le Chau N, Dao TP, Dang VA (2020) An efficient hybrid approach of improved adaptive neural fuzzy inference system and teaching learning-based optimization for design optimization of a jet pump-based thermoacoustic-stirling heat engine. Neural Comput Appl 32(11):7259–7273. Available from <https://doi.org/10.1007/s00521-019-04249-y>
- Chaugule M, Desai A (2016) Reliable metrics for wireless mesh network. Int Res J Eng Technol (IRJET) 03(01):932–938. Available from <https://www.irjet.net/archives/V3/i1/IRJET-V3I1163.pdf>
- Daoqing Z, Mingyan J (2020) Parallel discrete lion swarm optimization algorithm for solving traveling salesman problem. J Syst Eng Electron 31(4):751–760
- El-Kenawy ES, Eid M (2020) Hybrid gray wolf and particle swarm optimization for feature selection. Int J Innovative Comput Inf Control 16(3):831–844
- Fister I, Yang XS, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. Elektrotehniski Vestn/Electrotech Rev 80(3):116–122
- Gulcu S, Mahi M, Baykan OK, Kodaz H (2018) A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. Soft Comput 22(5):1669–1685
- Hajipour V, Khodakarami V, Tavana M (2014) The redundancy queuing-location-allocation problem: a novel approach. IEEE Trans Eng Manage 61(3):534–544
- Hassanat A, Almohammadi K, Alkafaween E, Abunawas E, Hammouri A, Prasath VBS (2019) Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. Information (Switzerland) 10(12):1–36
- Rejina Parvin J (2019) An overview of wireless mesh networks. Wireless mesh networks-security, architectures and protocols, IntechOpen, vol 1, pp 1–13. Available from <https://doi.org/10.1039/C7RA00172J%0A; www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics%0A; https://doi.org/10.1016/j.colsurfa.2011.12.014>
- Kiziloluk S, Özer AB (2019) Hybrid parliamentary optimization and big bang-big crunch algorithm for global optimization. Turk J Electr Eng Comput Sci 27(3):1954–1969
- Kora P, Yadlapalli P (2017) Crossover operators in genetic algorithms: a review. Int J Comput Appl 162(10):34–36
- Kum DW, Le AN, Cho YZ, Toh CK, Lee IS (2010) An efficient on-demand routing approach with: directional flooding for wireless mesh networks. J Commun Netw 12(1):67–73
- Kumar S, Singh A, Walia S (2018) Parallel Big Bang–Big Crunch global optimization algorithm: performance and its applications to routing in WMNs. Wirel Pers Commun 100(4): 1601–1618. Available from <https://doi.org/10.1007/s11277-018-5656-y>
- Li Z, Hu J, Gui N, Xu L, Zhao W, Jiang L, Jin J (2013) Multi-path anonymous on demand routing protocol. In: Proceedings—3rd international conference on instrumentation and measurement, computer, communication and control, IMCCC 2013, pp 858–863
- Saeed N, Amin RU, Malik AS, Kasi MK, Kasi B (2017) Performance evaluation of AODV, DSDV and DSR routing protocols in unplanned areas. Tech J Univ Eng Technol (UET) Taxila 22(1):143–150
- Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl Soft Comput J 62:1019–1043. Available from <https://doi.org/10.1016/j.asoc.2017.09.039>
- No I, Thamizhmaran K (2017) Modified ABR (M-ABR) routing protocol with multi-cost parameters for effective communication in MANETs. Int J Adv Res Comput Sci 8(1):288–291

21. Ben Seghier MEA, Carvalho H, Keshtegar B, Correia JAFO, Berto F (2020) Novel hybridized adaptive neuro-fuzzy inference system models based particle swarm optimization and genetic algorithms for accurate prediction of stress intensity factor. *Fatigue Fract Eng Mater Struct* 43(11):2653–2667
22. Sharma S, Kumar S, Singh B (2014) Hybrid intelligent routing in wireless mesh networks: soft computing based approaches. *Int J Intell Syst Appl* 6(1):45–57
23. Sharma S, Kumar S, Singh B (2013) Routing in wireless mesh networks: two soft computing based approaches. *Int J Mob Netw Commun Telematics* 3(3):29–39
24. Sharma S, Malik A (2019) Routing in wireless mesh networks based on termites' intelligence. *Int J Appl Metaheuristic Comput* 8(2):1–21
25. Shoukat IA, Al-Dhelaan A, Iftikhar M (2013) Realization of correlation between Round Trip Time (RTT) and hop counts in packet switched networks. *Life Sci J* 10(4):569–576
26. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12(6):702–713
27. Singh A, Kumar S, Walia SS (2017) Routing protocols for WMNS: a survey. *Int J Adv Res Comput Sci Softw Eng* 7(7):1
28. Singh A, Walia SS, Kumar S (2017) FW-AODV: an optimized AODV routing protocol for wireless mesh networks. *Int J Adv Res Comput Sci* 8(3):1131–1135
29. Singh A, Walia SS, Kumar S (2017) P3PGA: Multi population 3 parent genetic algorithm and its application to routing in WMNs. *Int J Adv Res Comput Sci* 8(5):968–975
30. Singh N, Singh SB (2017) A novel hybrid GWO-SCA approach for optimization problems. *Eng Sci Technol Int J* 20(6):1586–1601. Available from <https://doi.org/10.1016/j.jestch.2017.11.001>
31. Song PC, Pan JS, Chu SC (2020) A parallel compact cuckoo search algorithm for three-dimensional path planning. *Appl Soft Comput J* 94(106443):1–16. Available from <https://doi.org/10.1016/j.asoc.2020.106443>
32. Wang J, Kumbasar T (2019) Parameter optimization of interval Type-2 Fuzzy neural networks based on PSO and BBBC methods. *IEEE/CAA J Autom Sin* 6(1):247–257
33. Wang JJ, Kumbasar T (2020) Optimal PID control of spatial inverted pendulum with big bang-big crunch optimization. *IEEE/CAA J Automat Sin* 7(3):822–832

COVID-19 Identification on Chest X-rays with Deep Learning Technique



Preeti Sharma and Devershi Pallavi Bhatt

Abstract The COVID-19 infection has firmly affected all nations globally. COVID-19 disease is a lung infection by the novel CORONA virus. The present study aims to develop a binary classification deep neural network that identifies the COVID-19 disease on chest X-ray scans. The proposed model divides the chest X-rays into two classes; one is a normal chest X-ray or the other is covid infected. The model has utilized the benefit of the transfer learning method and implemented the ResNet-50 pre-trained model as the backbone model. 1200 chest X-rays have been used to conduct this study while the achieved accuracy is 97.92%. The proposed model also manifests the effect of deep learning techniques in the medical imaging domain.

Keywords Medical imaging · Deep learning · Lung disease · COVID-19

1 Introduction

Lung diseases have been affecting human life immensely. In the year 2020, the outbreak of COVID-19 disease declared as pandemic; has raised the worldwide death rate due to lung diseases. COVID-19 is a kind of lung infection due to a new strain of virus called CORONA VIRUS. Corona virus-infected patients can infect other healthy patients. The major problem with this virus is its asymptomatic nature. There have been many cases where a covid infected patient does not show any kind of symptoms but can infect others though. The spreading of this virus must be stopped, which could be possible with the timely diagnosis of positive patients and isolate them from others. Some typical symptoms of having this lung infection are fever, chronic chest pain, lasting cough, mucus production, briefness of breath, etc. A person having any of these symptoms should consult with the doctor.

RT-PCR test is being practiced as a standard lab test concerning the detection of COVID-19 infection [1]. The problem with the RT-PCR test is an infected patient with symptoms that may get a negative report, thus making the situation even worst in

P. Sharma · D. P. Bhatt (✉)

Manipal University Jaipur, Dehmi Kalan, Jaipur 303007, India

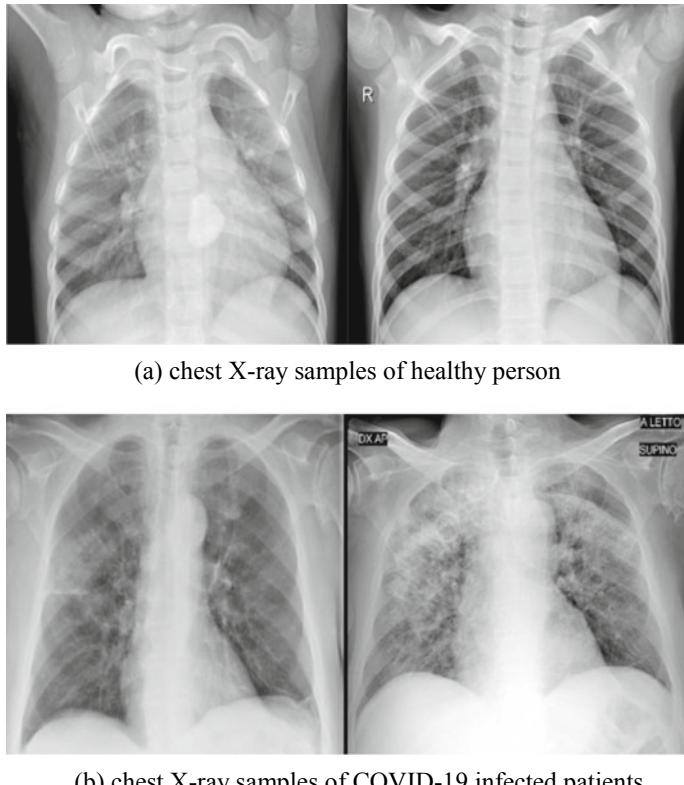
spreading this virus. Doctors also suggest lung scans from medical imaging methods like X-ray scans and CT scans to get detailed information of lungs. Medical imaging methods help doctors to perceive inside details of inner body parts. X-ray is the most prevalent and most practiced imaging method. The low scanning cost of X-ray instruments and availability are reasons for presenting X-ray as the most suitable imaging technique. Chest X-rays provide images of the many organs, including images of the heart, images of lungs, airways, chest bones, spine bones, blood vessels. X-ray scans can indicate the presence of fluid or air nearby the lung. Chest X-ray assists doctors to get a detailed idea of the predicament that can be related to heart, lung collapse, pneumonia, tumor, cracked ribs, etc. Deep learning has become the most preferred method to develop computer models that can intimate the working of the human brain. Deep neural models can help in the diagnosis of any kind of abnormality inside the lungs with the help of medical imaging techniques. The high-performance rate of deep models is attracting researchers to develop stronger deep models. A deep learning model amidst a high accuracy level and low-performance speed can support doctors in the timely detection of diseases.

The present study proposes a deep learning-based classification model that detects the COVID-19 infection. The dataset used in the present study are X-ray images of the chest including X-rays of healthy patients, and X-rays showing covid infection (Fig. 1). The proposed model is a binary classification model to classify the chest X-rays (CXRs) in categories of normal X-ray or COVID infected. The proposed model is based on the backbone of ResNet-50 [2] pre-trained deep network. ResNet-50 is a 50 layer deep convolutional neural network (CNN) that is trained on the ImageNet dataset. ResNet-50 [2] is among the top pre-trained deep models that are frequently applied for image classification. The present study further manifests the influence of deep learning architectures to diagnose diseases in the medical field. The proposed binary classification model is trained and tested on 1200 CXRs. The model worked well with a 97.92% accuracy and sensitivity of 99.14%; better than previous studies of the same problem domain.

The design of the rest of the work is represented in the following segments: The introduction part is described in segment 1. Segment 2 reviews the previous studies. Segment 3 explains the methodology and dataset used for the proposed model; also provides a brief understanding of the deep learning technique. Segment 4 displays the outcomes obtained by the proposed model. Segment 5 presents the comparative analysis of the present study to previous research works in the discussion part followed by segment 6 that illustrates the conclusion and future scope.

2 Related Work

Amidst the spread of the COVID-19 infection, researchers have shown their concern in generating several artificial intelligence (AI) based systems for the automatic



(b) chest X-ray samples of COVID-19 infected patients

Fig. 1 The chest X-ray samples in the dataset: **a** CXR sample of healthy person; **b** CXR sample of COVID-19 positive patients

diagnosis of this infection. Machine learning (ML) and deep learning (DL) algorithm-based methods models are among them [3]. While deep learning is a sub-part of ML, the self-learning ability of deep neural networks makes them more popular.

Multiple studies have been conducted so far for the diagnosis of covid infection in CXRs. The strength of transfer learning was presented [4] for the identification of COVID-19 infection on CXRs, authors developed a hybrid model using various pre-trained deep models. Authors in a study [5] proposed a patch-based CNN network that works on a small number of trainable parameters. The authors evaluated the performance of deep CNN for COVID-19 identification on CXRs. A new model convolutional CapsNet [6] was proposed that detects covid-19 from CXRs with capsule networks. The CoroNet [7] deep CNN model can identify COVID-19 disease on CXRs. The proposed model was based on Xception architecture. CNN model CoroDet [8] performed well for the detection of COVID-19 disease and different classes of pneumonia from CXRs and CT scans. Another study [9] used the deep

model to identify COVID-19 infection on CXRs. The suggested neural network classifies CXRs in COVID-19, pneumonia, and healthy samples. A deep LSTM (Long Short-Term Memory) method was recommended [10] to recognize the COVID-19 infection on CXRs. Studies [11, 12] furthermore manifest the prominence of deep learning models in the identification of COVID infection on X-ray scans.

3 Methodology

In this segment, the procedure used for the proposed model and dataset used to develop the proposed model will be discussed. A concise outline of deep learning technology is also included in this section. Following are the sub-parts of this section.

3.1 Dataset

Dataset performs the most important part in deep learning. Deep neural networks work well with the large size of the dataset. Many hospitals have provided the chest X-rays of covid infected patients for research purposes, these datasets are available online. Github and kaggle are online available dataset repositories that are provided chest X-rays of covid infected patients including various others lung diseases also. One can obtain the dataset from these online available data sources.

To conduct this study, chest X-ray images are obtained from kaggle.com. The dataset available in kaggle has CXRs of covid patients, normal CXRs, and chest X-rays of pneumonia patients. Total 2542 chest X-rays (covid and normal) were downloaded from kaggle though the number of images used in this study is 1200 CXRs (600 X-rays of covid positive patients while other 600 are normal X-rays). Table 1 shows the total dataset downloaded and the number of the dataset used in this study i.e 1200 chest X-rays.

The chest X-rays (CXRs) dataset is split into two sub-categories, one is training, and the other is testing. 80% of the images of the dataset are taken for the training purpose of the model and the rest of 20% is used for testing the performance of the proposed model. Therefore 480 CXRs of the covid category are used for training and 120 CXRs of the covid category are used for validation purposes, same implies with the normal CXRs category. In total 960 CXRs are used for the training part and 240

Table 1 Dataset used in proposed study

Total X-ray images	CXRs of covid cases	CXRS of normal cases	CXRs of covid cases used	CXRS of normal cases	Final dataset (CXRs) used in this study
2542	1201	1341	600	600	1200

CXRs are used for the testing part. The sample X-ray scan of COVID-19 infected and normal category is displayed in Fig. 1.

3.2 Deep Learning

Artificial Intelligence (AI) has brought the world to the forefront of technology. With the help of AI, humans can automate the work around them. Through artificial intelligence, an effort is made to make machines such intelligent that they can behave and take decisions like humans. AI has a vast area to perform various applications in it. One of the very important parts of AI is Machine Learning (ML). Such algorithms are developed under machine learning that uses data and learns on their own. The purpose of machine learning is to make the machine so efficient that it does its automatic learning without outside programming. The concept of Artificial Neural Network (ANN) was introduced by Geoffrey Hinton in the 1980s. The way the human brain has a lot of neurons that are connected and exchange narratives and information, an identical concept has been introduced in the ANN where multiple neurons are arranged in different layers. These neurons are mathematical functions applied to data, are connected, and share pieces of information. If we understand the architecture of ANN in a general way, then it has an input layer that takes the input, the information from the input layer is transferred to the hidden layers, where the actual processing of the data happens, and then the final output is transferred to the output layer.

Deep learning is a sub-category of ML which works on the concept of artificial neural networks where multiple neurons are arranged in a layered structure and this structure works like a human brain. Deep learning algorithms imitate the functioning of the human brain to perform complicated tasks automatically. A Deep Neural Network (DNN) is trained from a very large database. The more the data used for training, the more the model will work effectively. During training, deep learning models extract features from data and identify objects with these features.

Automatic feature extraction and learning make deep learning models more powerful. The architecture of the deep network (Fig. 2) is alike the design of ANN with more hidden layers that connect the input and output layers. The larger the amount of hidden layers is used to create a deeper model. Transfer learning is a process that is used when we do not want to train the network from zero; a pre-trained deep model is useful instead to save the processing time. These pre-trained models are already trained on large datasets. The effective working of deep neural network depends upon the number of datasets they are trained and tested on. Augmentation is a technique where some morphological operations are performed on available images to increase the size of the database.

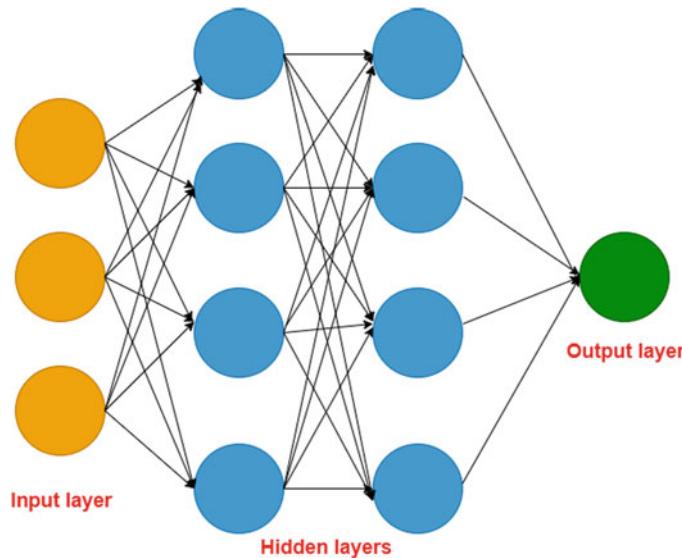


Fig. 2 Architecture of deep neural network

3.3 ResNet-50

In the year 1993, author Pratt [13] shown the power of transfer learning in her study. Transfer learning is the part of ML in which a pre-trained deep learning model is applied as a base model in the development of a new model; instead of starting the new model from scratch. Models that are developed by someone other and trained on different datasets can be used in various studies, having the same problem domain. This saves the time of creating a new model from zero and can help improve the system performance. There are several pre-trained models present that are trained specifically for the classification task.

ResNet-50 (Residual Network) [2] belongs to the ResNet family of neural networks. It is among the most popular pre-trained networks that perform classification on images. ResNet-50 is a CNN model, built in 2015 by Microsoft. The model is trained on over one million images of the ImageNet dataset. This model is consisting of 50 layers which make it deeper. This model can classify up to 1000 objects. This model is trained on colored images with 224×224 pixel quality.

3.4 Proposed Classification Model

The proposed deep convolutional neural network is a binary classification model. The model takes X-ray images of the chest, processes them, and assigns the label to the image after the processing. The labels that can be assigned to input images

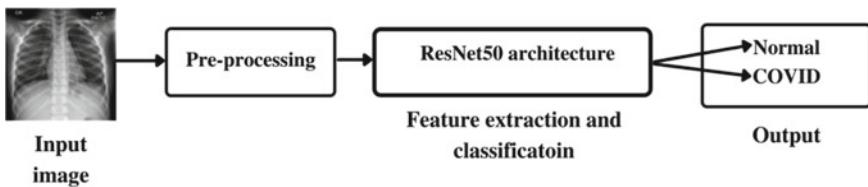


Fig. 3 Workflow of proposed model

are ‘normal’ and ‘covid’ depending upon which category the image belongs to. Proposed model is trained and tested on 1200 CXRs. The dataset has two categories; one category is ‘normal’ having 600 CXRs of a healthy person and the other one is ‘covid’ having 600 CXRs of COVID-19 positive patients. The ResNet-50 pre-trained model is applied as the backbone of the proposed model. The proposed model is executed on the MATLAB 2018a software tool. MATLAB is very popular software for the application of image processing. MATLAB also provides several inbuilt libraries that help in building the models effectively and fast.

The proposed model first applies some pre-processing steps to the input image. The pre-processing step includes the following operations:

- **Resize:** ResNet-50 model is trained on the 224×224 image size therefore the input image is first resized into the required 224×224 size.
- **Noise removal:** Gaussian filter is applied to remove noise from images.
- **Color change:** ResNet-50 network is trained on colored images therefore the input gray-scale X-ray image is converted into an RGB image before processing.

After the pre-processing of the input image, the image is assigned to the first layer of the ResNet-50 model. The first convolutional layer of the model extracts the basic features from the image and assigns the extracted information to the next layer. The batch size for training and testing of the proposed model is set to 16, where the performance of the model is better compared to batch sizes 8 and 32 respectively. The ResNet-50 has 48 convolution layers and one max pool layer and an average pool layer. The final layer of the model has two neurons, presenting two required classification categories as output. According to the features extracted from the network, the label is assigned to the image according to its features of covid or normal CXRs. Figure 3 presents the basic working of proposed model.

4 Results

This segment provides details about the performance measures of the proposed model. The proposed model is trained and tested on 1200 chest X-rays with a split ratio of 80:20. Batch size is the number of samples that are delivered by the system. The higher number of batch sizes requires an increased memory area. The batch size for this model is set on 16; the performance level was not good with batch sizes 8

and 32. To define the performance measures; few terms are defined related to testing of patients that give the condition of the patient that is healthy or unhealthy. These terms are:

- True positive (TP) is number of properly recognized positive cases.
- False positive (FP) is number of incorrectly recognized positive cases.
- True negative (TN) is number of correctly recognized healthy cases.
- False negative (FN) is number of incorrectly recognized cases that are healthy.

4.1 Accuracy

The accuracy of a test is calculated to correctly identify a person's health status that could be healthy or unhealthy.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{TN} + \text{FN}) \times 100 \quad (1)$$

4.2 Sensitivity

The sensitivity describes the capability of a test to recognize the positive cases correctly.

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}) \times 100 \quad (2)$$

4.3 Specificity

The specificity describes the capability of a test to recognize the negative cases correctly.

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}) \times 100 \quad (3)$$

Performance measurements of the presented deep model are shown in Table 2. The suggested model performed well on the given dataset with an accuracy of 97.92%.

Table 2 Performance measures of proposed model

Dataset	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F-1 score
1200 CXRs	97.92	99.14	96.74	96.66	48.94

Sensitivity is the major concern regarding COVID-19 disease. The model achieved a good sensitivity rate of 99.14% though the specificity rate is 96.74%, precision is 96.66% and F1-score is 48.94.

5 Discussion

Deep neural networks are based on the construction of the human mind. The working of deep learning models is faster compare to machine learning algorithms. X-ray machines help radiologists to get the inside picture of the human body. The cheap working cost of X-ray machines makes it the commonly practiced medical imaging method. Chest X-ray can provide internal pictures of the lungs, heart, airways, chest bones, spine bones, and blood vessels and can confirm the presence of fluid or air around the lung. Tuberculosis, pneumonia is dangerous lung diseases. The early and accurate diagnosis of these diseases can help doctors to treat the patient and also can decrease the mortality rate. The proposed classification model is built on the ResNet-50 design. The system is developed to do the binary classification of chest X-rays to identify COVID-19 disease. The proposed model was tested on online available chest X-ray datasets and performed well. The accuracy obtained by our proposed model is 97.92% with a sensitivity rate of 99.14%. The performance level obtained by our proposed deep model has exceeded the studies included in the literature. Table 3 presents the relative performance summary of proposed model by other studies. According to Table 3, it is clearly shown that proposed model has obtained higher accuracy level 97.92%, compared to previous models included in the literature. All studies have used CXR images as input to the model. Though there is a difference in dataset value used by different studies, the ResNet-50 model performed well than VGG19, Xception and ResNet-18.

The results obtained by the proposed model and previous studies have noted the significance of deep neural networks in the medical imaging field. Though there are huge chances of improvement to the proposed model, the main motive behind developing this self-learning classification model is to assist the doctors in this crucial time so that the timely and accurate diagnosis of COVID-19 disease-infected patients can be achieved.

Table 3 Performance analysis of proposed model with other models

Study/ref.	Architecture	Dataset (CXR)	Accuracy (%)
[3]	ResNet-18	502	88.90
[4]	CapsNet	1331	97.24
[7]	CNN	320	97.56
[11]	Xception	2870	85.57
[12]	VGG19	50	90.0
Proposed model	ResNet-50	1200	97.92

6 Conclusion and Future Scope

The novel CORONA VIRUS is a deadly virus that has affected a large population all around the world. The worldwide death rate has increased due to the COVID-19 disease. This study presents a deep neural network based on ResNet-50 architecture. The proposed model is a binary classification model that classifies the CXRs in two categories; ‘normal’ and ‘covid’. The study present that, with the support of the deep learning models, doctors can identify lung infection due to COVID-19 disease. The self-learning nature of the deep learning model makes them superior to other machine learning models. The review of several previous studies in literature has shown that the pre-trained deep learning models can be used in different ways to enhance the performance of models. Deep CNN, hybrid deep learning, data augmentation, and transfer learning are some of the foremost techniques that have been included in various studies. The proposed model is trained on chest X-ray scans and has achieved 97.92% accuracy.

There is a scope of upgrading the accuracy level achieved by the proposed model. For this data augmentation and hybrid deep learning techniques can be applied. Though the sensitivity rate achieved is good but the specificity rate is not high. The performance of the model on a very large number of X-ray datasets is also considerable for the future.

References

1. Bhatt DP, Bhatnagar V, Sharma P (2021) Meta-analysis of predictions of COVID-19 disease based on CT-scan and X-ray images. *J Interdisc Math* 24(2):381–409
2. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
3. Agrawal R, Gupta N (2021) Analysis of COVID-19 data using machine learning techniques. In: Data analytics and management. Springer, Singapore, pp 595–603
4. Jin W, Dong S, Dong C, Ye X (2021) Hybrid ensemble model for differential diagnosis between COVID-19 and common viral pneumonia by chest X-ray radiograph. *Comput Biol Med* 131:104252
5. Oh Y, Park S, Ye JC (2020) Deep learning covid-19 features on CXR using limited training data sets. *IEEE Trans Med Imaging* 39(8):2688–2700
6. Toraman S, Alakus TB, Turkoglu I (2020) Convolutional capsnet: a novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. *Chaos, Solitons Fractals* 140:110122
7. Khan AI, Shah JL, Bhat MM (2020) CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. *Comput Methods Programs Biomed* 196:105581
8. Hussain E, Hasan M, Rahman MA, Lee I, Tamanna T, Parvez MZ (2021) CoroDet: a deep learning based classification for COVID-19 detection using chest X-ray images. *Chaos, Solitons Fractals* 142:110495
9. Haque KF, Haque FF, Gandy L, Abdelgawad A (2020) Automatic detection of COVID-19 from chest X-ray images with convolutional neural networks. In: 2020 International conference on computing, electronics & communications engineering (iCCECE). IEEE, pp 125–130
10. Demir F (2021) DeepCoroNet: a deep LSTM approach for automated detection of COVID-19 cases from chest X-ray images. *Appl Soft Comput* 103:107160

11. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 43(2):635–640
12. 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:2003.11055](https://arxiv.org/abs/2003.11055)
13. Pratt LY (1993) Discriminability-based transfer between neural networks. *Adv Neural Inf Process Syst*, 204–204

IoT-Cloud Enabled Statistical Analysis and Visualization of Air Pollution Data in India



Manzoor Ansari and Mansaf Alam

Abstract Air pollution is a significant issue in our environment. Over the last few decades, urbanization and industrialization have accelerated in developing countries, resulting in an outsized problem of air pollution. In this paper, we have studied the air quality data gathered from IoT-enabled devices and stored onto Cloud storage infrastructure, published by the Indian government Web site. For many Indian states, we have considered the various analysis factors and tried to visualize the IoT-enabled air pollution data, and analyzed the impact of pollution on human life in our society. Furthermore, we have analyzed the air pollution on dataset based on certain conditions using the correlation and heatmap.

Keywords IoT (Internet of Things) · Cloud technology · Air pollution visualization/analysis · Heatmap · Correlation matrix

1 Introduction

Over the last two-decade, air pollution is one of the most serious problems that every nation faces. It is regarded as the sixth most lethal assassin in South Asia. An unprecedented seven million people worldwide are affected by air pollution each year. According to WHO (World Health Organization), statistics show that nine out of ten individuals breathe air that contains high levels of contaminants [1]. Approximately 91% of the world's population lives in places where air quality falls short of WHO standards. About 4.2 million people die each year directly as a result of indoor and outdoor air pollution exposure [2]. Some researchers have discovered a broad range of health consequences in the recent 30 years that are believed to be related to exposure to air emissions. Among them, mostly are respiratory diseases

M. Ansari (✉) · M. Alam
Department of Computer Science, Jamia Millia Islamia, New Delhi, India
e-mail: manzoor188469@st.jmi.ac.in

M. Alam
e-mail: malam2@jmi.ac.in

(including asthma and assessment of pulmonary function, cardiovascular diseases, unfavorable pregnancy effects such as preterm birth), and even death [3]. In this paper, the author discovered air quality data generated by IoT sensors and devices, which were released by India's Ministry of Environment and Forests and Central Pollution Control Board (CPCB) under the National Data Sharing and Accessibility Policy (NDSAP) [4]. This dataset is stored on the cloud storage, the author gathered this IoT-enabled air quality data from the Cloud storage and analyzed it by different artificial intelligence (AI) techniques to identify the effects of particular air pollutants on the people of different states of India. Furthermore, this paper also shows a graphical representation of data in the form of a heatmap and correlation matrix. The following sections describe the structure of this article. Section 2 addressed the many causes of air pollution and its varied forms. Section 3 discusses the researchers' prior work in this field. Section 4 describes the proposed research for air pollution data analysis and visualization which is further separated into three sub-sections: datasets used, proposed workflow diagram, and methodology and technique involved. Section 5 details the experimental design and findings. The conclusion and future directions are discussed in Sect. 6.

2 Background Details

2.1 Cause of Air Pollution

Due to the fact that air pollution may originate from a multitude of sources, many of the primary causes are mentioned. Biomass burning encompasses the combustion of live or dead plants, such as grassland, woodland, and agricultural waste, as well as the combustion of biomass for energy [5]. According to the environmental protection agency, vehicle exhaust contains a variety of contaminants, including NO_x, volatile organic compounds, CO, CO₂, particulates, SO₂, and polycyclic aromatic hydrocarbons (PAHs) [6]. Open burning, which emits carcinogenic compounds, is a significant impediment to India's implementation of an effective urban solid waste management system [7]. There are some known or assumed carcinogens found in diesel exhaust fumes, such as benzene, arsenic, and formaldehyde [8].

2.2 Type of Pollutant in Ambient Air

The various air contaminants present in ambient air are illustrated in Table 1.

Table 1 Type of pollutant in ambient air

Air pollutant	Description
SO ₂	It irritates the nose, throat, and airways, resulting in coughing, wheezing, shortness of breath, or a chest tightness. Sulfur dioxide inhalation has been associated with an increase in shortness of breath and illness, as well as difficulties breathing and chronic disease
NO ₂	It has been shown to produce bronchoconstriction, inflammation, and a decreased immunological response, as well as possible effects on the heart. Irritation and burns can result from direct contact with the skin. It increases the susceptibility of plants to disease and frost damage
Particulates	These are also referred to as atmospheric aerosol particles, particulate matter (PM), or suspended particulate matter (SPM)
Spm (Suspended Particulate Matter)	SPM in the air is used to assess the quality of the air. In the form of suspended particulates, air contains a variety of allergens, fibrous compounds, heavy metals, and even several organic carcinogens
Rspm (Respirable Suspended Particulate Matter)	It is a significant cause of death and morbidity. In the short term, small particles aggravate respiratory and heart problems, while in the long term, they contribute to the development of lung cancer

2.3 Various IoT Sensors and Devices Used for Data Gathering

Table 2 depicts a huge number of air quality sensors/devices that detect multiple hazardous gases in the environment.

3 Related Works

Air contamination has been linked to adverse health consequences. Individuals engage in compensatory practices improving their atmosphere to avoid expensive emissions exposure. The researchers in their research [11] have discussed trends in particle air pollution in three small cities in India over a year. This analysis is crucial for developing state and regional air pollution strategies, as well as SDG goals associated with public health. In [12], the authors suggested Menn-Kendall trend analysis to estimate AOD (aerosol optical depth) trends across India and the percentage rise in AOD, NO₂, and SO₂ during the lockdown. Eastern India has higher levels of air

Table 2 Air pollution IoT sensors/devices

IoT sensors/devices	Description
Grove-multichannel gas sensor [9]	Multiple gas sensor that detects various gases such as carbon monoxide (CO), nitrogen dioxide (NO ₂), ammonia (NH ₃), and methane (CH ₄)
MH-Z19 [10]	This sensor was developed by Winsen Sensors and can detect the CO ₂ concentration in ppm
MQ9	It can detect carbon monoxide (CO), methane (NH ₃), and flammable gas that is suitable for a variety of applications
MQ135	It can detect high sensitivity of ammonia (NH ₃), Sulfur dioxide (SO ₂), and benzene steam, as well as smoke and other hazardous gases
DHT11	It is an ultra-low-cost sensor that can monitors temperature and humidity
ME4-SO ₂	It is used to detect hydrogen sulfide in agricultural areas and the field of environmental conservation

pollution as a result of coal-fired power plants. The researchers study various emissions and mortality relationships. This study used modern satellite data on PM2.5 developments in India. as well, first time in India, by running a fixed-effects model to tackle measurement error and endogeneity issues. The authors in this work [13] studied pollution dispersion using the US Environmental Protection Agency's (EPA) industrial source complex model short-term (ISCST3) air simulation model. This study will assist the petrochemical sector and public health agencies in effectively managing air quality hazards. The study team [14] evaluated annual PM2.5 and BC (black carbon) levels in residential areas using land-use regression models and simulations based on actual measurements from 402 randomly selected individuals. They used linear and logistic mixed models to assess the relationship between exposure variables and health outcomes.

4 Proposed Work

4.1 Air Pollution Dataset Used

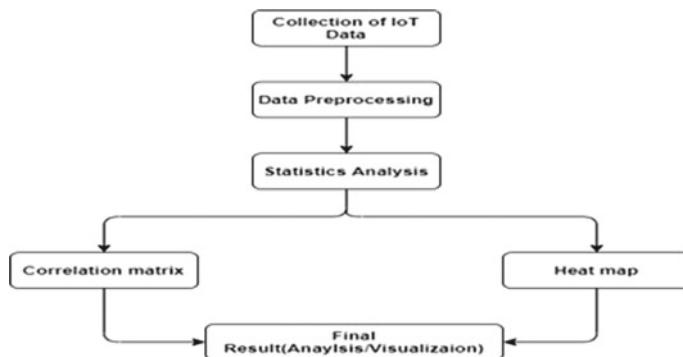
Under the National Data Sharing and Accessibility Policy, the Ministry of Environment and Forests and the Central Pollution Control Board of India have produced and published an IoT-enabled air pollution dataset (NDSAP). This dataset comprises a range of air pollution-related parameters, such as the location, station code, date, and state, as well as a variety of air contaminants that have an effect on human health. The dataset includes the parameters mentioned in Table 3.

Table 3 Description of dataset

Parameters	Description
stn_code	Station identification code. Each station that recorded the data is assigned a code
sampling_date	The time stamp for the data collection
State	It depicts the states for which statistics on air quality are collected
Location	It denotes the city for which statistics on air quality are collected
Agency	The name of the organization that collected the data
Type	The region in which the measurement was taken
SO ₂	Sulfur dioxide concentration as determined
NO ₂	Nitrogen Dioxide concentration determined
Rspm	Respirable Suspended Particulate Matter is quantified
Spm	Suspended Particulate Matter concentrations were determined
Location monitoring station	It specifies the monitoring area's geographic location
Pm2_5	It is the quantity of particulate matter that has been quantified
Date	It indicates the recording date (a more streamlined form of the 'sampling date' function)

4.2 Proposed Workflow Diagram

This study has been conducted systematically, as shown in Fig. 1. The proposed approach begins with the gathering of datasets CPCB (central pollution control board) under the ministry of environment, the forest of climate change [4]. The data collected has been preprocessed to eliminate redundancy. Preprocessing data entails measures

**Fig. 1** Flowchart of the proposed approach

such as data parsing, noise reduction, data cleaning, preparation, and scaling. Additionally, statistical analysis was performed using two distinct approaches: Correlation Matrix [15] and Heatmap [16]. Finally, Statistics and visualization approach was performed to determine the final findings.

4.3 Methodology and Technique Involved

Based on the given dataset, two methods are adopted in this study. Correlation matrix and heatmap matrix.

4.3.1 Correlation Matrix

In statistics, correlation or dependence is any statistical relationship, whether causal or not, between two random variables or bivariate data. A correlation matrix is a table that displays the coefficients of correlation between variables. The correlation between two variables is shown in each cell of the table. A correlation matrix can be used to summarize the data, as an input to a more advanced study, or as a diagnostic for further analyses. Let a and b be two real-valued random variables. The Pearson's correlation coefficient is defined as [15, 17, 18].

$$\rho(a, b) = \frac{\text{cov}(a, b)}{\sigma_a \sigma_b}$$

where $\text{cov}(a, b)$ is the covariance between a and b and σ_a, σ_b the standard deviation of a and b respectively. The formula for ρ can be expressed in terms of mean and expectation.

$$\text{cov}(a, b) = \mathbb{E}[(a - \mu_a)(b - \mu_b)]$$

ρ can also be expressed as below

$$\rho(a, b) = \frac{\mathbb{E}[(a - \mu_a)(b - \mu_b)]}{\sigma_a \sigma_b}$$

where σ_a and σ_b are defined as μ_a is the mean of a , μ_b is the mean of b and \mathbb{E} denote the expectation. The formula for ρ can also be written in the form of expectation.

$$\rho(a, b) = \frac{\mathbb{E}[ab] - \mathbb{E}[a]\mathbb{E}[b]}{\sqrt{\mathbb{E}[a^2] - (\mathbb{E}[a])^2}\sqrt{\mathbb{E}[b^2] - (\mathbb{E}[b])^2}}$$

4.3.2 Heatmap

A heatmap is a two-dimensional data visualization tool that displays the magnitude of a phenomenon as a color scale. The color variation may be by hue or intensity, giving simple visual clues of how the phenomenon is grouped or differs over time.

4.4 *Experimental Setup and Results*

In this section, we have used Python (version 3.7.4), for data visualization and analysis. In this analysis, pandas are used for data gathering and manipulation, and plotly for visualizations. Seaborn is an open-source Python visualization toolkit that utilizes the great library matplotlib and includes built-in support for the Python libraries NumPy and pandas. Seaborn gives us the ability to generate attractive and insightful statistics visualizations. Matplotlib enables a wide range of visualization possibilities, but making them visually appealing is frequently challenging and time-consuming. Seaborn is frequently used to improve the appearance of default matplotlib plots and to create new plot types. Additionally, we discussed how to conduct a visual analysis: histograms for numerical data; count graphs for categorical variables. When using graphs to show relationships between numerical variables, the researchers are most interested in scatter plots, joint plots, and pair plots. The analyst must be able to visualize all of the data's variables and their connections in order to detect emerging patterns, trends and outliers. Once we understand the plot, it will help impact our decisions and improve our quantitative models.

4.5 *Statistical Analysis*

In this section, the authors conducted statistical analyses on the dataset and determined if these characteristics are associated and initiate by plotting the pair plot for each of the features depicted in Fig. 2, the concentrations of SO₂ and NO₂ in the supplied dataset are evenly distributed around the origin, meaning that these are negligible for the vast majority of measurements. In terms of other features, the authors may observe that SO₂ and NO₂ tend to follow a very similar pattern. Although spm and rspm undoubtedly address the linear relationship, the remaining features are not exactly related. Demonstrate correlation matrix in Fig. 3, for a more in-depth analysis. The authors observe any similarity between spm and rspm in the correlation matrix, which supports our study of the pair plot map. Other characteristics exhibit a low degree of correlation

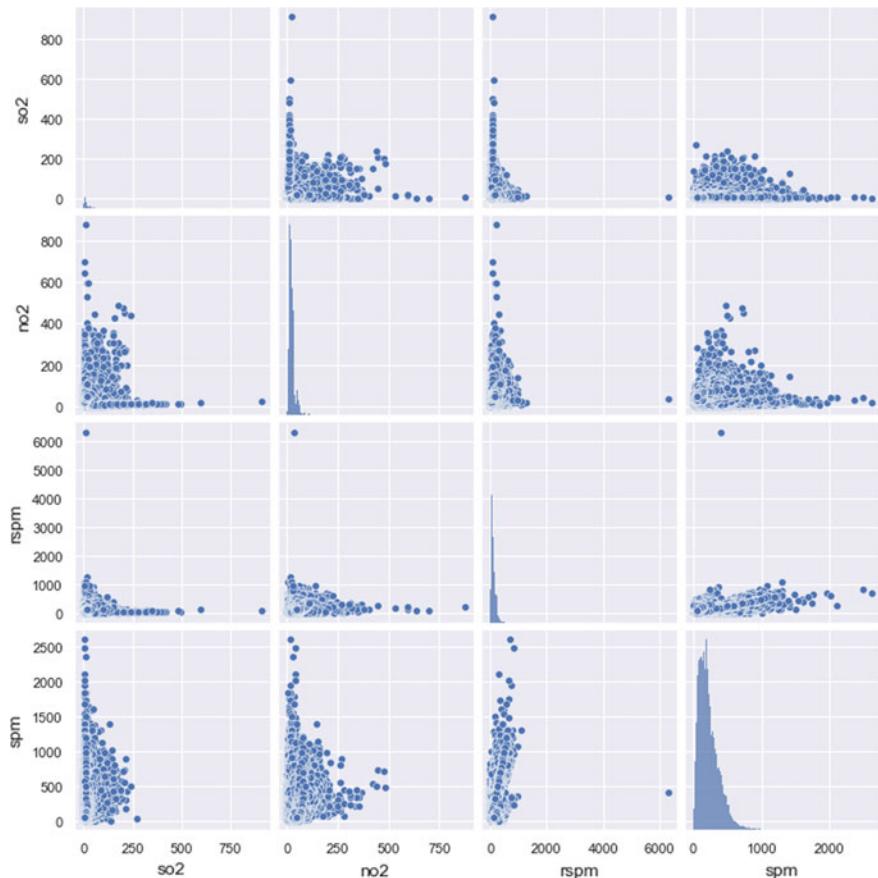


Fig. 2 Pair plot for each column

4.5.1 SO₂ and NO₂ Analysis Using the Heatmap

Figure 4 depicts the rising concentration of SO₂ in Bihar state from 1987 to 1999. Likely also, Gujarat had a high concentration of SO₂ about 1995. Also, in Haryana, the SO₂ level began rising in 1987 and peaked in 2003. SO₂ concentration has grown progressively since 1987. In 1996, SO₂ levels in Puducherry rose significantly. Like in Rajasthan, SO₂ concentrations also spiked in 1987. Uttarakhand has always had a high SO₂ content. SO₂ was constantly high in West Bengal from 1987 to 2000. The data above show that pollution from sulfur dioxide has been prevalent in certain states in the past and is now decreasing (from 2000). In the heatmap in Fig. 5, rows indicate states, and columns indicate years. Regions like Rajasthan, Delhi, Haryana, Bihar, Jharkhand, West Bengal, and Puducherry, have seen significant levels of NO₂. Rajasthan has had annual decreases in the levels of NO₂, whereas other states have experienced increases. Other states, including West Bengal and Jharkhand, have

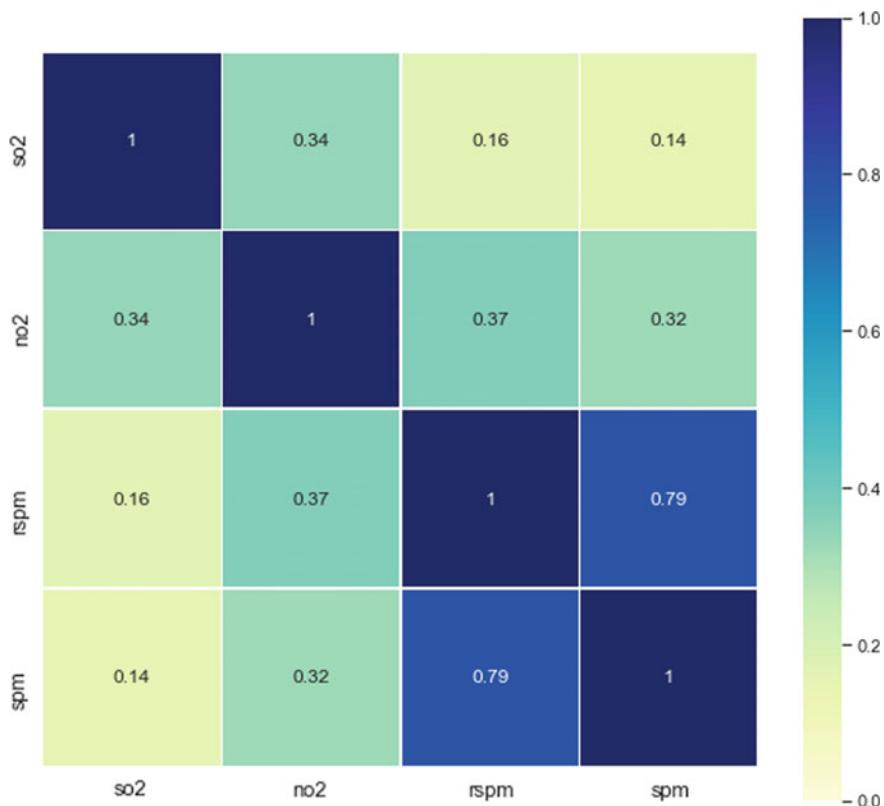
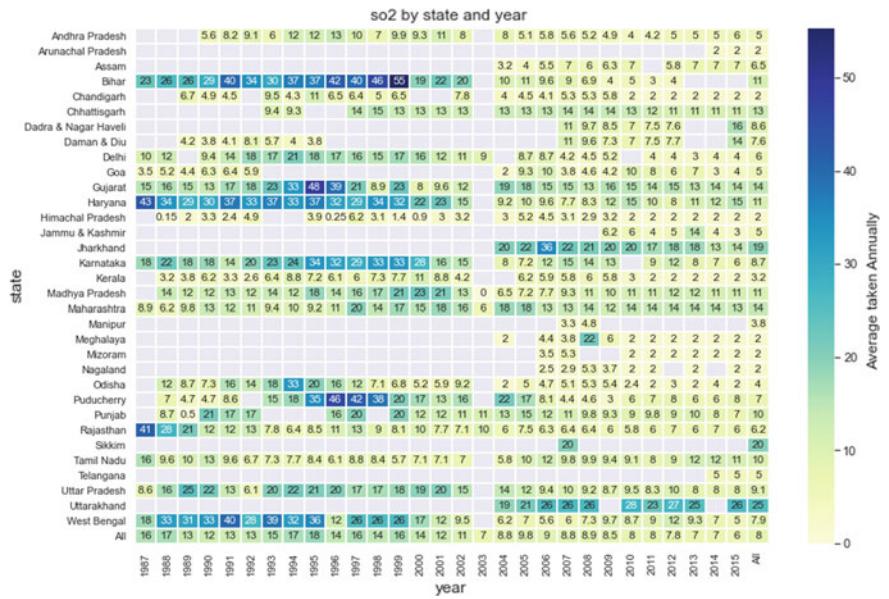
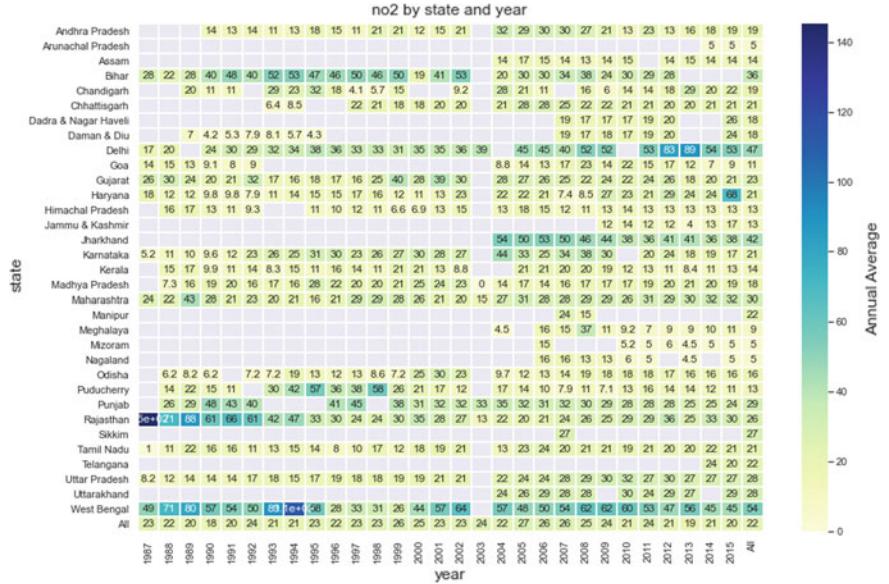


Fig. 3 Correlation matrix for the dataset

maintained high levels of NO₂. On average, across the state, NO₂ concentrations have grown since 2000. An extremely noticeable rise in NO₂ emission occurred throughout South Asia, especially India, between 2005 and 2014, which adversely affected air quality.

4.5.2 Rspm and Spm Heatmap Analysis

In Fig 6, the following row indicates year attribute value whereas column indicates rspm state attribute. Heatmaps are such an essential method for data processing that they simplify the process of analyzing everything. One can see the shifts, the different rspm levels in a state over a year, and so forth. As seen in Fig. 6, states such as Punjab, Delhi, Haryana, Uttar Pradesh, and Jharkhand have experienced elevated rspm volumes. The following heatmap illustrates the relationship between the row

Fig. 4 Heatmap of SO₂ by state and year

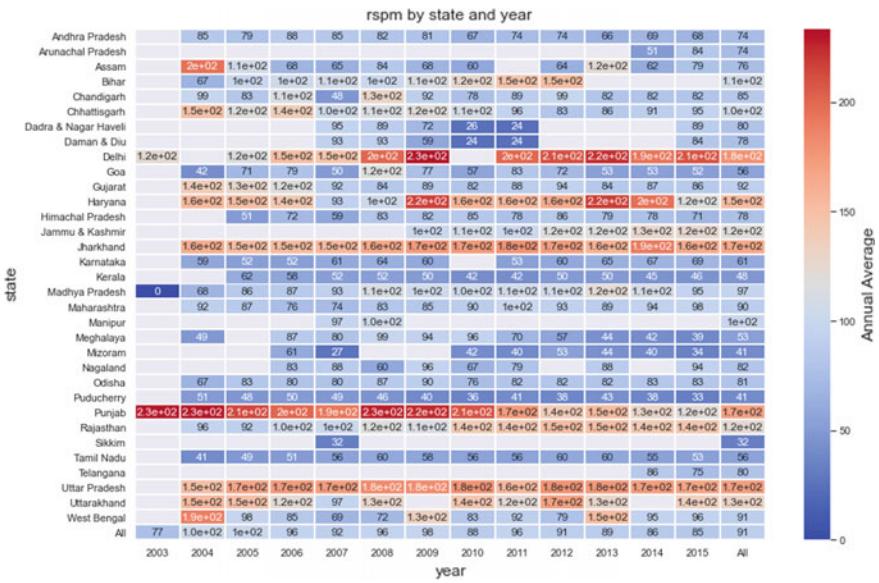


Fig. 6 Heatmap of rspm by state and year

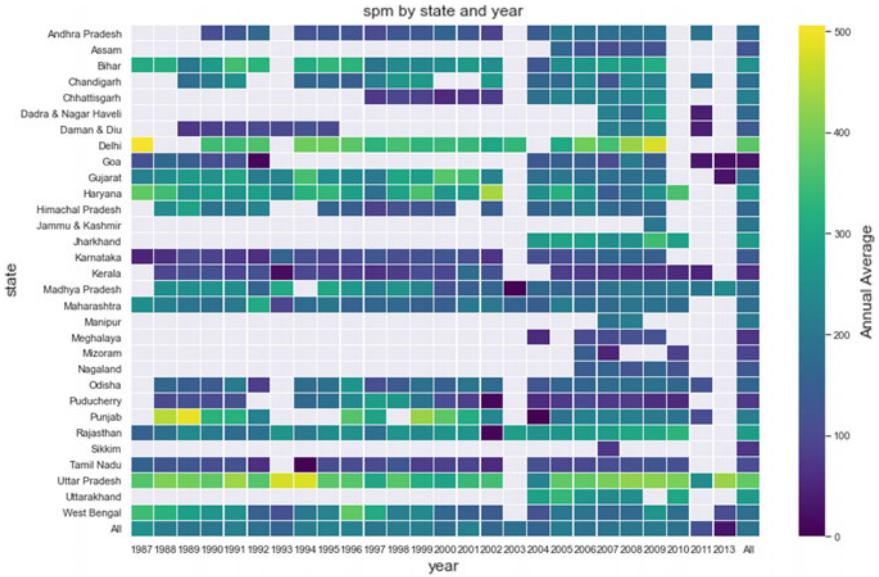


Fig. 7 Heatmap of spm by state and year

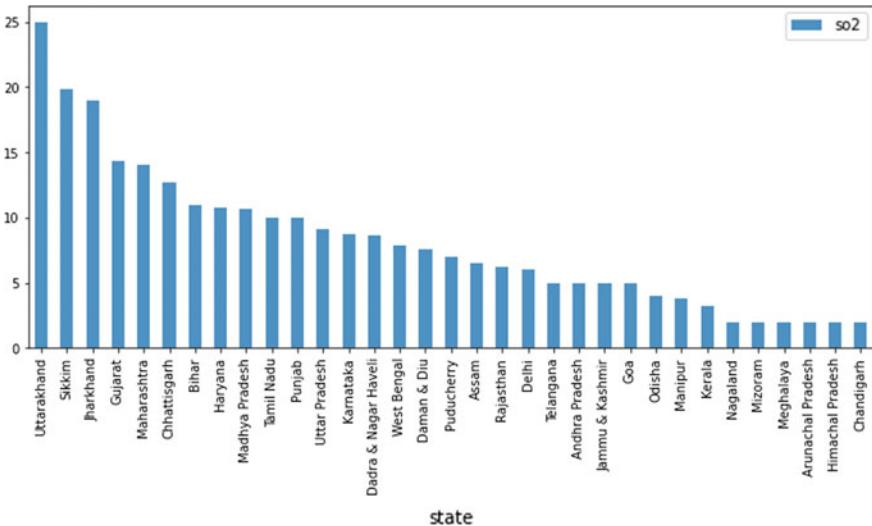


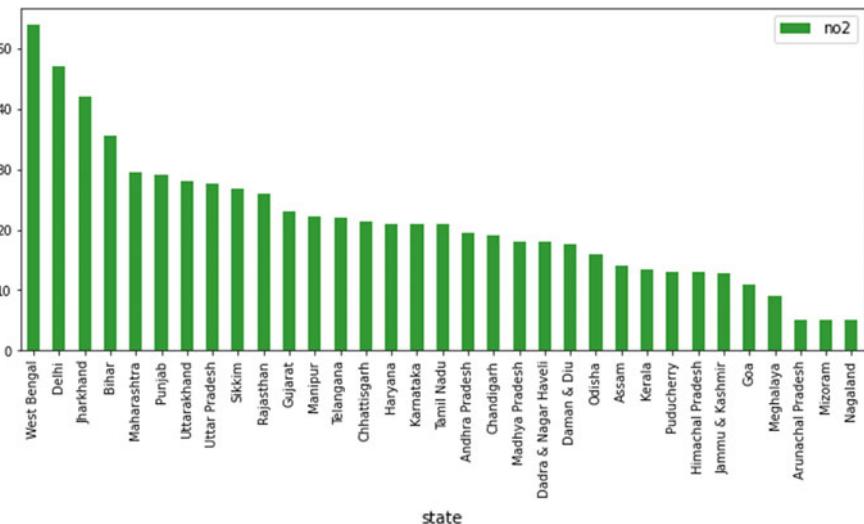
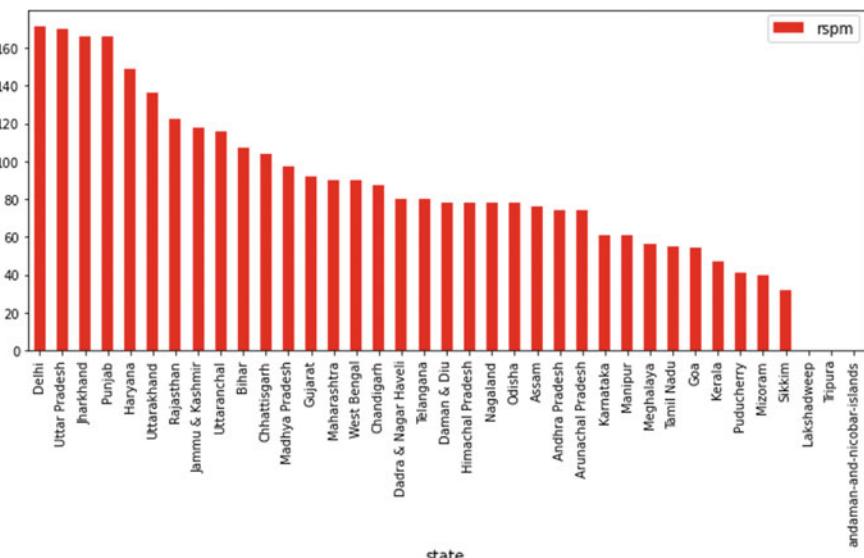
Fig. 8 Bar plot versus SO₂

attribute state and the column attribute year. As shown in Fig. 7, high spm concentrations have exacerbated the situation in states such as Delhi, Haryana, Punjab, and Uttar Pradesh.

4.6 Data Visualization

In Fig. 8, the Bar Plot shows the concentration of SO₂ in various states. As shown in Fig. 8, Chhattisgarh is consistent with the high concentration of SO₂, whereas Uttarakhand is where it is least concentrated. Uttarakhand, Sikkim, Gujarat, Maharashtra, Jharkhand, and Chhattisgarh should take efforts to counteract growing SO₂ levels.

According to Fig. 9, the West Bengal concentration of NO₂ is the greatest, while the concentration in Nagaland is the lowest. Delhi, the country's capital, ranks second, while Jharkhand ranks third. It's unsurprising, given that Delhi has made headlines many times in recent years due to air pollution, especially NO₂ concentrations. As shown in Fig. 10, Delhi's rspm level is surprisingly high, given the prevalent pollution that India has experienced over the past few years. It is an extremely important issue when it comes to air pollution. Massive suffering and thousands of deaths have been caused by the enormous increase in pollution levels in Delhi. This fact is evident from the preceding figure, since we can see that Uttar Pradesh is about equidistant from Delhi. Because the state of Uttar Pradesh is the most populated in the world, the air quality is extremely hazardous.

**Fig. 9** Bar plot NO₂ versus state**Fig. 10** Rspm versus state

We need to take immediate measures to deal with the growth in rspm levels, especially in UP, which has almost 20 crore people as its residents. Sikkim, on the other hand, has the lowest rspm concentration, followed by Mizoram and Puducherry. In

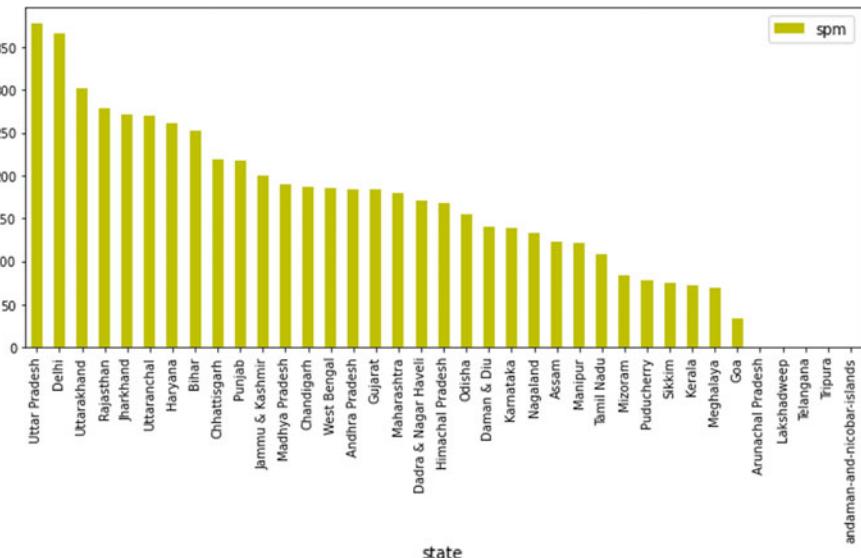


Fig. 11 Bar plot spm versus state

Fig. 11, the Delhi and UP tandem have reclaimed the leading position. The concentrations of spm and rspm in Uttar Pradesh and Delhi are comparable. The latest study claims that a suspended particulate matter (spm) level of 2339 g/cum was recorded in Lucknow, the capital of Uttar Pradesh, in 1997, well beyond the levels recommended for residences and industrial zones. This is the highest amount ever recorded in Delhi, which is surpassing the previous peak of 2340 g/cum which was measured in 1992.

5 Conclusions and Future Directions

As a result of the above research, we can deduce that the northern states of India are the most severely impacted by air pollution. States such as Punjab, Delhi, Haryana, and Uttar Pradesh are extremely contaminated and urgently need measures. Additionally, we have observed that even in states with high pollution levels, many regions inside the state remain unpolluted. Similarly, statistical pair plots have revealed that states with high rspm concentrations frequently have high spm concentrations. Using the heatmap, we have determined that during the early stages of contamination in Puducherry and Bihar, both states were highly contaminated, but have since been cleaned up.

In the future, we will apply certain prediction models (machine learning algorithms) to forecast the effect of air quality on human life based on a given dataset (state-by-state, month-by-month, year-by-year), and we will also forecast the quality

of the air (amount of air pollutant present in the air) to take appropriate measures to minimize air pollution in a specific state.

References

1. Air pollution (2021) Retrieved 13 Feb 2021, from https://www.who.int/health-topics/air-pollution#tab=tab_1
2. Ambient (outdoor) air pollution (2021) Retrieved 13 Feb 2021, from [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
3. Air Pollution (2021) Retrieved 13 Feb 2021, from <https://www.niehs.nih.gov/health/topics/agents/air-pollution/index.cfm>
4. Central Pollution Control Board. Retrieved 11 April 2021, from <https://cpcb.nic.in/air-pollution/>
5. Wu J, Kong S, Wu F, Cheng Y, Zheng S, Qin S, Liu X, Yan Q, Zheng H, Zheng M, Yan Y, Liu D, Ding S, Zhao D, Shen G, Zhao T, Qi S (2020) The moving of high emission for biomass burning in China: view from multi-year emission estimation and human-driven forces. Environ Int 142:105812
6. Lu J (2011) Environmental effects of vehicle exhausts, global and local effects: a comparison between gasoline and diesel
7. Krecl P, de Lima CH, Dal Bosco TC, Targino AC, Hashimoto EM, Oukawa GY (2021) Open waste burning causes fast and sharp changes in particulate concentrations in peripheral neighborhoods. Sci Total Environ 765:142736
8. American Journal of Respiratory and Critical Care Medicine (2012). <https://www.atsjournals.org/doi/full/>; <https://doi.org/10.1164/ajrccm.183.10.1437>
9. Zuo B (2021) Grove—Multichannel Gas Sensor—Seeed Wiki. Wiki.seeedstudio.com [Online]. Available: https://wiki.seeedstudio.com/Grove-Multichannel_Gas_Sensor/. Accessed 24 Apr 2021
10. Winsen-sensor.com (2021) [Online]. Available: <https://www.winsen-sensor.com/d/files/PDF/Infrared%20Gas%20Sensor/NDIR%20CO2%20SENSOR/MH-Z19%20CO2%20Ver1.0.pdf>. Accessed 24 Apr 2021
11. Agrawal G, Mohan D, Rahman H (2021) Ambient air pollution in selected small cities in India: observed trends and future challenges. IATSS Research
12. Tyagi B, Choudhury G, Vissa NK, Singh J, Tesche M (2021) Changing air pollution scenario during COVID-19: redefining the hotspot regions over India. Environ Pollut 271:116354
13. Lin YC, Lai CY, Chu CP (2021) Air pollution diffusion simulation and seasonal spatial risk analysis for industrial areas. Environ Res 194:110693
14. Curto A, Ranzani O, Milà C, Sanchez M, Marshall JD, Kulkarni B, Bhogadi S, Kinra S, Wellenius GA, Tonne C (2019) Lack of association between particulate air pollution and blood glucose levels and diabetic status in peri-urban India. Environ Int 131:105033
15. Kohonen T (1972) Correlation matrix memories. IEEE Trans Comput 100(4):353–359
16. Wilkinson L, Friendly M (2009) The history of the cluster heat map. Am Stat 63(2):179–184
17. Benesty J, Chen J, Huang Y, Cohen I (2009) Pearson correlation coefficient. In: Noise reduction in speech processing. Springer, Berlin, Heidelberg, pp 1–4
18. Bartko JJ (1966) The intraclass correlation coefficient as a measure of reliability. Psychol Rep 19(1):3–11

Dental Cavity Detection Using YOLO



Apurva Sonavane and Rachna Kohar

Abstract Oral health diseases are very usual diseases as well as most human beings suffer from this. Because of destitution or unhygienic habits, these oral issues are frequent, also it is projected that 4.6% of total medical expenditure in the world is on the oral health domain. In this study, our main focus is on detecting cavities. Recent advancements in Machine Learning and Artificial Intelligence have contributed very much to the medical domain. With the help of these algorithms, diagnosis, as well as treatment for most of the diseases, have been done easily. To find out dental carries, various imaging techniques are used by dental experts and doctors, though, in this study, we have used RadioVisioGraphy images and used You Look Only Once (YOLO) to detect the dental cavity. We have used a dataset which is gathered from SMBT DENTAL COLLEGE & HOSPITAL, and after training the model with YOLO we were able to achieve 87% accuracy where threshold is set to the 0.3.

Keywords Dental cavity · Object detection · Classification · YOLO

1 Introduction

Oral health diseases and conditions are some of the most affected diseases in the world. As per (Oral health n.d.) [26], 3.5 billion people are suffering from oral diseases which include dental caries, oral cancer, etc. However, most of them are preventable if treated at an early stage. It is still estimated that 4.6% of world medical expenditure corresponds to oral health [16]. Today, a completed oral diagnosis requires many tests and examinations, and this paper focuses on dental caries classification using X-ray. Figure 1 shows the percentage of the population affected by dental caries. With the advancement of technology and medicine, today medical field is taking an advantage of new concepts and algorithms. Machine Learning is a topic that has revolutionized the way to reach a solution in a comparable time to the standard algorithm. Many imaging modalities exist which can be used for

A. Sonavane (✉) · R. Kohar
SCSE, Lovely Professional University, Jalandhar, India

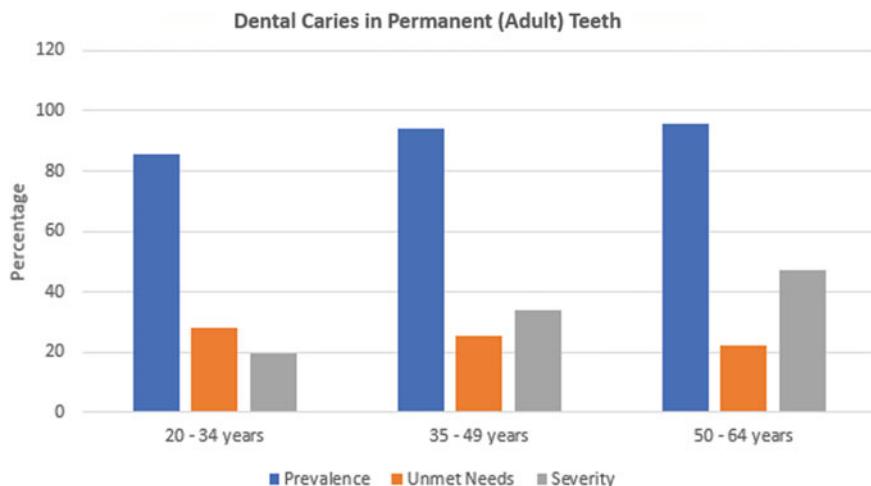


Fig. 1 Percentage of population affected from dental caries in the different age group

dental classification like charge-coupled device (CCD), photostimulable phosphor plates (PSP), and cone-beam CT images, X-ray [31]. Figure 2 shows the imaging modalities which are used for dental classification [32, 33].

Oral diseases are painful, uncomfortable, and in some cases deadly, but with improvement in medicine, these problems are resolved fairly adequately than before. However, there is also an increase in cost associated with oral treatments which results in not getting treatment at the proper time. Poor countries face this effect more due to inadequate medical facilities, non-availability of experts, and more. In this paper, we have used X-ray images which are easy and cheap to obtain than CT images. The dataset which is used in this experiment is gathered from SMBT DENTAL COLLEGE & HOSPITAL, Sangamner, Maharashtra. Due to the availability of huge data, new algorithms and techniques from the Machine learning field can be applied to it to get meaningful results. Researchers have experimented on this using support vector machine (SVM), decision tree, and different variants of artificial neural network [6, 14, 19, 30, 38]. This paper focuses on YOLO for classification.

While using X-ray we noticed that the image suffers from low contrast which makes it difficult for human perception and further image processing steps [5]. To resolve this issue, we have used the image enhancement algorithm CLAHE (Contrast Limited Adaptive Histogram Equalization). To check the quality of output we have used the PSNR value of CLAHE, Histogram Equalization (HE), Recursive Mean-Separate Histogram Equalization (RMSHE) after enhancement and found out CLAHE performs better than the rest for our application [2]. The program is coded in python and a clip limit of 0.2 is used in CLAHE [1, 9].

In the dataset image, if we observe there is more than one cavity in some images. Instead of classifying images as with a cavity and without a cavity, it is easy to just highlight or identify cavities in an image. This is an object detection approach. So,

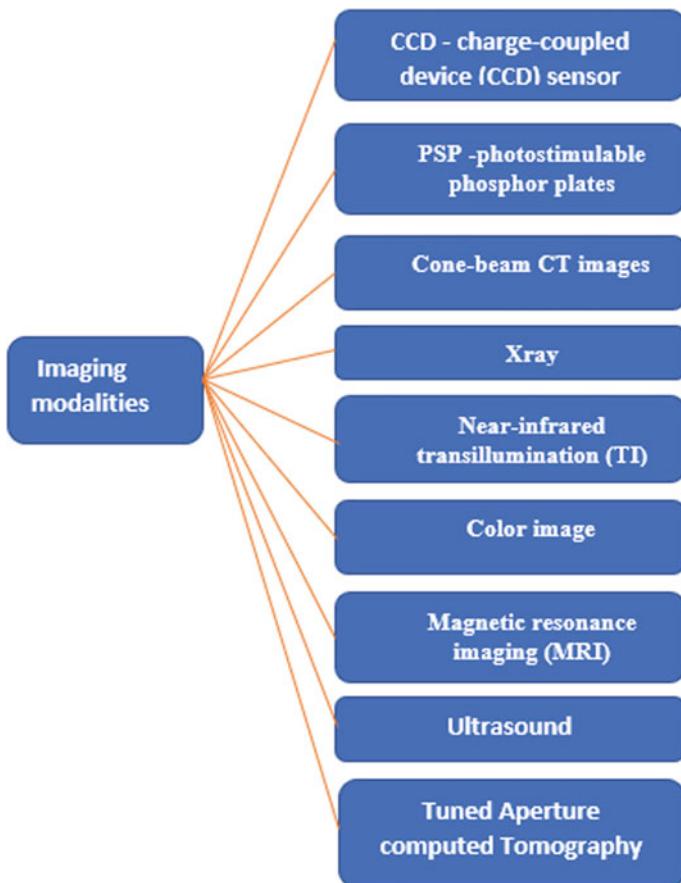


Fig. 2 Imaging modalities used for dental caries detection

we used YOLO for object detection purposes. YOLO is used for object detection as well as recognition purposes. And it has the capability of detecting 45 frames/s as well as 2000 bounding boxes/s.

The research objective of this study is to develop a method to perform dental cavity detection and to test, validate and evaluate developed method performance. Paper organization is as follows:

We have reviewed research papers and work in the Literature review in Sect. 2. We have discussed dataset, image preprocessing (image labeling and annotation), and YOLO algorithm with the workflow of the case study as Methodology in Sect. 3. Outcomes of dental cavity detection have been discussed in Sect. 4. And finally, we concluded our paper in Sect. 5.

2 Literature Review

In this section, we have reviewed the papers which have used different techniques for image enhancement, different classification algorithms used by multiple authors using machine learning and deep learning techniques.

While using an annotated dataset containing 3000+ radiographs, the authors of the paper [37] have presented a deep fully convolutional neural network (FCNN) having 100+ layers. They created a CAD system that works as a second opinion to the dentist. For their system, they have used 5% of the training dataset ad validation data. They verified their results with 3 dentists and showed better results in parameters (sensitivity and F1-score). The authors of the paper [15], have presented a Deep CNN for their experiment. Using 3000 periapical radiographic images, and dividing them into training (80%) and testing (20%). They used GoogleNet Inception v3 which is a pre-trained network and implemented it for preprocessing and transfer learning. The classification accuracy of premolar and molar is found to be 89.0 and 88.0%. They verified their results using parameters accuracy, sensitivity, specificity, positive predictive value, negative predictive value, ROC, and AUC [17] implemented CLAHE for contrast enhancement in Mammograms. They used 20 images from the Mini-MIAS dataset. Median filter, Min-max filter, and Wiener filter were used for noise removal. Their paper showed that using wiener filter with CLAHE showed the best results with RMSE value between 0.586 and 0.7294 and PSNR between 49.5352 and 50.8783 [2]. While working on different medical images presented a combination of CLAHE with 2D discrete wavelet-based fusion technique for efficient results. To compute the performance calculated the SNR and entropy of their finding showed that entropy was increased due to CLAHE and neared maximum and average entropy [29]. Working on the DDSM dataset presented a self-adjusted mammogram solution as Adaptive clip limit CLAHE (ACL-CLAHE). While comparing their work with basic CLAHE their method presented better performance. Bhat and Tarun [3] purposed a two-step process in which first they applied the CLAHE using estimated clip size and secondly chose the optimum clip size based on AMBE and PSNR values. They used STARE, DDSM, and OASIS for their analysis, and the SSIM index showed better results than the traditional CLAHE [13]. While working on mammograms from the MIAS dataset purposed a novel fuzzy clipped CLAHE which automates the clip limit size for enhancement. They presented a fuzzy rule-based flexible system that updates the control parameters. They compared the algorithm performance on CII, DE, AMBC, PSNR and showed improved results than traditional algorithms. Pawar and Talbar [25] presented a DWT coefficient fusion based on local entropy maximization algorithm in the fused original image with corresponding CLAHE image at different levels after decomposition using Haar wavelet. They tested their work on the TMCH dataset comprising 322 images and showed improvement that BBHE and CLAHE. Nababan et al. [21] presented a paper using Evolving Connectionist Systems (ECoS) for detection of breast cancer in women. In their method, they implemented CLAHE after segmentation. For textual features extraction, they used the GLCM matrix. 75.00% sensitivity and 88.89% specificity are achieved on

the INbreast dataset (410 images) and 96.20% sensitivity and 99.24% specificity in the Wisconsin Breast Cancer dataset (699 images).

Also using CNN, the authors of the paper [34] have implemented their experiment on 251 RVG x-rays. They classified their dataset into 3 classes and presented 3 different CNN architectures for classification along with transfer learning for better results. While optimizing their CNN using dropout layers for overfitting, they have used transfer learning for feature extraction and fine-tuning. While testing each model and training each one for 35 epochs they were able to achieve 88.46% accuracy. Singh and Sehgal [35] presented a novel approach based on CNN-LSTM for classification. Their system classifies into 6 classes based on caries location. Their CNN model was used for feature extraction. Using the Dragonfly optimization technique, they were able to achieve 96.0% accuracy. They compared their method with pre-trained networks like Alexnet, GoogleNet, and found their method performs better than these two. Casalegno et al. [4] presented a CNN method with a new imaging technique known as near-infrared transillumination (TI). While using 185 samples, their methods mean IOU score was 72.7%. Their CNN model resembled U-Net [22]. They discussed the advantages of using deep learning and resolved the drawbacks in their case by data augmentation and batch normalization for overfitting issues. Finally, to validate its effectiveness they compared the result with another CNN (DeepNet) and found better results. While using BPNN, authors [8] have achieved an accuracy of 97.1%. They used 105 sample images and used 10-fold cross-validation for their NN.

While authors of the paper [11] concluded in their experiment of dental caries classification that age is the most relevant variable in dental diseases. They conducted their experiment using the 2015 National Health and Nutrition Examination Survey dataset. They used multiple algorithms to select the best relevant feature and applied SVM. With their method, they were able to achieve accuracy, precision, sensitivity, and specificity of 97.1, 95.1, 99.6, 94.3%. Olsen et al. [23] also used machine learning techniques for their experiment in the classification of dental caries. First, they extracted 7 features from the image. Second, they segmented the image into various regions and used the different regions for training and testing. Finally, they used the C4.5 decision tree and 96.62% of pixels were correctly classified.

The authors of the paper [27], have presented an International Caries Detection and Assessment System (ICDAS) system and its management. ICDAS is a trademarked product for dentists which assists them in recode keeping and dental caries classification. Based on the evidence approach their system classifies cavity based on historical data and presented their work clinical practice, education and research, and public health. With a decade of information feed into it, it supports multiple formats and decision-making skills while being in up-gradation for a friendly user interface. However, there are multiple dental caries classification and management applications which are published by many dental organizations like ICCMS [12], DMF-T [18], ICDAS 2 [20]. While authors of the paper [7] presented a smartphone application using SVM as their classifier, their model used color photographs taken from a smartphone which is minimalistic in cost and classifies into 3 output classes.

Table 1 Literature review of related work

Author	Dataset	Methods used	Remarks
Srivastava et al. [37]	3000+ radiographs	FCNN	<ul style="list-style-type: none"> Created a CAD system Recall-80.5 and F1-score-70
Prajapati [34]	251 RVG X-ray	CNN	<ul style="list-style-type: none"> 3 CNN architectures Transfer learning for feature extraction and fine-tuning
Casalegno et al. [4]	185 samples	CNN	<ul style="list-style-type: none"> TI images Used data augmentation and batch normalization IOU score 72.7%
Olsen et al. [23]	40,000 pixels (6 images)	C4.5 decision tree	<ul style="list-style-type: none"> 7 features extracted, image segmented into 7×7 pixel blocks
Sonavane et al. [36]	Kaggel dataset (74 Images)	CNN	<ul style="list-style-type: none"> Accuracy 71.43%

While using only 620 images and ICDAS II codes for evaluation, their model was able to achieve 92.37, 96.6, and 88.1% of accuracy, specificity, and sensitivity (Table 1).

3 Methodology

3.1 Dataset

This study has been performed on RGV (RadioVisioGraphy) image dataset collected at SMBT DENTAL COLLEGE & HOSPITAL, Sangamner, Maharashtra. RGV imaging presented using 256 gray shades. The dataset contains a total of 800 images. Out of 800 images selected image count is 250 of which 200 training images and 50 for testing. Every single image has at least one cavity or more than one. Figure 3 shows the snapshot of the dataset.

3.2 Image Labeling and Annotation

We have used YOLO for cavity detection as an object detection model. So, this needs to be labeled and annotated dataset. For labeling, LabelImg is used and for the annotation step, Bounding boxes are used. LabelImg generates a vector that contains object or class name, x -axis, y -axis, upper-left corner, lower-right corner of the bounding box as shown in Fig. 4.

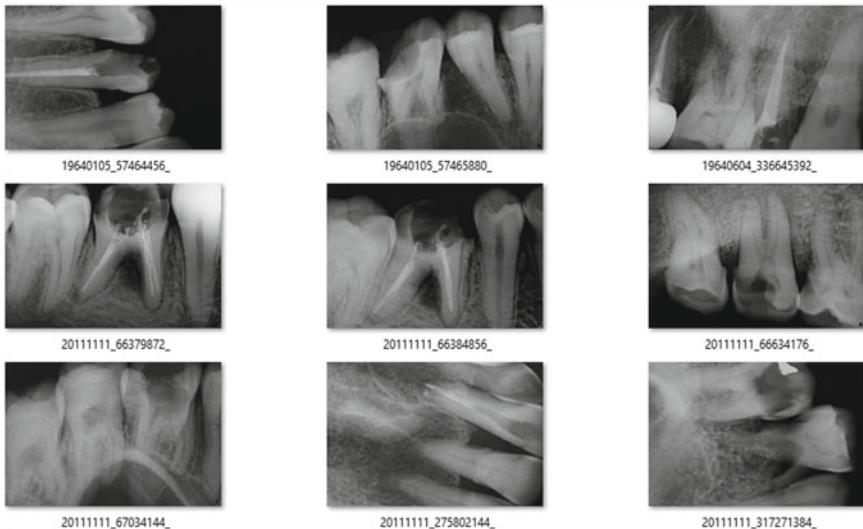


Fig. 3 Snapshot of the dataset

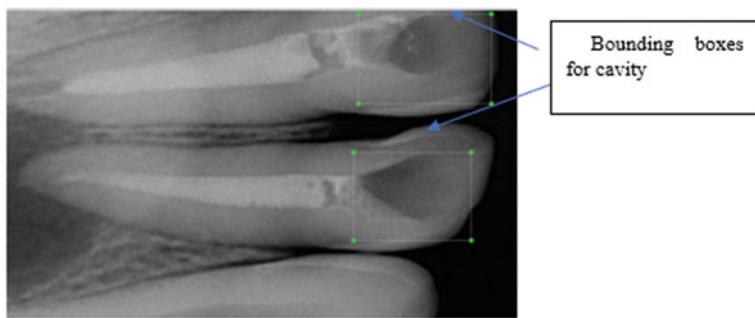


Fig. 4 Image labeling with bounding boxes in LabelImg Tool

3.3 YOLO V4

YOLOV4 [10] came into existence after doing an improvement on YOLOV3 [28]. YOLOV4 is an object detection model with some modifications in its architecture. As in a simple object detection model, it could consist of one-stage detector or two-stage detectors. One-stage detector is fast and could perform faster detection, whereas two-stage detectors first analyze the position and then classifies the object. In object detector models, an image is passed which goes into the ‘Backbone’ of the architecture, then passes through neck, dense prediction, and sparse prediction. In the case of YOLOV4, the backbone consists of many convolution layers, mainly of three parts as bag of freebies, bag of specials, and CSPDARKnet53. All these three

layers perform tasks like data augmentation, pixel detection, label smoothing and many more things. Then after the backbone of the architecture of YOLOV4, there is neck (detector), whose main feature is to collect different feature maps which it acquires from different bottom-up and top-down paths. After Neck next part in this architecture is Head (detector), whose main role is to perform dense predictions. It consists of vectors which are responsible for the prediction of the bounding boxes which are formed whenever an image is identified for which the model was trained. So, mainly backbone, neck, and head are the key performers, in which there are several more small architectures working synchronously with them. Mean average precision in the neck was increase by 10 and 12% more frames which were able to identify accurately. So, YOLOV4 is able to do single-stage (one-stage) prediction known as dense prediction and multi-stage (two-stage) prediction known as sparse prediction. Step-wise procedure for this study as shown in Fig. 5.

In our custom training configuration file, we have set some parameters as per requirement of system. Our images are gray-scaled images, so, we have set channels as 1. Batch (no. of sample images per batch) as 32 and subdivision as 64.

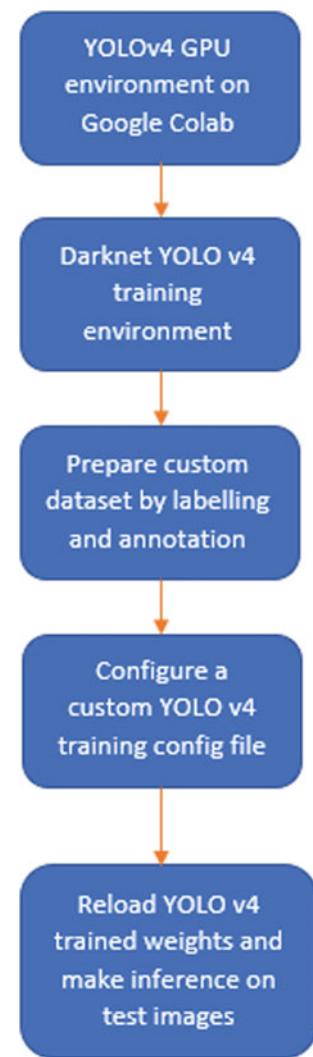
Minimum system requirement:

- CMake \geq 3.8
- CUDA 10.0
- OpenCV \geq 2.4
- cuDNN \geq 7.0 for CUDA 10.0
- GPU with CC \geq 3.0.

4 Result and Discussion

The pre-trained YOLO framework is for colored images. In this study, we have applied YOLO on gray-scaled images, i.e., RVG by initializing channel as 1 in the configuration file. YOLO is a precise object detector and fast, this makes YOLO perfect for applications of computer vision. In this study of dental cavity detection, our YOLO framework was formed with a threshold of 0.3. By default, YOLO only shows objects detected if confidence \Rightarrow 0.25. We can set another confidence value by passing the- *thresh<val>* flag to the YOLO command. However, the accuracy can be increased by maximizing the image dataset. And after training the model with YOLO, we were able to achieve 87% accuracy. For Fig. 6 if we observed it shows cavity is 0.73 means it states that detection of cavity in input image is 73%. As per our literature review, we have got highest accuracy in our detection system.

Fig. 5 Flowchart of steps involved in the study



5 Conclusion

With the advancement in technologies, the medical and health domain can also make progress in the diagnosis and treatment of various diseases. With the help of advanced imaging techniques in the medical domain, a large image dataset can be collected and this can help to train models for diagnosing or finding the disease easily using Machine learning and AI algorithms.

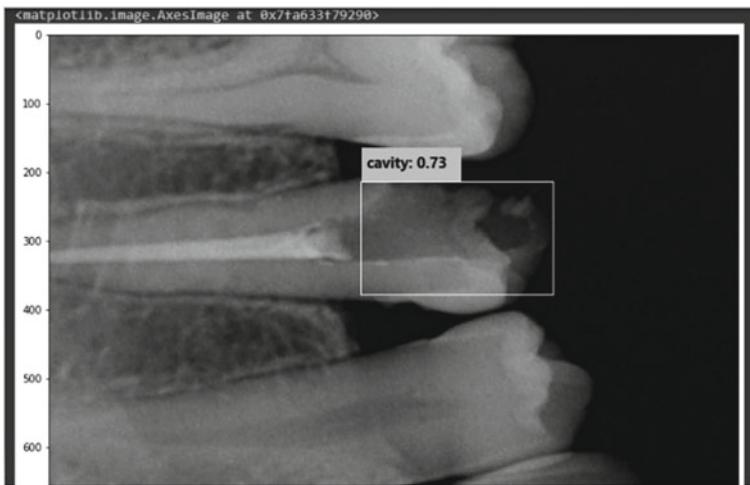


Fig. 6 Sample output image

We have used a dataset which was gathered from SMBT DENTAL COLLEGE & HOSPITAL, and after training the model with YOLO, we were able to achieve 87% accuracy where threshold was set to the 0.3.

This may help to create whole automated methods which are assisting doctors in a particular domain. This study shows that mobile or web applications can also be created to detect cavities directly for RGV images at hospitals, meanwhile, with an increase in the dataset, the accuracy of the model will increase.

References

1. Alzubi OA et al (2020) An optimal pruning algorithm of classifier ensembles: dynamic programming approach. *Neural Comput Appl* 32(20):16091–16107. <https://doi.org/10.1007/s00521-020-04761-6>
2. Bhan B, Patel S (2017) Efficient medical image enhancement using CLAHE enhancement and wavelet fusion. *Int J Comput Appl* 167(5):1–5
3. Bhat M, Tarun PMS (2015) Adaptive clip limit for contrast limited adaptive histogram equalization (CLAHE) of medical images using least mean square algorithm. In: Proceedings of 2014 IEEE international conference on advanced communication, control and computing technologies, ICACCCT 2014 (978), pp 1259–1263
4. Casalegno F et al (2019) Caries detection with near-infrared transillumination using deep learning. *J Dent Res* 98(11):1227–1233
5. Chen H, Rogalski MM, Anker JN (2012) Advances in functional X-ray imaging techniques and contrast agents. *Phys Chem Chem Phys* 14(39):13469–13486
6. Dong M et al (2015) An efficient approach for automated mass segmentation and classification in mammograms. *J Digit Imaging* 28(5):613–625
7. Duong DL, Kabir MH, Kuo RF (2021) Automated caries detection with smartphone color photography using machine learning. *Health Inf J* 27(2):146045822110075

8. Geetha V, Aprameya KS, Hinduja DM (2020) Dental caries diagnosis in digital radiographs using back-propagation neural network. *Health Inf Sci Syst* 8(1):8
9. Gupta D et al (2020) Usability feature extraction using modified crow search algorithm: a novel approach. *Neural Comput Appl* 32(15):10915–10925. <https://doi.org/10.1007/s00521-018-3688-6>
10. Huang Z et al (2020) DC-SPP-YOLO: dense connection and spatial pyramid pooling based YOLO for object detection. *Inf Sci* 522:241–258. <https://doi.org/10.1016/j.ins.2020.02.067>
11. Hung M et al (2019) Application of machine learning for diagnostic prediction of root caries. *Gerodontology* 36(4):395–404
12. Ismail AI, Nigel BP, Marisol T, Authors of the International Caries Classification and Management System (ICCMS) (2015) The international caries classification and management system (ICCMS™) an example of a caries management pathway. *BMC Oral Health* 15(1):S9
13. Jenifer S, Parasuraman S, Kadhirvelu A (2016) Contrast enhancement and brightness preserving of digital mammograms using fuzzy clipped contrast-limited adaptive histogram equalization algorithm. *Appl Soft Comput* J
14. Král P, Lenc L (2016) LBP features for breast cancer detection. In: 2016 IEEE international conference on image processing (ICIP), pp 2643–2647
15. Lee JH, Kim DH, Jeong SN, Choi SH (2018) Detection and diagnosis of dental caries using a deep learning-based convolutional neural network algorithm. *J Dent* 77(June):106–111
16. Listl S, Galloway J, Mossey PA, Marenes W (2015) Global economic impact of dental diseases. *J Dent Res* 94(10):1355–1361
17. Makandar A, Halalli B (2015) Breast cancer image enhancement using median filter and CLAHE. *Int J Sci Eng Res* 6(4):462–465
18. Melgar RA et al (2016) Differential impacts of caries classification in children and adults: a comparison of ICDAS and DMF-T. *Braz Dent J* 27(6):761–766
19. Merati M, Mahmoudi S, Chenine A, Chikh MA (2019) A new triplet convolutional neural network for classification of lesions on mammograms. *Egypt J Radiol Nuclear Med* 33(3):213–217
20. Mitropoulos P, Rahiotis C, Stamatakis H, Kakaboura A (2010) Diagnostic performance of the visual caries classification system ICDAS II versus radiography and micro-computed tomography for proximal caries detection: an in vitro study. *J Dent* 38(11):859–867
21. Nababan EB, Iqbal M, Rahmat RF (2017) Breast cancer identification on digital mammogram using evolving connectionist systems. In: 2016 International conference on informatics and computing, ICIC 2016 (ICIC), pp 132–136
22. Navab N, Hornegger J, Wells WM, Frangi AF (2015) U-net: convolutional networks for biomedical image segmentation. In: Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 9351(Cvd), pp 12–20
23. Olsen GF, Brilliant SS, Primeaux D, Najarian K (2009) An image-processing enabled dental caries detection system. In: 2009 ICME international conference on complex medical engineering, CME 2009, pp 1–8
24. Frank C (2019) Everything you need to know about dental and oral health. https://www.healthline.com/health/dental-and-oral-health#TOC_TITLE_HDR_1. Accessed 20 Mar 2021
25. Pawar MM, Talbar SN (2018) Local entropy maximization based image fusion for contrast enhancement of mammogram. *J King Saud Univ Comput Inf Sci*
26. Peres MA et al (2019) Oral diseases: a global public health challenge. *The Lancet* 394(10194):249–260
27. Pitts NB, Ekstrand K (2013) International caries detection and assessment system (ICDAS) and its international caries classification and management system (ICCMS)—methods for staging of the caries process and enabling dentists to manage caries. *Commun Dent Oral Epidemiol* 41(1):41–52
28. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition 2016-Decem, pp 779–788

29. Sajeev S, Bajger M, Lee G (2015) Segmentation of breast masses in local dense background using adaptive clip limit-CLAHE. In: 2015 International conference on digital image computing: techniques and applications, DICTA 2015
30. Sathyaranayanan R, Carounnanidu U (2002) Classification and management of dental caries. new concepts. Indian J Dent Res Official Publ Indian Soc Dent Res 13(1):21–25
31. Şenel B et al (2010) Diagnostic accuracy of different imaging modalities in detection of proximal caries. Dentomaxillofacial Radiol 39(8):501–511
32. Senior A et al (2018) Digital intraoral imaging re-exposure rates of dental students. J Dent Educ 82(1):61–68
33. Shah N, Bansal N, Logani A (2014) Recent advances in imaging technologies in dentistry. World J Radiol 6(10):794–807
34. Prajapati SA, Nagaraj R, Mitra S (2017) Classification of dental diseases using CNN and transfer learning. In: 5th International symposium on computational and business intelligence
35. Singh P, Sehgal P (2021) G.V Black dental caries classification and preparation technique using optimal CNN-LSTM classifier. Multimedia Tools Appl 80(4):5255–5272
36. Sonavane A, Yadav R, Khamparia A (2021) Dental cavity classification of using convolutional neural network. IOP Conf Ser Mater Sci Eng 1022(1)
37. Srivastava MM, Kumar P, Pradhan L, Varadarajan S (2017) Detection of tooth caries in bitewing radiographs using deep learning. arXiv (Nips 2017)
38. Viswanath VH, Guachi-guachi L (2019) Breast cancer detection using image processing techniques and classification algorithms

Development of Data Set for Automatic News Telecast System for Deaf Using ISL Videos



Annu Rani, Vishal Goyal, and Lalit Goyal

Abstract Sign Deaf people use sign language to communicate with others. There are numerous languages because of countries and their cultural variations. So, every country carries its sign language to serve the impaired people. In this paper, we have outlined data collections and research challenges for our system. The data is collected related to “DD news with hearing-impaired people” from YouTube. To get the news data into script format, we used the Downsub site. This site supports script format into different languages script but according to our requirements, we downloaded news scripts into English. Then scripts are converted into unique unigram words by using the wordlist online tool. We have analyzed each word one by one and eliminated unwanted (articles, helping verb, inflections, etc.) words and finally, unique words are converted into Sign Language with the help of ISL experts.

Keywords Communication · Deaf people · News · Indian sign language · Script · Sign interpreters · Sign language · Translation system · TV channels · YouTube

1 Introduction

Communication is a mode by which people express their views, thoughts, feelings, ideas among themselves. Without communication, there is no life of human beings on earth. Normal people use spoken or written forms of communication to communicate and deaf people use sign language for communication purposes [1]. Sign Language is a mode of communication using visual gestures, postures, signs, and face expressions, as used by hearing-impaired people [2]. It has a rich vocabulary and each word has a specific sign. There are existed so many sign languages all over the world named American Sign Language (ASL), British Sign Language (BSL), Indian Sign Language (ISL), Japanese Sign language (JSL), and Korean Sign Language (KSL),

A. Rani (✉) · V. Goyal
Department of Computer Science, Punjabi University, Patiala, India

L. Goyal
Department of Computer Science, DAV College, Jalandhar, India

etc. [3]. Sign Languages change from area to area, state to state, country to country, and nation to nation due to variations of their geographical, historical, social, and cultural factors [4]. Every region or country has its gestures or signs to express the words [5]. For example, the normal people of the United States and the United Kingdom (UK) share the same spoken language but their sign languages as ASL, BSL respectively are quite different [6]. Even though. Some countries or state has more than two sign languages. Sign Language is the only way for deaf people to express their ideas, plans to others. The deaf community also wants to communicate with normal people but both communities do not understand each other languages. Normal people have no time and interest to learn sign language. So, language ignorance is a big barrier between deaf and normal people for communication [7]. To solve this problem, they needed a human interpreter who translates signs to equivalent language and vice versa to make them understand their communication [8]. It is not an easy task to find professional, and skilled human sign language interpreters for their life period, and also human interpreters are very expensive and not available all time. Also, listen to TV news, deaf people require sign language interpreters because mostly deaf people have only low to moderate reading skills to read news subtitles [9]. Deaf people feel more enjoyable watching TV shows in their country's sign language; much in the same way as normal people would like to listen to someone vocalize news instead of reading close captions on television. Even though the news programs are broadcasted on many television channels but the sign interpretation is not available on all the TV channels. This creates a barrier among deaf people to access information about the world. Even though the demand for having a news program in gesture language is established, but the difficulty lies with having a full-time (24*7) presence of a sign language interpreter in the broadcasting room [10]. To solve this problem, the main aim of our system is to convert the television news into Indian Sign Language (ISL) so that hearing-impaired people get updated news around the world.

The organized structure of the paper is as follows: In Sect. 2 presents the literature survey, Sect. 3 presents the research objectives, Sect. 4 presents the collection, implementation process of news data, and research challenges, Sect. 5 Research Challenges, Sect. 6 presents the conclusion.

2 Literature Survey

Filhol et al. [11] presented a system that translates French text to French Sign Language (LSF) by using a machine translation system. The system was made with the help of two components such as formalizing LSF production rules which involves analysis of linguistic functions from link LSF form features and triggering them with text processing which performed extraction tasks from the text by using grammar rules and broken down the extraction task in many subtasks. The authors presented the set of production rules and compared it with traditional methods.

Shelke et al. [12] presented a process of a system that automatically translates finger-spelled to speech and speech to finger-spelled. Spoken and written words were spelled by using finger spelling. This system was performed as an interface between hearing-impaired and normal people. The translation system was helpful for deaf people to make communication, exchange ideas, etc. The proposed system was suffering three crucial challenges that were extracting features from speech, for the recognition; inadequate sound gallery and the independent voice recognition to enhance speaker dependency.

Mishra et al. [13] proposed a machine translation system that translates the English text to Indian sign language based on Indian sign language rules. The machine identified the structure of English code and transferred it to the Indian sign language according to ISL grammar rules and sub-rules. Presented rules analyzed by linguistic experts. There were two techniques, used to evaluate the translated text. The first technique was manual and the second was automatically used by linguistic experts and machine translation systems respectively to validate it.

Oliveira et al. [14] presented the detail of the virtual sign platform which was used for the conversion from sign to text and text to sign. It was a bidirectional language conversion system that had been developed since 2015. The system had received positive compliments on various trials and pilot experiments. There were some preplanned enhancements and future upcoming process that was getting developed by the six distinct European countries. With the help of the sign interpreter's partnership, future goals were detailed in this context.

Yang et al. [15] proposed a system named Structured Feature Network (SF-Net) that was used to eliminate the challenges, faced by a continuous Sign Language recognition system. The efficiency of the proposed network was proved by its features which were used in the process of different levels like frame, gloss, and sentence to represent the sign gestures and motions into information. The two different trained databases called as RWTH-PHOENIX-Weather-2014 dataset and the Chinese Sign Language (CSL) dataset had been made ready by them for the testing process without the concern of other existed systems or prior training. After testing the outcomes of the system were shown clearly outstanding satisfaction in form of accuracy and versatility.

Oh et al. [16] presented the framework of text to animation conversion system for the presentation of Korean sign language to broadcast the weather forecast report. They had considered the weather reports scripts for the last three years to get the 500 words and 2700 motions to stream on the weather forecast. They analyzed each single captured motion as well as the frequency of repetitive words in the news script. Some of motions were already existed and belonged to daily life. They were used the Korean WordNet and KorLex dictionary to get the replacement of absent words by using their synonyms. As a result, these were proved as helpful to enhance the performance of translation systems and provided their services via different resources like the internet, PC, and mobile to view the news report.

Hayashi et al. [17] The author proposed a television system that delivered TV programs by Computer Graphics (CG) animation using TVML (TV program Making Language) technology. By using TVML, they had created a Text Generated-TV

system on the internet where TV scripts were written in the form of TVML, sent on the remote side. Then the viewer could access the server system to download the written data. After downloading the written script put it into specific software on the user side to create the CG animations and deliver them to the users. To estimate the performance of the system, they had conducted a web survey and inspired the testers to write comments on the performance of the system. According to the survey, most of the viewers thought that the news program presented by the CG animation was made well and satisfactory. But the system was suffering from some drawbacks like the CG animated character was not looking like a news anchor and was unable to show non-manual expressions.

3 Research Objectives

- To study and understand Indian Sign Language (ISL) and its sentence formation, grammar rules.
- To formulate rules for reordering words in English sentences other than simple required for its conversion into ISL.
- To develop a database of news telecasted for deaf people through several TV channels.
- To develop all-inclusive gazetteer list of words for various categories like Named Entities, titles, designations, general words, legal terms, academic words, etc. generally needed for news telecast and then develop equivalent animated ISL Videos for these words.

4 Data Set Collections

- We needed a full list of news that is used by the news channel. So First of all, we have collected the news in text form by visiting different sites. To get more familiar with the proposed proposal, I have visited the deaf school in Patiala.
- After getting the full list of news, distinct words are extracted from the list and a list of all distinct words of news was constructed. All the distinct words are then translated into Indian Sign Language. All the words are translated by Indian Sign Language Teacher and video footage of each word is collected one by one.
- Total 13,196 ISL videos of distinct words are recorded with the help of an Indian Sign Language Interpreter.
- After the preparation of the list of different words, some of them are coded into HamNoSys (HamNoSys) code one by one. Then HamNoSys code to SiGML (Signing Gesture Mark-up Language) is created of each word and all non-manual components (eyes gaze, eyes blinking, body movements, head movements, etc.) are added into each sign. Each word synonym, inflections, and part of speech are

also written in excel sheet. We are showing some collected data in form of tables below. Table 1 represents types of words with manual and non-manual code also (Table 2).

4.1 Implementation

- **YouTube:** It is a source site to get a drop-down list of news for hearing-impaired people from various channels.
- **News:** This module is used to select particular news from the drop-down list.
- **DownSub:** This is a very useful online website to get downloaded YouTube news into script form. It supports different languages. But we downloaded the news script in English according to our requirements.
- **Wordlist:** It is an online website to get a unique unigram words list from downloaded news scripts.
- **Analysis:** In this module, we manually analyzed and remove unwanted words such as identifies, inflections, linking words, etc.
- **ISL Videos:** All the final unique words are converted into ISL videos (Fig. 1).

4.2 Research Challenges

- Creation of English word to ISL sign using HamNoSys notation for English Words is a very time consuming process.
- There are no standardized grammar rules available for ISL.
- To collect synonyms and inflections of every word is also very time consuming task.
- Unavailability of standard sign tools, study materials which give any awareness.

5 Comparison Analysis

The comparison review presents the analyzed work in the tabular form (Table 3).

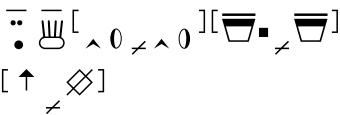
6 Conclusion

In this paper, we introduced a dataset about DD news for hearing-impaired people in the form of ISL videos. Out of 13,196 ISL videos, we have coded 3000 ISL Videos into Synthetic animations by using the HamNoSys tool. In the previous research, some

Table 1 Shows words with type, synonyms, inflections, manual and non-manual HamNoSys Code

Word	Sleep	Understand	Light	Fool
Type	Verb	Verb	Noun	Noun
Synonym-1	Doze	Recognize	Brightness	Half-wit
Synonym-2	Snooze	Know	Luminescence	Idiot
Synonym-3	Drowse	Acknowledge	Luminosity	Block head
Inflection-1	Sleeps	Understands	Lights	Fools
Inflection-2	Sleeping	Understood	-	-
Inflection-3	Slept	Understanding	-	-
Manual HamNoSys	-Ø₂λ₁ø₂x₂λ₁ø₁χ₂ø₂	ø₁ø₂x₂ø₁ø₂χ₂ø₂	ø₁ø₂χ₂ø₁ø₂χ₂ø₂	WB
Non-manual HamNoSys	TRTRTRCB	CB	SRRBBB	hmm_mouthpicture tag="life"/>
SiGML lipsing Code	<hmm_mouthpicture tag="slipping"/>	<hmm_mouthpicture tag="understand"/>	<hmm_mouthpicture tag="life"/>	hmm_mouthpicture tag="fool"/>

Table 2 Some words with HamNoSys Code and their animated signs by Virtual Avatar

Above			
Below			
No			

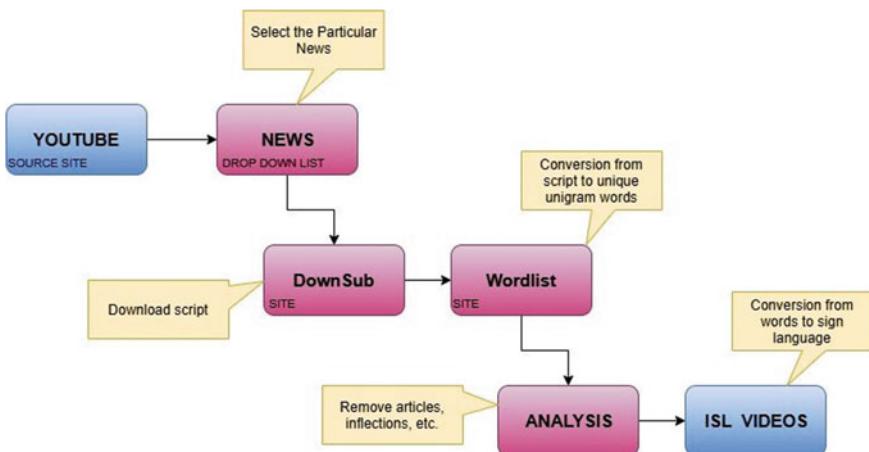
**Fig. 1** Process to implementation of data collections

Table 3 A comparison analysis of Text to Sign Language (SL) conversion systems

S. No.	Author's name	Language	Aim	Domain	Output	References
1	Sarkar et al. [18]	Bangla	Text to Sign Language	Education at Primary School	Pre-recorded Videos	https://doi.org/10.11109/INDCON.2009.5409449
2	Hiroyuki et al. [19]	Japanese	News to Japanese SigLanguage	Television News	CG Animations	https://doi.org/10.1145/190240179.190240
3	Cynthia J. Kellett Bidoli [20]	Italian	Italian to global sign language	Television news bulletins	-	https://core.ac.uk/download/pdf/41173608.pdf
4	Othman et al. [5]	American	English Text to ASL	English Text	3D avatar	10.1.1.402.5171
5	Mohamed Jenni et al. [21]	-	Text to sign language	Mobile phone communication	Sign Animation Avatar	https://doi.org/10.1145/2207016.2207049
6	Syed Faraz Ali et al. [22]	Indian	English text to sign language	Railway reservation counter	Virtual character	https://doi.org/10.47893/IJCSL.2014.1169
7	Jestin Joy et al. [23]	Malayalam	Malayalam text to sign language	Public places	-	https://arxiv.org/ftp/arxiv/papers/1412/1412.7415.pdf
8	Lalit Goyal et al. [2]	Indian	Different writing notations of SL	-	-	http://ijoes.vidyapublications.com/paper/Vol14/Vol14.pdf
9	Ghada Dahy Fathy et al. [24]	Arabic	alphabets into Arabic sign language	-	-	https://doi.org/10.15849/icit.2015.0024
10	Juhyun et al. [25]	Korean	Weather forecast news to Korean Sign Language	Weather forecast news	3D virtual Animation	https://doi.org/10.5594/JMI.2016.2632278

(continued)

Table 3 (continued)

S. No.	Author's name	Language	Aim	Domain	Output	References
11	Quach Luyt Da et al. [26]	Vietnamese	Text to Sign Language	TV News	3D Virtual Character	https://doi.org/10.1100/7978-3-030-05873-9
12	Taro Miyazaki et al. [27]	Japanese	Japanese to Sign Language	NHK TV Channel	Pre-recorded videos	https://www.aclweb.org/anthology/2020.signlang-1.23.pdf
13	Hao Zhou et al. [28]	Different Spoken Language	spoken language translated into source sign language	Corpus data set	Pre-recorded Videos	https://arxiv.org/pdf/2105.12397v1.pdf

researchers used pre-recorded videos. Due to the novelty, the Video representation introduces some obstacles too like more space for storage and a lot of time for execution. To overcome the above-mentioned obstacles, Synthetic animation is more reliable and still in progress.

References

1. Zwitserlood I, Verlinden M, Ros J, Van Der Schoot S, Netherlands T (2015) Synthetic signing for the deaf: eSIGN. January 2005
2. Goyal L (2015) 2. Sign Language hierarchy, vol 6913, pp 70–80
3. Dutta KK, Kumar SRK, Kumar AGS, Arokia SSB (2016) Double handed Indian Sign Language to speech and text. In: Proceedings 2015 3rd international conference image information processing. ICPIP 2015, pp 374–377. <https://doi.org/10.1109/ICPIP.2015.7414799>
4. Mishra J, Mishra G, Ravulakollu K, Rastogi R, Rafi KM (2014) Machine translation of Indian Signs for endocrinologist, vol 4(4):112–116
5. Othman A, Jemni M (2011) Statistical Sign Language machine translation: from English written text to American Sign Language Gloss, vol 8(5), pp 65–73 [Online]. Available <http://arxiv.org/abs/1112.0168>
6. Dangsaart S, Cercone N, Bridging the gap: Thai—Thai Sign Machine translation, pp 191–199
7. Mahesh M, Jayaprakash A, Geetha M (2017) Sign Language translator for mobile platforms. In: 2017 International Conference on Advance in Computing, Communications and Informatics, ICACCI 2017, vol 2017-Janua, pp 1176–1181. <https://doi.org/10.1109/ICACCI.2017.8126001>
8. Rokade YI, Jadav PM (2017) Indian Sign language recognition system. Int J Eng Technol 9(3S):189–196. <https://doi.org/10.21817/ijet/2017/v9i3/170903s030>
9. Bungeroth J, Stein D, Dreuw P, Zahedi M, Ney H (2006) A German sign language corpus of the domain weather report. In: Proceedings 5th international conference on language resources and evaluation. Lr. 2006, pp 2000–2003
10. Martin PJM, Belhe S, Mudliar S, Kulkarni M, Sahasrabudhe S (2013) An Indian Sign Language (ISL) corpus of the domain disaster message using Avatar. In: Proceedings of the third international symposium on Sign Language Translation and Avatar Technology, pp 1–4
11. Filhol M et al (2019) A rule triggering system for automatic text-to-sign translation. To cite this version: HAL Id: hal-01849003 A rule triggering system for automatic Text-to-Sign translation
12. Shelke VV, Khaire VV, Kadlag PE, Reddy KTV (2019) Communication aid for deaf and dumb people, pp 1930–1933
13. Mishra GS, Nand P, Pooja (2019) English text to Indian Sign Language machine translation: a rule based method. Int J Innov Technol Explor Eng 8(10 Special Issue):460–467. <https://doi.org/10.35940/ijitee.J1084.08810S19>
14. Oliveira T, Escudeiro P, Escudeiro N, Rocha E, Barbosa FM (2019) Automatic sign language translation to improve communication. IEEE global engineering education conference EDUCON. April-2019, no. June, pp 937–942. <https://doi.org/10.1109/EDUCON.2019.8725244>
15. Yang Z, Shi Z, Shen X, Tai YW (2019) SF-net: structured feature network for continuous sign language recognition. arXiv
16. Oh J et al (2016) An Avatar-based weather forecast Sign Language system for the hearing-impaired. To cite this version: HAL Id: hal-01391353 An Avatar-based weather forecast Sign Language System for the hearing-impaired
17. Hayashi M, Shishikui Y, Bachelder S, Nakajima M (2016) An attempt of mimicking TV news program with full 3DCG—aiming at the text-generated TV system. In: IEEE International symposium on broadband multimedia system broadcast. BMSB, vol 2016-July, pp 1–5. <https://doi.org/10.1109/BMSB.2016.7521902>

18. Sarkar B et al (2009) A translator for Bangla text to sign language. In: Proceedings of INDICON 2009—an IEEE India Council conference, pp 3–6. <https://doi.org/10.1109/INDCON.2009.5409449>
19. Kaneko H, Hamaguchi N, Doke M, Inoue S (2010) Sign language animation using TVML. In: Proceedings—VRCAI 2010, ACM SIGGRAPH conference on virtual-reality continuum and its application to industry, vol 1(212), pp 289–292. <https://doi.org/10.1145/1900179.1900240>
20. Kellett Bidoli CJ (2010) Interpreting from speech to sign: Italian television news reports. *Interpret News* 15:173–191
21. Boulares M, Jemni M (2012) Mobile sign language translation system for deaf community. In: W4A 2012—international cross-disciplinary conference on web accessibility, no. April. <https://doi.org/10.1145/2207016.2207049>
22. Ali SF, Mishra GS, Sahoo AK (2013) Domain bounded English To Indian Sign Language translation model, no. 1, pp 41–45
23. Joy J, Balakrishnan K (2014) A prototype Malayalam to Sign Language automatic translator, pp 2000–2002 [Online]. Available <http://arxiv.org/abs/1412.7415>
24. Fathy GD, Emery E, ElMahdy HN (2015) Supporting Arabic Sign Language recognition with facial expressions, no. April 2016, pp 164–170. <https://doi.org/10.15849/icit.2015.0024>
25. Oh J, Kim B, Kim M, Kang S, Kim I, Song Y, Avatar-based sign language interpretation for weather forecast and other TV programs, no 2
26. Da QL, Khang NHD, Ngon NC (2019) Converting the Vietnamese television news into 3D sign language animations for the deaf, vol 257, no. February. Springer International Publishing
27. Miyazaki T, Morita Y, Sano M (2020) Machine translation from spoken language to Sign language using pre-trained language model as encoder. In: Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages, no. May, pp 139–144 [Online]. Available <https://www.aclweb.org/anthology/2020.signlang-1.23.pdf>
28. Zhou H, Zhou W, Qi W, Pu J, Li H (2021) Improving Sign Language translation with monolingual data by sign back-translation [Online]. Available <http://arxiv.org/abs/2105.12397>

A Comprehensive Survey on Content-Based Image Retrieval Using Machine Learning



Milind V. Lande and Sonali Ridhorkar

Abstract In computer vision systems, however, retrieving a necessary image for a normal user is difficult. Over the last two decades, several studies have been conducted to improve the efficiency of automated image annotation, which has mainly concentrated on content-based image retrieval (CBIR) and attempts to identify specific images from a large dataset that are close to a query image. Various hybrid feature descriptors based on image cues such color, texture, shape, and so on that depict images were examined. For more than a decade, machine learning has been gaining traction as a viable alternative to hand-designed feature engineering. Feature extraction techniques and mathematical models are used to improve the performance and complexity of the image retrieval process. Furthermore, using the existing AlexNet convolutional neural network classifier, to increase the accuracy and performance of retrieval of Corel and Wang Datasets. Present state-of-the-art methods are described from different viewpoints for a deeper understanding of the development. Over the last period, the survey paper has given a comprehensive overview of hybrid feature extraction methods and machine learning-based enhancements for CBIR. In this paper, we hope to include a succinct overview of recent developments in CBIR and hybrid feature extraction methods, as well as machine learning-based enhancements for content-based image retrieval.

Keywords CBIR · Machine learning · Convolutional neural networks · Classification · Feature extraction

1 Introduction

The options illustrations and equivalency measure have a significant impact on the search efficiency of content-based image retrieval. Since prehistoric times, humans

M. V. Lande · S. Ridhorkar

Department of Computer Science, G H Raisoni University Amravati, Nimbhora, India

S. Ridhorkar

e-mail: sonali.ridhorkar@raisoni.net

have used pictures to communicate. The last century have seen rapid developments in computer vision technology, network infrastructure, data repository technology, smart phones, and cameras [1]. As a result of multimedia data being created, put on the Internet, and observed, there will be an explosion in the amount and scope of internet evidence being created, processed, shared, analyzed, and viewed [2] Looking for images that represent objects or situations, creating an unique style, or simply looking for images of the same theme or shape are all part of the process of assembling a picture set. Getting the correct image among a wide and diverse series becomes more difficult. Image retrieval issues are now more commonly accepted, and the pursuit for a solution is becoming a widely researched area. Using one or more keywords to describe the image is the traditional method of annotating an image with text images. The image may not have an efficient and accurate overview [3]. In recent years, CBIR has surpassed text retrieval in popularity. Accessing visual data by content retrieval is one of the most powerful and efficient methods of doing so. By using labeled text, the approach described above use image content such as shape, color, and texture.

With as little human interference as possible, an image retrieval system must be able to search and sort similar images from the database. According to the study, the visual features chosen for any device are determined by the needs of the end user. Discriminative feature representation is another key aspect for any image retrieval [4, 5]. And large processing cost is required to obtain better results in terms of representation fusion of low-level image elements, making the function more robust and unique [6].

The incorrect selection of features, on the other hand, can degrade the output of an image retrieval system [7]. Image feature vector can be feed across machine learning techniques via training and test models, improving CBIR performance [8, 9]. To apply a machine learning algorithm, you'll want a training–testing method whether in supervised or unsupervised. Recent image retrieval trends appear to be focusing on deep neural networks, which can produce better performance at a high speed [10, 11]. We hope to provide a succinct description of recent research developments in CBIR and feature representation that are challenging in this paper. The following are the primary goals of this pilot study: (1) How can low-level visual characteristics be used to improve CBIR performance? (2) How can you solve the rotation-invariant features problem? (3) How do machine learning-based methods help CBIR work better?

We performed a systematic analysis for the above-mentioned objectives in this review. Current developments are carefully reviewed by highlighting key contributions, with an impact on CBIR and feature extraction. The following is the review paper structure; Sect. 2 discusses the CBIR Framework. Sections 3 discusses the machine learning in the image processing context; Sect. 4, the different image retrieval techniques focused on feature fusion are investigated in the literature. Sections 5 describes the performance evaluation criteria as well as comparison measures used to test retrieval methods. Section 6 includes a conclusion unveiling suggestions in image retrieval methods based on image content.

2 CBIR Framework

CBIR systems are image search and retrieval systems that use computers to search for and retrieve image based on their visual information. The mandatory image features in the Jadhav and Ahmed Paper can be extracted and used as an index or search basis based on these visual contents [12]. These systems are a highly appealing and rapidly growing field of study for the effective development of image analysis methods. The application of machine learning to the challenge of image retrieval is known as content-based image retrieval (CBIR), query by image content (QBIC), and content-based visual information retrieval (CBVIR) (CBVIR). In term “content-based” refers to the examination of an image’s actual contents for the purposes of searching. Shape, color, texture, or other information that can be derived from the image itself will be related to as material. CBIR makes use of a variety of various image processing methods.

Figure 1 shows that a general framework of a CBIR system proposed by Hirwane et al. is split into offline and online stages. The system extracts visual attributes from each image in the database during the offline stage and stores them in a feature database [13], which is a separate database within the system. The rate of increase of each image are much smaller in size than just the data sets, and it can be considered as an abstraction in addition to feature data (also referred to as image signature) (compact form) of the images in the image database.

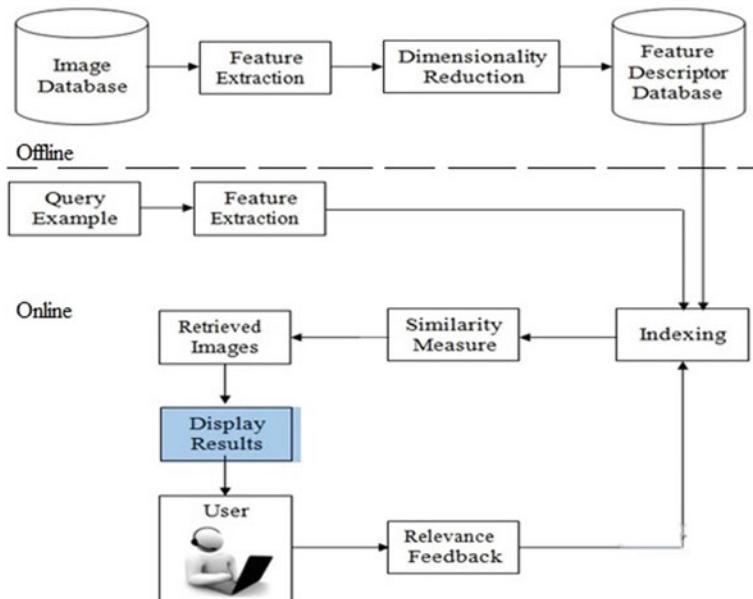


Fig. 1 Content-based image retrieval system: a general framework

2.1 Low-Level Features Extraction

Several low-level image descriptors for image representation and indexing have been reported in the literature in recent years. There are two types of image features which can be extracted: global and local. Global image characteristics such as color, texture, and form define the entire image and provide representative information derived from pixel analysis [14]. Local image features, such as edges, are typically obtained during feature extraction and are used to define distinct in the image. This section will show how important global and local image features are in CBIR systems, as well as the increasing interest in new algorithms, particularly several based on specific descriptors.

2.1.1 Global Image Features

Color, texture, and spatial position are most frequently extracted feature sets in image retrieval systems. The following features are presented and discussed in this section.

1. Color: Due to its close association with image sources, foregrounds, and backgrounds, it is one of the most robust vision features. Color histograms, moments, correlogram [15], and color co-occurrence matrix [16] are the most commonly used color representations. Color spaces are described as linear (e.g., RGB, XYZ, CMY, YIQ, and YUV) or nonlinear (e.g., L*a*b, HSV, Nrgb, Nxzy, and L*u*v). One of the MPEG-7 color descriptors, dominant color descriptor (DCD) [17, 18], It has been widely used in image retrieval applications to reflect the color knowledge of the target image with a small number of representative colors. DCD defines the representative color features and distributions in the image or regions of interest using an easy, accurate, and convenient format. Another method suggested by Hong et al. [19] MPEG-7 DCD is built on a fixed number.
2. Texture: Texture is characterized as regular patterns of pixels across a feature space in the field of computational linguistics. Regularity, directionality, smoothness, and coarseness are all texture properties are human eye perceives. The key disadvantages of texture-based image retrieval systems are their computational complexity and retrieval accuracy [20].
3. Shape: An image's shape produces domain knowledge and can be classified as border or region-based. The region method extracts features from the entire region, while the border method extracts features from the area's outer part.
4. Spatial information: In their derived representation, most common low-level features, such as histograms and shape points, ignore spatial information. As a consequence, explicit representations are inadequate to convey the perceptual quality of images in pictorial form ROIs, and statistic representations are relatively new concepts. But they contain a wealth of spatial information that is critical in region-based image retrieval (Table 1).

Table 1 Global and local image features' key characteristics

Features	Important characteristics	Limitations	Illustration	Semantic contributions
Color [16]	Regardless of its image's quality (e.g., size)	There is a scarcity of spectral data	<ul style="list-style-type: none"> - Histograms, HSV - Moments [21] - DCD and Hu moment 	It can be used to identify and retrieve high-quality images for image labeling and retrieval
Texture [22]	Entropy, linearity, sharpness, and uniformity are all discussed in detail	<ul style="list-style-type: none"> - Noise intolerances - Computer power 	Gabor filters <ul style="list-style-type: none"> - GLCM, MRF [23] - DCT,DWT - Tamura features 	<ul style="list-style-type: none"> - Describes an object's intrinsic surface properties and its relationship to its surroundings
Shape [24]	Image objects are represented in binary form	Translationally sensitive, rotation and scaling inconsistencies and stability	Fourier descriptors <ul style="list-style-type: none"> - Mass, centroid and dispersion 	Contains semantic data based on boundaries and regions
SIFT [25]	Scale and rotation are unaffected	They must be represented with texture analysis in specified vectors	PCA and SIFT [26], FLDA	Simply saves highlights that are relatively stable throughout time
SURF	<ul style="list-style-type: none"> - Uses integral images to reduce computation costs 	<ul style="list-style-type: none"> - Rotational invariance performance was poor 	SURF in a hurry	Identical to SIFT
Local patterns	<ul style="list-style-type: none"> - Vulnerability to nonlinear low resolution changes 	Noise tolerance in image regions with a near-uniform pattern	LTP [27], LBP, LTrPs [28], RMLBP	Extracts discriminative texture features that are uniform and non-uniform
HOG	<ul style="list-style-type: none"> - There is no need for precise edge positions - Normalization of local contrast 	Object detection with multiple bounding boxes	CHoG	Describes the appearance and shape of a local entity

2.2 Decreasing Dimensionality and Indexing

Deny the reality that many dimensionality reduction algorithm have been developed, and the curse of dimensionality remains an issue in CBIR. More work on indexing algorithms and frameworks is needed to develop more discriminating image descriptors which can effectively link low-level features with additional semantic data. When images are represented in higher semantics, therefore, high-dimensional image

features with sparse data distribution are usually generated. The retrieval efficiency of CBIR systems will undoubtedly suffer as a result of this.

2.2.1 Principal Component Analysis

It is a popular and effective way to reduce dimensionality. PCA is a linear transformation approach that uses an orthogonal matrix to project input vectors into new ones [26] (i.e., the sample covariance matrix's eigenvector). Just considering the first few eigenvectors that are sorted in descending order of Eigen values on Gaussian characteristics reduces the number of principal components.

2.3 *Machine Learning in the Image Processing Context*

Machine learning (ML) would be a type of data analysis in which the development of mathematical tools is optimized for a wide variety of data types, including images [29]. Machine learning, according to Mitchell [22], is the ability to enhance efficiency in completing a task by experience can learn from data, identify patterns, and make decisions with little human intervention.

When image processing issues occur, it is important to reduce the number of data entries in order to use certain predictive models. An image could be divided into millions of pixels for tasks such as classification. Data entry would make processing extremely difficult in this case. The picture is then reduced to a smaller collection of features to make it simpler to work with. This operation reduces raw input data to a reduced form and selects and tests representative properties [30]. Furthermore, such a set represents the pertinent piece of data needed to complete a desired mission. Color, texture, form, or a simple portion of an image may all be used to represent it [31].

By analyzing the underlying structure of items and attributes, the CBIR algorithm has recently been shifted to perform correlation retrieval on image data. Support vector machines (SVM) and other supervised algorithm are commonly used to classify data.

As previously stated, current retrieval techniques attempt to represent images using components specific to a location (e.g., SIFT & LTP). Legislate review, which categorizes image descriptors into predefined semantic classes by determining the best dividing path between conceptual communities of descriptors, maximizing similarity within clusters while minimizing similarity within clusters, was typically followed by it. We will take a look at some of the most interesting new ideas, including machine learning and image labeling is done automatically. This will greatly improve retrieval accuracy and performance.

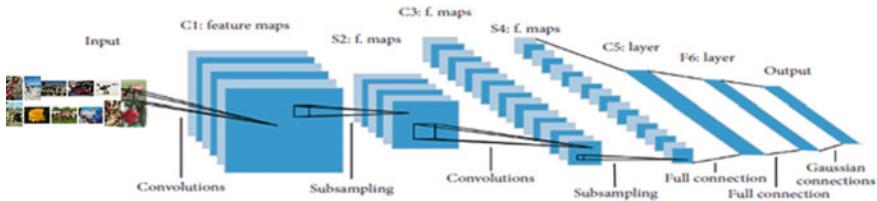


Fig. 2 Basic structure of convolution neural network

2.3.1 Supervised Learning with Classification

1. SVM classifier [32]: It has a long history of use in image classification and pattern recognition as a supervised classifier. The SVM decides the image category a new data set belongs to and represents a significant distance between the image classes in order to maximize the distance between them. The SVM maximizes the distance between the hyperplane and the nearest distance measure of each class by specifying the category (Y) to which the pixel (x) belongs, where the class is either +1 or -1. Assist vectors are training samples that are similar to the hyperplane.
2. Bayesian classifiers: It is been used to solve a wide range of machine vision problems, including those in the CBIR area. The basic concept behind image classification is that a collection of images is divided into groups, with each image belonging to one. Random variable samples with a priori membership variables are used to display class items. When a class is chosen, the 0 & 1 loss function is used to calculate the loss. For image retrieval issues, such decision needs to be made related to image characteristics instead of local texture values. As a result, instead of raw images, features have independent class-conditional densities, and the supervised learning can be carried out on the basis [33]: “provided evaluation metrics, assign the image into most of the category [34] (Fig. 2).

3 Literature Review

Shakaramia et al. [26]. Machine learning was used in a study to construct an efficient image description method. This approach was created using an enhanced AlexNet CNN, Histogram of Oriented Gradients (HOG), and Local Binary Pattern (LBP) descriptors. To minimize the number of dimensions, the principle component analysis algorithm was used.

Shikha and Gitanjali [9] proposed a unique hybrid was presented, and gray-level co-occurrence matrix (GLCM), color moment, and various area assets techniques are used to extract unique characteristics of an image, such as shape, color, and structure.

Another HFV (hybrid feature matrix or vector) is generated by combining relevant features from three different visual attributes.

Liu and Chen [22]. A systematic analysis of texture representations was provided using Bag of Words and convolutional neural networks with impressive results. Given this time of remarkable change, the purpose of this paper would provide a comprehensive overview on texture representation progress over the last two decades.

Arunkumar and Ranjith Ram [31] describing the advancement of retrieval techniques, with a focus upon its growth, benefits, drawbacks, and challenges of image retrieval. It focuses on a review of existing strategies as well as the issues that go along with them. The rapid expansion of image data necessitates the advancement of science, which contributes to the creation of effective and precise image retrieval.

Ghrabat et al. [30] conducted a study on features are optimized using a modified genetic algorithm, and new SVM-based convolutional neural network is used to classify them (NSVMBCNN). The result becomes excellent then graded on sensitivity, specificity, precision, recall, and retrieval. In experiments using four separate datasets, scene classification was outperformed by the expected extracting features and novel optimization classifier optimization technique.

Wei et al. [35] the proposed network has two streams, each of which performs two tasks at once. The main focus is on extracting discriminative visual features associated through properties which are semantic. Similarly, the auxiliary source aids the main stream by rerouting extraction of features to the most relevant image material that an individual is likely to find. Image similarity can be computed in the same way that a person does it by allocating prominent material and suppressing irrelevant regions through integrating these two streams into a main and auxiliary CNN (MAC).

Dubey [11, 36] a systematic review of deep learning-based advances in content-based image retrieval in the past ten years present state-of-the-art techniques are listed from different viewpoints to better understand of how it proceeds. Various types supervision, networks, descriptor types, and retrieval types are included in the taxonomy used in this analysis. A detailed summary of the above-mentioned machine-learning-based features for CBIR is represented in Table 2.

4 Performance Evaluation Criteria and Similarity Measurement

CBIR has a number of performance assessment requirements, all of which are handled according to a set of guidelines. It is worth noting that there is not a single standard rule or criteria for evaluating CBIR performance. Here is a list of some of the most common interventions that have been published in the literature. The following performance assessment metrics are widely used.

Table 2 Performance of various feature fusion-based image retrieval techniques for CBIR is summarized here

Author	Features	Feature extraction methods	Dataset	Similarity measure technique	Interpretation, accuracy and performance	Future work and limitation
Ashkan and Hadi [26]	Color, texture, and Shape	Improved AlexNet CNN, HOG and LBP descriptors	Corel-1K, OT and FP	Mean average precision	Accuracy 95.80 threefold cross validation Dimension 1 X 128	Use EfficientNet and mobileNetV3 instead of Alex Net CNN
Xie and Huang [37, 38]	Color, texture, and edges	Combining region and orientation correlation descriptors	Corel-1K, 5K, 10K	Euclidean distance, χ^2 -statistics	CROCD is 7.807 percent, and precision is improved by 6.65%, when the two become merged	For retrieval, combine color, texture, and shape features
Liu [39]	Color, texture, and spatial	Color difference histograms (CDHs)	Corel-5K, 10K	Canberra distance	Precision (%) 57.23 Recall (%) 6.87 Precision (%) 45.24 Recall (%) 5.43	The local and high-level features cannot be extracted
Xie and Long [37]	Color	DCD and Hu moments	Corel-1K, 5K, 10K	Euclidean distance	Corel-5k and 10k, the precision of DCD-HM is 4.72 and 4.25% higher than the DCD	For use in skin and face detection
Al-Mohamade et al. [40]	Color, texture, and concatenate features	Optimal feature relevance weights, weight-learner	Corel database	(ANMRR) score	Proposed approach based on Color Histogram 0.301, Texture Histogram 0.3959 and concatenated feature 0.2906	The observed weight would then improve the next query

(continued)

Table 2 (continued)

Author	Features	Feature extraction methods	Dataset	Similarity measure technique	Interpretation, accuracy and performance	Future work and limitation
Win [41]	Hybrid features	Hybrid features (GIST + HOG + BoVW + MobileNet + DenseNet) Optimized SVM PSO	Shenzhen dataset	Bayesian algorithm	Using the hybrid feature set, the SVM classifier achieved 92.7% accuracy and 99.5% AUC for the MC and 95.5% accuracy and 99.5%	Develop even more robust TB classifier
Singh [35]	Color, texture, Shape	Bi-layer CBIR, HSV, Zernike Moments	Corel database	Euclidean and Cosine	The suggested system is more precise and quicker	Extended with CNN-based image features
Wang and Fan [1]	Color	SVM and CNN algorithms	Corel-5K	–	The accuracy of SVM is 0.86 and the accuracy of CNN is 0.83	Data on a broad scale and identification accuracy
Varish [42]	Color and texture, shape	HSV, GLCM and HOG descriptors	Corel-1K GHIM-1K	F-score	In subsequent phases, the search space is limited. Best result 98.00% precision	Better accuracy as compared to the other CBIR Scheme
Khalid [43]	Color (DWT)	DBSCAN, SVM, KNN, and decision tree	Corel-1K, 5K, 10K	Average Precision (AP)	average accuracy on Corel-1K = 98.3%, Corel-5K = 98.8% Corel-10K = 98.8%	Discerning research with larger datasets and better feature vector formation
Dhingra [2]	Color, shape and texture	LBP, color moment	Corel-1K	Average precision	The retrieval time and AP is 97% improved	Improve precision and retrieval time

(continued)

Table 2 (continued)

Author	Features	Feature extraction methods	Dataset	Similarity measure technique	Interpretation, accuracy and performance	Future work and limitation
Kumar [24]	Shape	Canny edge detection algorithm and SVM	Wang	Euclidean	Sensitivity and specificity are 99.20 and 96.48%	Used in diverse realistic environmental conditions like illumination,

- **Precision and Recall.** Two widely used parameters for assessing CBIR study outcomes are precision (P) and recall (R). The following formula is used to calculate the value of the amount of valid images inside the first k results to the total number of images retrieved: The ratio of accurate images retrieved to the total number of images retrieved (NTR) determines the precision.

$$R = \frac{tp}{N_{TR}} = \frac{tp}{tp + fp'} \quad (1)$$

while tp stands for the actual images collected and fp stands for false—positive or images labeled as image content.

- **Recall.** Recall is defined as the proportion of relevant images retrieved to the total number of relevant dataset.

$$R = \frac{tp}{N_{RI}} = \frac{tp}{tp + fp'} \quad (2)$$

- NRI represents the number of related image database, and tp stands for the number of specific images collected. tp + fn yields the false negative, or pictures that originally looked similar to the appropriate class or were wrongly classified to another.
- **F-Measure** is an acronym for “F-Measurement. It is the harmonic mean of P and R, the higher the F-measure value, the more predictable its outcome:

$$F = 2 \frac{P \cdot R}{P + R'} \quad (3)$$

- Precision and recall are denoted by the letters P and R, respectively.
- **Average Precision (Mean):** The MAP is calculated for such a collection of queries. S is the average of each question’s AP values, which is determined as follows:

$$MAP = \frac{\sum_{q=1}^s AP(q)}{S} \quad (4)$$

5 Similarity Measurements

The main goal of a CBIR is to analyze and retrieve images from a database that are identical to the query image found by a user in a professional manner. Finding strong image similarity measures based on a feature set is a difficult job [31]. The method of determining Similarity calculation is the process of calculating the stylistic similarities between database images and query images based on their features. To measure similarity, distance equations like Euclidean distance, City block metric,

Minkowski distance, Mahalanobis distance, and Quadratic Form distance [7] can be used.

6 Conclusion and Future Directions

This paper discusses CNN's utility in image retrieval applications, given its widespread use in image classification and representation. We looked at numerous techniques for improving retrieval performance utilizing relevance feedback, as well as the effectiveness of features generated by CNNs versus hand-crafted features. By combining different image patterns, which reflect the image in the type of maps and boost performance, the semantic gap can be narrowed. Merging locally and globally characteristics is one of the potential research directions in this area. Conventional machine learning methods delivered excellent results in a number of domains in previous CBIR and image representation study. Recent CBIR research use of deep neural networks, which outperformed hybrid features under the condition of network fine-tuning and showed good corel dataset performance. As a result, testing the efficiency of a deep network in unsupervised learning mode on an unfavorable variance dataset is among the potential future work in this area.

References

1. Wang P, Fan E (2020) Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recogn Lett Elsevier*
2. Dhingra S et al (2020) A novel & efficient fusion based image retrieval model for speedy image recovery. *EAI Endorsed Trans Scalable Inf Syst* 05 2020–10 2020:7(27)
3. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. *IEEE Trans Pattern Anal Mach Intell* 22(12):1349–1380
4. Verma B, Kulkarni S (2006) Neural networks for content based image retrieval. *Semantic-based visual information retrieval*, pp 252–272
5. Li J, Allinson NM (2013) Relevance feedback in content-based image retrieval: a survey. In: *Handbook on neural information processing*. Springer Berlin Heidelberg, pp 433–469
6. Wan J, Wang D, Hoi SCH, Wu P, Zhu J, Zhang Y, Li J (2014) Deep learning for content-based image retrieval: a comprehensive study. In: *Proceedings of the ACM international conference on multimedia*, pp 157–166. ACM
7. Jørgensen C (2003) *Image retrieval: theory and research*. Scarecrow Press
8. Shrestha A, Mahmood A (2019) Review of deep learning algorithms and architectures. *Dig Object Identif. IEEE Access*. 1 May 2019
9. Shikha B, Gitanjali P, Pawan Kumar D (2020) An extreme learning machine-relevance feedback framework for enhancing the accuracy of a hybrid image retrieval system. *Int J Interact Multimedia Artif Intell* 6(2)
10. Qureshi AS et al (2020) A survey of the recent architectures of deep convolutional neural networks. *Artif Intell Rev. © Springer Nature B.V*
11. Shriwas MK, Raut VR (2015) Content based image retrieval: a past, present and new feature descriptor. In: *2015 International conference on circuits, 2015 power and computing technologies*, pp 1–7

12. Jadhav SH, Ahmed SA (2012) Content based image retrieval system with hybrid feature set and recently retrieved image library. *Int J Comput Appl* 59(5):46–55
13. Hirwane R (2012) Fundamental of content based image retrieval. *IJCST Int J Comput Sci Technol* 3(1):114–116
14. Mistry Y (2017) CBIR using hybrid features and various distance metric. *Sci Direct*, 2314–7172. Elsevier
15. Jing H, Kumar SR, Mitra M, Zhu WJ, Zabih R (1997) Image indexing using color correlograms. In: IEEE computer society conference on computer vision and pattern recognition, Proceedings 1997, pp 762–768
16. Xiaoyin D (2010) Image retrieval using color moment invariant. In: The seventh international conference on information technology: new generations (ITNG), Las Vegas, NV, 12–14, pp 200–203
17. Manjunath BS, Salembier P, Sikora T (2002) Introduction to MPEG-7: multimedia content description interface. Wiley, Chichester
18. Qiu GP (2003) Color image indexing using BTC. *IEEE Trans Image Process* 12(1):93–101
19. Shao H, Wu Y, Cui W, Zhang J (2008) Image retrieval based on MPEG-7 dominant color descriptor. In: The 9th international conference for young computer scientists, ICYCS 2008, pp 753–757
20. Nair AS, Jacob R (2017) A survey on feature descriptors for texture image classification. *IRJET* 4
21. Gonzalez RC, Woods RE (2002) Digital image processing
22. Lu L, Chen J (2018) From bow to CNN: two decades of texture representation for texture classification. *Int J CV* 6 Oct 2018
23. Wei Z, Liu G-H (2020) Image retrieval using the intensity variation descriptor. *Math Probl Eng* 2020:12, Article ID 6283987
24. Shijin Kumar PS (2020) Key point oriented shape features and SVM classifier for content based image retrieval. *Mater Today Proc*, 2214–7853. Elsevier
25. Zhou W, Li H, Tian Q (2017) Recent advance in content-based image retrieval: a literature survey. Cornell University, Ithaca, NY, USA
26. Ashkan S, Tarrahb H (2020) An efficient image descriptor for image classification and CBIR. *Optik—Int J Light Electron Opt* 214:0030–4026. © 2020 Elsevier GmbH
27. Pradhan J, Kumar S, Pal AK, Banerji H (2018) A hierarchical CBIR framework using adaptive tetrolet transform and novel histograms from color and shape features. *Digit Sig Process Rev J* 82:258–281
28. Zhang B, Gao Y, Zhao S, Liu J (2010) Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Trans Image Process* 19(2):533–544
29. Ai L, Yu J, He Y, Guan T (2013) High-dimensional indexing technologies for large scale content-based image retrieval: a review. *J Zhejiang Univ Sci C* 14(7):505–520
30. Jalil M, Ghrabat J (2019) An effective image retrieval based on optimized genetic algorithm utilized a novel SVM-based convolutional neural network classifier. *Comput Inf Sci* 9:31
31. Arunkumar N, Ranjith Ram A (2020) CBIR systems: techniques and challenges. In: International conference on communication and signal processing, 28–30 July, 2020
32. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
33. Wei S, Liao L, Li J, Zheng Q, Yang F, Zhao Y (2019) Saliency inside: learning attentive CNNs for content-based image retrieval. *IEEE Trans Image Process* 28(9)
34. Kittler J, Roli F (2010) Multiple classifier systems for robust classifier design in adversarial environments. *Int J Mach Learn Cybern* 1(1–4):27–41
35. Singh S, Batra S (2020) An efficient bi-layer content based image retrieval system. *Multimedia Tools Appl* 79(25–26):17731
36. Dubey SR (2020) A decade survey of CBIR using deep learning. Computer Vision Group
37. Xie G, Guo B et al (2020) Combination of dominant color descriptor and hu moments in consistent zone for content based image retrieval. *Dig Object Identif. IEEE Access* 8
38. Fadaei S, Amirfattah R, Ahmadzadeh MR (2017) New content-based image retrieval system based on optimised integration of DCD, wavelet and curvelet features. *IET Image Process* 11(2):89–98

39. Liu G-H, Image retrieval using the intensity variation descriptor. Hindawi Math Probl Eng 2020:12, Article ID 6283987
40. Mohamade A et al (2020) Multiple query content-based image retrieval using relevance feature weight learning. J Imag 6(2)
41. Win KY et al (2020) Hybrid learning of hand-crafted and deep-activated features using particle swarm optimization and optimized support vector machine for tuberculosis screening. Appl Sci 10:5749
42. Varish N et al (2020) Image retrieval scheme using quantized bins of color image components and adaptive Tetrolet transform. Dig Object Identif IEEE Access 8
43. Khalid MJ et al (1886) Integration of discrete wavelet transform, DBSCAN, and classifiers for efficient content based image retrieval. Electronics 2020:9

Keyphrase Extraction from Twitter Data—A Supervised Deep Learning Approach



K. B. J. Lemuel and V. Subramaniyaswamy 

Abstract A successful keyphrase extractor extracts exclusive, relevant keyphrases that capture the summary about the topic of the data and help in a fast information process. There are various methods previously available for keyphrase extraction, especially from Twitter data. Despite their availability, performance enhancement of these methods is a challenging problem. Also extracting keyphrases from twitter can help in extracting an idea from tweets related to specific social issue, disaster, user reviews about a product which in turn help in various applications which use keyphrases in awareness making, law abiding and recommendation applications. Hence, focusing on enhancing the performance of the existing models and to extract relevant keyphrases, we propose a supervised keyphrase extraction model using Extractive BERT Summarizer (BERT SUM) with Bidirectional Long Short-Term Memory (Bi-LSTM) using Global vectors (GloVe) word embedding from the Twitter tweets. We implement the model on the real-time Twitter data of 6000 tweets. We use BERT SUM for summarizing the tweets and Bi-LSTM for classification and extraction of keywords. To evaluate the performance of the proposed architecture the performance metrics precision, recall and F1-score are used, which are accepted commonly by the previous keyphrase extraction works. The results from the experiment showed that the proposed architecture outperforms the already available state-of-the-art keyphrase extraction methods.

Keywords Twitter · Supervised Keyphrase extraction · Twitter analytics · Information retrieval · Deep learning

1 Introduction

Keyphrase Extraction (KPE) models do the task of extracting the important phrases related to the topic from the data, providing a descriptive idea and condensed summary. These extracted keyphrases also are used in the different recommendation

K. B. J. Lemuel · V. Subramaniyaswamy (✉)
School of Computing, SASTRA Deemed University, Thanjavur 613401, India

and indexing applications [1–4]. Keyphrases are an extension of keywords where keyphrases contain two or more words that are highly informative and descriptive than that of the keywords containing a single word. The evolution of the social media platform, World Wide Web and the Internet has made researchers interested in studying technologies related to content summarization and keyphrase extraction which is increasing day by day. Twitter is one of the online social media platforms which provide the people to communicate their opinions and emotions with the help of short messages [5] called “tweets”.

Twitter is used widely around the world that people around the globe are connected on the same platform and are free to post their views on a particular topic. Due to the ease of access and simple to use by all the people, Twitter has become one of the real-time news sources that can provide up-to-date information around the globe. As Twitter can be accessed anywhere at any time, this helps people to post their situation at times of natural calamities, disasters, global activities, etc. This can help in developing situational awareness and has helped the enforcements in the past in countering the disasters and also lawbreaking by just knowing information that is tweeted at disaster times or such bad incidents [2, 6]. Thus, extracting the keyphrases and getting the description of the tweet, filtering out the information from a large data of Twitter, in real-time, plays a vital role in this modern world. Considering the importance of the keyphrases, studies are being done to extract keyphrases automatically using supervised [7–9] as well as unsupervised methods [10, 11]. Earlier methods were focused on the extraction of keyphrases from a single document or an article and they use statistical and linguistical features to learn the importance or weight of the words or phrases which are present in the document. Moreover, in addition to the available methods, few of the researchers proposed some methods to counter the problem of keyphrase extraction from real-time tweets from Twitter [1, 12, 13]. Topic-based PageRank method was used to extract keyphrases which is sensitive to the context of the tweets tweeted [14–16].

All the previous works have followed the traditional technique of finding the candidate keyphrases first and then applying the statistical or linguistic approach on these candidate keyphrases building the training set to train the model. That is candidate keyphrases were first extracted from the whole data making the search for the exact relevant keyphrase to a small space. Then the candidate keyphrases were labelled according to their presence in the gold-standard keyphrase list. In supervised models for the classification of keyphrases and non-relevant keyphrases, the candidate keyphrases were labelled 0 or 1, in consultation with the gold-standard keyphrase list and they train a classifier model on top of the word embeddings [1, 17]. In unsupervised methods, clustering is done on the word embeddings where the clusters which were formed, determined the keyphrases of the tweets [18, 19].

In this study, we found that it was difficult to extract the keyphrases from single Twitter tweets to that of extracting the keyphrases from the documents as the tweets contain shorter text to than of documents containing a whole lot of words. Since the number of tweets was high some other method was to be found to make the search space of relevant candidate keyphrases to decrease. We also found that labelling every single tweet is not feasible, as manual labelling is time-consuming and consistency

cannot be maintained. Since it is a real-time data which is collected up-to-date, there is no specific training data, evaluation data, or gold standard keyword list available already with the collected data which were ought to be generated by us.

Thus, to counter the challenges, in this proposed work, we first collected Twitter tweets which consist of hashtag words in it. These hashtag words would be relevant to the tweet posted, so selecting the tweets which only contain hashtag words would eliminate the selection of the posts which are irrelevant to the subject. For example, for tweets under the topic “#covid19” would consists of at least two to three hashtag words related to the topic. As these hashtag words would be relevant to the topic, these words can be taken as ground truth of the model. Thus, gold-standard keyphrase list was generated using collected hashtag words.

The relevancy of the gold-standard keyphrase list was also enhanced by adding the frequently occurred words in the tweets. After the construction of gold-standard keyphrase list, we joined all the tweet sequences into one sequence and make BERT Summarizer summarize the single sequence of text. BERT SUM as a pre-trained model has pre-trained methods to tokenize and summarize the raw data given to it. The summary was then split into sentences.

Each sentence was tokenized such that each word in the sentence was labelled according to the occurrence of the words in the sentence consulting with the gold standard keyword list. This labelling technique is said to be sequence labelling [17]. This sequence labelling technique makes the effort of manual labelling and doubts in the consistency to null by making the labelling technique easier and automatic. In our case, three labels or class names were given based on the occurrence of words as keyphrases in the sentence and the presence of the word in the gold-standard keyphrase list.

The specific contributions are as follows;

1. We devise a Supervised model using Bidirectional Encoder Representations from Transformers Summarizer (BERT SUM) and Bidirectional Long Short-Term Memory (Bi-LSTM) to extract the keyphrases from the large Twitter data.
2. The whole raw Twitter data is first summarized using the extractive BERT summarizer.
3. To construct the train and test data, the summarized data is preprocessed and the candidate keyphrases are extracted and are labelled in consultation with the gold-standard keyword list.
4. The labelled data are fed into Bi-LSTM supervised model to train, test and evaluate the model.
5. Experimental results show that the proposed method could achieve better results than the previously available methods.

The remaining of the paper content is organized as given. The related work on the keyphrase extraction algorithms is discussed in Sect. 2. In Sect. 3, we discuss the extraction of candidate keyphrases, labelling the data and describe the implementation of the proposed work. In Sect. 4, we discuss the experimental evaluation, and in Sect. 5, we present the conclusion of the paper and the future work.

2 Related Works

The existing supervised methods consider the keyphrase extraction task as a binary classification task where n-gram keyphrases are extracted from scientific articles or documents, or any social media [20–22]. These known supervised methods first extract candidate keyphrases by n-gram tagging, POS tagging or sequence tagging or by both, mostly POS tagging, then label the data by constructing feature set and later the training set is fed into the model for classification, determining the candidate keyphrase to be a relevant keyphrase or not. Various supervised keyphrase extraction methods are proposed by Naïve Bayes [23, 24], neural networks [25], Support Vector Machine (SVM) [15]. But for real-time data of tweets labelling the data, is not an easy task since the length of some tweets is small. Previous works of extracting candidate keyphrases from the Twitter data have used hashtag words as the ground truth for labelling the dataset [1]. In an article [2], the collection of data was focused specifically to a particular disaster or natural calamity, where hashtags were used to annotate the data for ground-truth generation.

There are also many unsupervised methods proposed for the keyphrase extraction where the candidate keyphrases were extracted first and then these candidate keyphrases are ranked based on their co-occurrence between them and frequency of each keyphrase in the document [10, 11, 26–28]. However, because of the shorter length of the tweets in our Twitter data, the frequency of a keyphrase to reoccur was not more than once in a single tweet which was the challenge there where the keyphrases should at least occur more than once. However, the article [17], used sequence labelling using the context of the candidate words which we follow in the proposed model which makes the labelling task automatic and a lot easier. We also follow the previous models for consulting the gold-standard keyphrase as keeping the same as the ground truth of the model.

Previously proposed methods for extraction of keyphrases from Twitter data used various supervised models too. The Joint-layer Bi-LSTM model [2] was used to extract the keyphrases from disaster-related tweets from real-time disaster-related tweets collected from the Twitter. The Joint-layer Recurrent Neural Networks model [1] performed the keyphrase extraction task on a Twitter data which was previously collected and annotated. The model performed well to that of the Automatic Keyphrase Extraction on Twitter (AKET) model [18] which also performed on the same annotated Twitter dataset. Brown Clustering was used in the AKET which is an unsupervised technique hence performance was low compared to the former. Another work showed the Convolution Neural Network (CNN) model with Crisis embedding and the Bi-LSTM model with GloVe embedding was compared on performing over the Twitter data where latter outperformed the former with the good scoring results [29].

We aim to overcome the challenge of labelling the data manually using sequence labelling and also use hashtags to generate the gold-standard keyword list. Candidate keyphrases are found by summarizing the tweets where each word after summarizing are candidate keywords. The Bi-LSTM model is used in our proposed work

as it shows excellent results compared to the other works. These followed methods maintain the consistency of the supervised methods over the unsupervised methods.

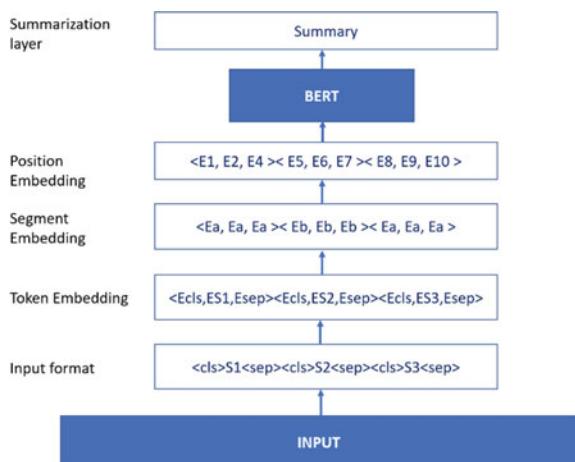
3 Methodology

3.1 BERT Summarizer

The Bidirectional Encoder Representations from Transformers (BERT) which was proposed by Google has been fine-tuned for the Natural Language Processing task of extractive summarization of text [30]. Extractive summarization is where the model summarizes long text into small sentences joined together a condensed paragraph giving a condensed information about the long text. For the long text of the data, the format of the input to the BERT SUM is adding [CLS] tag at the start of the sentence and [SEP] tag at the end of each sentence distinguishing one sentence from the other. Then these sentences are encoded to generate token embeddings, segment embeddings and position embeddings (Fig. 1).

Each sentence is assigned an embedding $E = \{E_a \text{ or } E_b\}$ depending on the position of the sentence whether odd or even. For example, if a multiple sentences $S = \{S_1, S_2, S_3\}$ then the segment embeddings are $E = \{E_a, E_b, E_a\}$. Thus, the sentences are encoded based on the segment and position embeddings. These encoded vectors are then fed into the BERT which generates sentence vectors. BERT SUM generates scores for each sentence such that scores show the importance of the sentence to the whole long text of data. For example, for the above sentence $S = \{S_1, S_2, S_3\}$ the scores generated might be, “y” array of scores for the sentences. For example, $y = [0.083, 0.056, 0.90]$. After which these sentence vectors are then fine-tuned by the summarization layers which are in-built module in the BERT SUM, are stacked above

Fig. 1 BERT summarizer architecture



the outputs to generate the summary. The summary is generated by summarization layers by rearranging the sentences of highest scores to give an overall condensed summary of the long text of data.

3.2 Bi-LSTM

One of the variants of Recurrent Neural Network (RNN) is Long Short-Term Memory (LSTM). They are used to catch long-distance dependencies within the texts. The LSTM remembers by the three gates which devise the section of information to remember and thus they can hold the contextual meaning of each word by the information surrounding it and store long dependencies between words (Fig. 2).

However, LSTM focus on one direction of the input text whereas the Bi-LSTM which is an extension of LSTM focuses on both the past and future directions of the text. Bi-LSTM outperforms LSTM by acquiring more contextual meaning about the text and hidden representations of the text. We use Keras library to build the Bi-LSTM. The word embeddings are given as input, where here in the proposed work we use GloVe embedding to make the dictionary of embedding vectors. The pre-trained GloVe embeddings are trained specially for Twitter data in 200 dimensions and 27 billion tweets. As our model is focused on the extraction of keyphrase from Twitter, we use this GloVe embedding pre-trained in Twitter. The summarized text was tokenized and every single word is assigned an integer. Thus, for a word an array of embedding vector was created.

For example, for a text “want”, the embedded vector will be [3.1705e–01, 5.1477e–01, –2.6564e–01, 2.4678e–01, –5.7847e–01, 1.6954e–02,

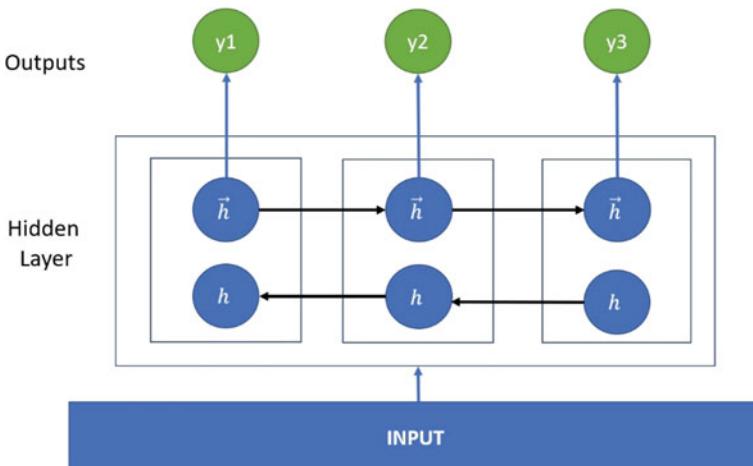


Fig. 2 Bi-LSTM architecture

$1.1577e+00, -3.2305e-03, 3.4475e-01, -1.3488e-01, 1.0422e-01,$
 $6.8276e-03...]$.

The training set was generated with embedded vectors corresponding to the words in the train data. The model was trained using the GloVe embedding vectors in the training set and keyphrases were extracted. The model was evaluated using performance metrics namely precision, recall and F1-score.

3.3 Candidate Key Extraction and Labelling

Here first we collected the data of tweets from the Twitter which contains hashtag words within it. We first remove the Web site addresses as a process of denoising the data. To shorten the search space of finding the candidate keyphrases, we combine all tweets in a single long sentence of text. This long sequence was then given as input to the BERT SUM, where the data was split into sentences, given token embeddings, segment embeddings and position embeddings. The scores of each sentence are generated and a relevant condensed summary was generated. The summary was then split into sentences. This reducing the search space for candidate keyphrase by BERT SUM is a unique way to approach the candidate keyphrase extraction problem.

Algorithm: 1 - Gold standard keyword list generation

Input: List of tweets (tw)

Output: Gold Standard Keyword List (g)

1. **for** do
 2. **for** tweet in tw **do**
 3. word \leftarrow Tokenize(tweet)
 4. **if** "#" in word
 5. append word to g
 6. **end if**
 7. **end for**
 8. **end for**
 9. **return** g
-

We then apply preprocessing techniques on these sentences like stopword removal, lower the text, replace numbers with equal text, punctuation removal and lemmatization. In preprocessing phase, we performed stopword removal where we remove the stop words mentioned in the standard English stopword list. We also performed Lemmatization where the words are properly checked to the vocabulary with analysis of morphology, focusing to remove the damaged ending texts and replacing it with the dictionary form of the same word, which is called a lemma. In punctuation removal we remove all punctuations by using a regular expression, we also include removal of emojis in the regular expression to get the text clean. We performed

lowering the text and replacing the number with equal text to not remove away any important keys from the training data. For example, for the number “19”, the equal text is “one nine” when the replacement phase is done on the data.

Then these sentences are tokenized and are then labelled using the sequence labelling technique. The hashtag words which were collected early with tweets are the source for the gold standard keyphrase list. As the hashtag word will be a word comprising of two words without space, we use Word Segment to split the word into two words. These words segmented will form the gold standard keyphrase list which acts as the ground truth for the model. Then in the labelling phase, each sentence was tokenized separately such that these tokens are labelled by three classes S-Key, I-Key, 0 key. For example, Let $A = \{\text{word1}, \text{word2}, \text{word3}, \dots\}$ be the input tokens for labelling. Let the classes be $C = \{S, I, 0\}$, then these labels are assigned consulted with the gold standard list, such that S be the start of the keyphrase, I be the inside in the keyphrase and 0 be assigned to keyphrases not in the gold standard list hence making it not a keyphrase. After the generation of the training set, it was fed to the model for classification.

Algorithm: 2 - Sequence Labelling

Input: Unlabelled list of sentences (s), Gold standard keyphrase list (g).

Output: Labelled list of labels ($Label$)

```

1. for do
2.    $A \leftarrow \text{Tokenize}(s)$ 
3.   for word in  $A$  do
4.     if word in  $g$  then
5.       if  $\text{Pos}(\text{word})$  is 0
6.          $Label \leftarrow S$ 
7.       Else  $\text{Pos}(\text{word})$  is  $I$ 
8.          $Label \leftarrow I$ 
9.       end if
10.    Else
11.       $Label \leftarrow 0$ 
12.    end if
13.   end for
14. end for
15. return  $Label$ 

```

3.4 Keyphrase Extraction

For the 6000 tweets collected under the hashtag topics, “#Covid-19”, “#ipl2021”, “#oxygen”, and “#TNelections” after preprocessing and summarization of BERT SUM, the training and evaluation data was constructed.

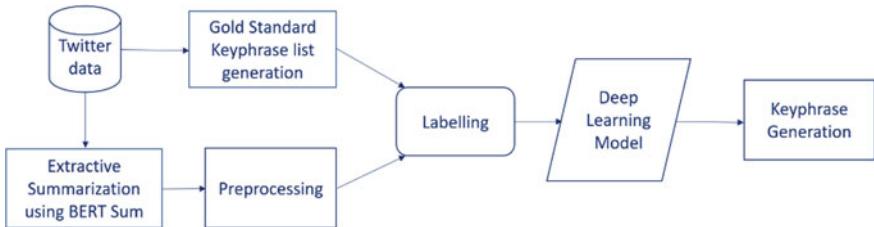


Fig. 3 Workflow diagram

The training set for the model was constructed by, for each token, in the sentence, we consult the gold standard keyphrase list and label the tokens with appropriate labels. Since we are assigning three classes the task has become multiclass classification. Thus, the training data was generated. The words in the training data for all four of the datasets collected were 4504, 6131, 5233 and 3289 respectively to the above-mentioned hashtag topics. The same way the evaluation data was 1333, 1742, 1451, 929 respectively.

We use Bi-LSTM for the classification task, where each word was given as input and to get classified by the model predicting the two words to be the keyphrase or not. Each word in the training set was embedded with a 200-dimensional GloVe vector pre-trained on a general Twitter corpus (Fig. 3).

GloVe embedding is a general pre-trained word embedding which had helped enhance NLP tasks. GloVe embedding is a 200-dimensional embedding trained on 27 billion words from Twitter tweets, which is publicly available. Because of using the GloVe embedding which is pre-trained concerning Twitter data, the performance of the model excels to that of the state-of-art models. We feed the embedded vectors of the training set to the Bi-LSTM model for the classification and extraction of keyphrases. The Bi-LSTM model had hidden units of 300 and a dropout of 0.5 to the first and second of the LSTMs. For training the model we used Adam optimizer with the learning rate of 0.0015 with batch size and epochs of 28 and 50 respectively. The first embedding has been fine-tuned and updated during the modification of the gradient using back-propagation. It was observed that for each epoch up to 50 epochs the accuracy increased and the model was learning better.

4 Experimental Result and Discussion

4.1 Dataset

In this paper, we implemented the model with the real-time data which are tweets from the Twitter. We collected top trending tweets without targeting any specific domain (e.g., disaster awareness tweets #australianfires), rather we collected the data generalized to any domain to make the model independent of the domain.

We collected 6000 tweets from four hashtags “#Covid-19”, “#ipl2021”, “#oxygen”, “#TNelections” using tweepy tool from the Twitter. We select the tweets which contain hashtags with it. We then removed the URLs from the tweets by using the tweet text preprocessor and clean the tweets using tweet preprocessor inbuilt with the tweepy tool. After collecting the tweets, the hashtag words were extracted from the tweets and combined with all the hashtag words as a gold standard keyphrase list. For the enhancement of the ground truth, we also added the words which are frequently occurring in the tweets. Thus, we have generated a gold-standard keyword list with hashtag words and the frequently occurring words. We extracted the hashtags and split the hashtag words as they contain multiple words when posted. For example, “#Covidsecondwave” was split into “Covid”, “second”, “wave”. After preprocessing of the summarized text is labelled with consulting the ground truth, the training data and evaluation data is created.

Algorithm: 3 - Data Construction

Require: Tweets with hashtags

Output: List of tweets (*tw*)

1. $tw \leftarrow \emptyset$
 2. **while** *i* in tweet **do**
 3. **if** *i* not contains latin letters **then**
 4. **continue**
 5. **end if**
 6. **removed any URL links from** *i*
 7. **if** *i* not exactly contains one hashtag **then**
 8. **continue**
 9. **end if**
 10. **get hashtag from** *i*
 11. **split hashtag into words**
 12. **tw.append((*i*, hashtag))**
 13. **end while**
 14. **return** *tw*
-

We constructed the training data, testing data and evaluation data where we found words in the “#ipl2021” more of the value 6131 compared to the rest of the data. The “TNelections” data was found to be having a smaller number of words in the training test. We separated the testing data with a text size of 0.2. The data is then summarized by the BERT SUM and preprocessed by the steps mentioned earlier. After all the are steps done, the training set generated for all the tweets are listed below (Table 1).

Table 1 Dataset summary

Name	Training data (words)	Testing data (words)	Evaluation data (words)
#Covid-19	4504	1180	153
#Ipl2021	6131	1506	236
#oxygen	5233	1326	125
#TNelections	3289	817	112

4.2 Experimental Setup and Evaluation

The proposed framework is implemented using python. Keras, Sklearn, Matplotlib, NLTK, Numpy and Pandas were the libraries used for various tasks. We used four real-time collected tweet datasets that can be used for a similar study. As the data is not already available in any public contributions, the gold standard keyphrase list is not available with it. Hence, we generated the gold standard list with use of hashtags and frequent words as mentioned above. The candidate keyphrases which are extracted by BERT SUM are labelled using the gold-standard keyphrase list. We conducted a performance evaluation with the Bi-LSTM model on our four constructed training datasets. Bi-LSTMs were fed with the GloVe vector equal to each word in the training set and the model with the vector and label classifies each of the keyphrase from the non-keyphrase. The result obtained would be of three classes S , I , 0. The predicted keyphrases would have the classes S and I and 0 would be the class for the non-relevant keyphrase. We compared the proposed model to the existing available models to investigate the performance of pre-trained Twitter GloVe embedding to the Bi-LSTM. We use precision-recall and F1 score for evaluating the model. All the evaluation metrics above mentioned were used in previous works and are accepted worldwide [1, 3, 21].

$$\text{Precision} = \frac{\text{number of correctly matched}}{\text{total number of extracted}} = \frac{p}{p + q} \quad (1)$$

$$\text{Recall} = \frac{\text{number of correctly matched}}{\text{total number of assigned}} = \frac{p}{p + r} \quad (2)$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2p}{p + q + r} \quad (3)$$

Table 2 Confusion matrix for keyphrase classification

	Classified as keyword by ground truth	Classified as non-keyword by ground truth
Classified as keyword by the model	p	q
Classified as non-keyword by the model	r	s

Table 3 Evaluation metrics on different datasets by BERT SUM—Bi-LSTM model

Data	P (%)	R (%)	F1 (%)
#Covid-19	86.02	100	91
#Ipl2021	84.12	92	92
#oxygen	84.75	93	92
#TNelections	85.59	92	91

4.3 Results

The experiment was done on all four datasets we collected. The results of our experiments for the four datasets are shown in this Table 2. It was seen that the model was well-performing in extracting the keyphrases for all four models.

The performance of the model on the covid19 dataset was high and the performance of the model on ipl2020 dataset was the least. Recall value was interestingly very higher on the covid19 dataset compared to that of the other three datasets. From Table 3 we find the precision, recall and F1-scores to be well high and are based on the correct prediction of the keyphrases which makes the model well-performed compared to the previous state-of-the-art models.

4.4 Discussion

The model generally scored around 85 percent accuracy when trained on general data not focused on a specific domain. The results show that the model has performed high to the previously available keyphrase extraction models, where they use many more features and statistical information attached to the training data. The unique way of extracting the candidate keyphrases using BERT SUM improved the performance of the model as the BERT SUM finds the sentences related to the topic and rank it with a score by which the output summary consists of only the words which are relevant to the topic. So, this makes the model even to be precise in the keyphrase extraction task. The sequence labelling technique used in the model is the state-of-the-art method for labelling the sequences. This has also made the model learn the

Table 4 Evaluation metrics of proposed BERT SUM—Bi-LSTM model compared with different existing models

Model	P (%)	R (%)	F1 (%)
Joint-layer RNN	87.45	85.38	86.40
J-Bi-LSTM+IPA+POS	69.06	61.98	65.33
AKET	20.68	87.56	33.46
Proposed model	85.12	94.25	91.50

sequences correctly and that helped the model to outperform. Bi-LSTM used in the model has also helped the model to perform well as the model keeps the words in the memory which helps well in the keyword extraction task. Finally, the model is seen to have a high precision score and recall for the #Covid-19 data and a higher F1-score on #Tnelections data which proves the model recalling the words which it has learned in the training phase. This shows that the proposed model has performed well in extracting the keyphrase. In summary, the experimental results conclusively show the proposed BERT SUM—Bi-LSTM method to be outperforming the state-of-the-art methods when evaluated by commonly accepted evaluation metrics on Twitter data.

From Table 4, comparing the performance with the existing models the proposed model had high performance and only the precision of the model was low yet F1-score was high compared to the other models.

5 Conclusion

Recent approaches of Keyphrase extraction from twitter data has shown that the supervised model outperform the unsupervised counterparts. This has made researchers focus on enhancing the supervised keyphrase extraction methods. Hence, in this work, we focused on the supervised keyword extraction, where we proposed a novel BERT SUM with Bi-LSTM method where BERT SUM summarizes and extracts the candidate keyphrases and Bi-LSTM classifies the keyphrase from non-keyphrase. We have overcome the labelling problem by using sequence labelling technique. We found the BERT Summarization method of extracting the candidate keyphrase has helped the model to be performing well focusing on the candidate keyphrases it has extracted in the task of keyphrase extraction. We also found the proposed model outperformed the state-of-the-art methods and the results show the performance of the proposed model for keyphrase extraction from Twitter data has enhanced compared to the existing models. We also found the model performed with high precision and recall score for Covid19 data and least performed at the ipl2021 data. In our future, we will try to enhance the extraction process by focusing on the construction of the gold-stand keyphrase list without depending on the hashtag words. Some tweets can contain hashtag words which are irrelevant to the topic or some tweets which are irrelevant to the topic can be given the hashtag related to what subject we search. So, carrying out the model with those tweets would make the

model to perform irrelevant and extracting keyphrases irrelevant to the topic. Also, the model depends on the gold-standard keyphrase list for labelling the data. In a supervised model, real-time data labelling is dependent on gold-standard keyword list. Hence, we try to interrogate on the other ways of construction of the gold-standard keyword list which make the model independent of the hashtag words.

Acknowledgements The authors gratefully acknowledge the Science and Engineering Research Board (SERB), Department of Science & Technology, India for financial support through Mathematical Research Impact Centric Support (MATRICS) scheme (MTR/2019/000542). The authors also acknowledge SASTRA Deemed University, Thanjavur for extending infrastructural support to carry out this research work.

References

1. Zhang Q, Wang Y, Gong Y, Huang XJ (2016) Keyphrase extraction using deep recurrent neural networks on twitter. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 836–845
2. Ray Chowdhury J, Caragea C, Caragea D (2019) Keyphrase extraction from disaster-related tweets. In: The world wide web conference, pp 1555–1566
3. Turney PD (2003) Coherent keyphrase extraction via web mining. arXiv preprintcs/0308033
4. D'Avanzo E, Magnini B (2005) A keyphrase-based approach to summarization: the lake system at duc-2005. In: Proceedings of DUC
5. Danilevsky M, Wang C, Desai N, Ren X, Guo J, Han J (2014) Automatic construction and ranking of topical keyphrases on collections of short documents. In: Proceedings of the 2014 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, pp 398–406
6. Elangovan R, Vairavasundaram S, Varadarajan V, Ravi L (2020) Location-based social network recommendations with computational intelligence-based similarity computation and user check-in behaviour. *Concurrency Comput Pract Exp* 24:e6106
7. Wang J, Peng H, Hu JS (2006) Automatic keyphrases extraction from document using neural network. In: Advances in machine learning and cybernetics. Springer, Berlin, Heidelberg, pp 633–641
8. Duari S, Bhatnagar V (2020) Complex network based supervised keyword extractor. *Expert Syst Appl* 140:112876
9. Wu YF, Li Q, Bot RS, Chen X (2005) Domain-specific keyphrase extraction. In: Proceedings of the 14th ACM international conference on Information and knowledge management, pp 283–284
10. Florescu C, Caragea C (2017) A position-biased pagerank algorithm for keyphrase extraction. In: Proceedings of the AAAI conference on artificial intelligence, vol 31(1)
11. Liu Z, Li P, Zheng Y, Sun M (2009) Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 conference on empirical methods in natural language processing, pp 257–266
12. Liu Z, Huang W, Zheng Y, Sun M (2010) Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 conference on empirical methods in natural language processing, pp 366–376
13. Bellaachia A, Al-Dhelaan M (2012) Learning from twitter hashtags: leveraging proximate tags to enhance graph-based keyphrase extraction. In: 2012 IEEE International conference on green computing and communications. IEEE, pp 348–357

14. Papagiannopoulou E, Tsoumakas G (2020) A review of keyphrase extraction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10(2):e1339
15. Zhao WX, Jiang J, He J, Song Y, Achanauparp P, Lim EP, Li X (2011) Topical keyphrase extraction from twitter. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp 379–388
16. Meladianos P, Nikolentzos G, Rousseau F, Stavrakas Y, Vazirgiannis M (2015) Degeneracy-based real-time sub-event detection in twitter stream. In: Proceedings of the international AAAI conference on web and social media, vol 9(1)
17. Sahrawat D, Mahata D, Kulkarni M, Zhang H, Gosangi R, Stent A, Sharma A, Kumar Y, Shah RR, Zimmermann R (2019) Keyphrase extraction from scholarly articles as sequence labelling using contextualized embeddings. arXiv preprint [arXiv:1910.08840](https://arxiv.org/abs/1910.08840)
18. Marujo L, Ling W, Trancoso I, Dyer C, Black AW, Gershman A, de Matos DM, Neto JP, Carbonell JG (2015) Automatic keyword extraction on twitter. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, vol 2: Short Papers, pp 637–643
19. Litvak M, Last M, Kandel A (2013) Degext: a language-independent keyphrase extractor. J Ambient Intell Humaniz Comput 4(3):377–387
20. Hulth A (2003) Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on empirical methods in natural language processing, pp 216–223
21. Nguyen TD, Kan MY (2007) Keyphrase extraction in scientific publications. In: International conference on Asian digital libraries. Springer, Berlin, Heidelberg, pp 317–326
22. Yih WT, Goodman J, Carvalho VR (2006) Finding advertising keywords on web pages. In: Proceedings of the 15th international conference on World Wide Web. Yih et al (2006)
23. Florescu C, Caragea C. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 1: Long Papers, pp 1105–1115
24. Medelyan O, Frank E, Witten IH (2009) Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 conference on empirical methods in natural language processing, pp 1318–1327
25. Sarkar K, Nasipuri M, Ghose S (2010) A new approach to keyphrase extraction using neural networks. arXiv preprint [arXiv:1004.3274](https://arxiv.org/abs/1004.3274)
26. Caragea C, Bulgarov F, Godea A, Gollapalli SD (2014) Citation-enhanced keyphrase extraction from research papers: a supervised approach. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1435–1446
27. Hammouda KM, Matute DN, Kamel MS (2005) Core phrase: Keyphrase extraction for document clustering. In: International workshop on machine learning and data mining in pattern recognition. Springer, Berlin, Heidelberg, pp 265–274
28. Osiński S, Stefanowski J, Weiss D (2004) Lingo: search results clustering algorithm based on singular value decomposition. In: Intelligent information processing and web mining. Springer, Berlin, Heidelberg, pp 359–368
29. ALRashdi R, O'Keefe S (2019) Deep learning and word embeddings for tweet classification for crisis response. arXiv preprint [arXiv:1903.11024](https://arxiv.org/abs/1903.11024)
30. Liu Y (2019) Fine-tune BERT for extractive summarization. arXiv preprint [arXiv:1903.10318](https://arxiv.org/abs/1903.10318)

Improving the Yield and Revenue of Indian Crop Production Using Data Engineering



Jayashree Domala, Manmohan Dogra, Kevin Dsouza, Dwayne Fernandes, and Anuradha Srinivasaraghavan

Abstract Agriculture is one of the top three contributors to the Indian economy. However, most of the agricultural practices are still quite traditional in nature. Farmers have to make innumerable decisions on various factors which in return affect the yield and productivity of the crop production. Some of these factors include season, yield, location, revenue, type of crop, etc. If farmers and agricultural businesses made decisions based on engineered crop data of various years, then the overall crop production and yield can be maximized. This paper presents a data engineering procedure to analyze crop data and provide recommendations. It is divided into three main modules, viz recommendation of crop season and location, recommendation of crop harvesting months and comparison of fruit and vegetable crops. The datasets are procured from the Open Government Data (OGD) Platform of India and the agricultural produce marketing committee (APMC). The techniques involve data collection, preprocessing, normalization and data querying. Interactive data visualizations were performed with the help of Plotly library in Python. The resultant system provides suggestions based on season, location and revenue. Moreover, it provides suggestions of crops based on comparison with respect to yield.

Keywords Crop recommendation · Data engineering · Crop analysis · Agriculture · Farming

1 Introduction

India has been an agriculturally predominant nation from several countries. It has adopted several farming techniques such as multi-layer farming, permaculture, zero budget natural farming and hydroponic farming in recent years. Even with such advanced techniques, there are several other critical factors that ought to be taken

J. Domala (✉) · M. Dogra · K. Dsouza · D. Fernandes · A. Srinivasaraghavan
St. Francis Institute of Technology, Borivali-West, Mumbai, Maharashtra, India
e-mail: g.anuradha@sfit.ac.in

into consideration that yield questions pertaining to identifying the right crop to maximize profit or the knowledge of whether or not the given conditions would best suit a given crop.

The most obvious approach to the aforementioned questions has always been to consult experts and agronomists who have significant knowledge in the same. However, the drawbacks associated with this approach are not just the farmer having to consult expert agronomists several times but also the uncertainties that are introduced with the results provided by different agronomists due to the human bias. A lot of the decisions taken by agro-experts are heavily influenced by past experience and may not take into account the dynamically changing conditions. This can, however, be abolished by using technology to achieve standardization in results while maximizing the decisions that directly influence the profits for that given crop.

A smart recommender system for crops is a potential solution to this problem which makes critical decisions by mining useful insights gained from the historical data of crops cultivated over several decades in India under different conditions. A statistical analysis that takes into account the cropping patterns, location, area, yield, production cost, cultivation cost, market value, etc., could prove to be very useful to any farmer for making informed decisions pertaining to crop selection at any given time.

The next section provides a review of the related work. In Sect. 3, the implementation accounts for the detailed explanation of three major modules of the project which helps in recommending season, the harvest month and a comparative analysis of various fruits and vegetables from the historical yield data that was accumulated. Experimental results achieved from our technique are presented in Sect. 4 which follows a conclusion that states the benefits of our technique to get advanced recommendations for crops based on several factors along with the future scope for its expansion. The research paper concludes with all the references and citations that were taken into consideration during the research.

2 Related Work

Numerous specialists have researched on crop recommendations using a multitude and varied algorithms. Some selective research studies we came across are as follows. Majumdar et al. [1] presented an approach for accurate yield estimation of several crops using data mining techniques. The authors have taken into consideration factors such as environmental conditions, soil variability, commodity prices and input levels, affecting crop yield for achieving a realistic and efficient solution to the stated problem. A rigorous data analysis on the data influencing crops helps in finding optimal parametric values to get maximum crop production. Data mining techniques such as partition around medoids (PAM), multiple linear regression (MLR), clustering large applications (CLARA) and density-based spatial clustering of applications with noise (DBSCAN) are used to get the optimal range of best temperature, worst

temperature and rainfall. During a comparative study of the mentioned algorithms, it was observed that DBSCAN produced the best results compared to MLR, PAM and CLARA.

Kumar et al. [2] proposed an estimation model of crop yield by utilizing the MLR data mining technique. An MLR model is used for statistical analysis of the data and crop production gathered for explicit states in India. The analysis and comparisons are performed on four major crops (arhar, barley, maize and potato) using analysis of variance (ANOVA) single-factor and two-factor study depending on factors like cultivation, season, area and production for the crop yield. Results were verified through the IBM Statistical Package for the Social Sciences (SPSS) package (a statistical software). The results are not very promising, and a conclusion is made that more effective techniques can be developed and utilized for complex crop yield analysis.

Ramesh [3] applied MLR and density-based clustering concerning an area explicit yield dataset for the crop evaluation. A predictive MLR model based on the least square method is built for developing models to reconstruct climate variables in climatology. The MLR model predicts the amount of production possible for given input predictors (year, rainfall, sowing area, fertilizer and yield). The estimated result of production by the model and the actual production amount for the past 20 years are compared and analyzed. The results verified that the density-based clustering method produced more relative prediction. The suggested technique is implemented over a large-scale area for Andhra Pradesh increasing the quality of cultivated crops.

Champaneri et al. [4] focused on estimating the yield of the cultivation before the farmer cultivates on land using random forest classifier. The machine learning model is trained on a dataset having input predictors as precipitation, temperature, cloud cover, vapor pressure and wet day frequency of specific crops. The model performs with an accuracy of 75%. The consolidated analysis result and the yield prediction regarding the crop are provided to the farmer on a Web-based user interface.

Sekhar et al. [5] presented a data clustering algorithm for planning the sowing and harvesting season of the crops utilizing big data analytics frameworks for faster processing. A clustering algorithm is developed for analyzing crop sales using the Hadoop platform. A predictive model using a decision tree is built for grouping the data based on the pattern observed. In addition to that, the K-means algorithm is also used for predictive analytics of future crop prices to help farmers adopting a cultivation plan. The clustering algorithm categorizes the crops according to the market demand of crops, resulting in more efficient production.

Ms. Fathima et al. [6] researched increasing crop yield productivity by using data mining and regression techniques. Crop production is analyzed using factors such as rainfall, temperature, atmospheric conditions, pesticides and fertilizers. A K-nearest neighbors (KNN) model is built for predicting the crop yield for particular regions (Mangalore, Kasargod, Hassan). The KNN model performs with an accuracy of 90%.

Khan et al. [7] present a method to improve the quality and quantity of tomatoes by early detection of disease by scanning tomato leaves. Deep learning-based disease detection and classification model is built in MATLAB for preprocessing and detection of the leaf image. Upon detecting, an appropriate action is taken to cure

Table 1 Comparison of the related work

Reference No.	Algorithm	Result
[1]	PAM, Multiple linear regression, CLARA and DBSCAN	Analysis is performed on five crops (cotton, wheat, ground nut, jowar and rice). DBSCAN gives the better clustering quality than PAM and CLARA, and CLARA gives the better clustering quality than the PAM
[2]	MLR model using ANOVA single-factor and two-factor study	Four crops production and different season (rabi, kharif and summer) were compared and analyzed
[3]	Multiple linear regression and density-based clustering	Linear regression algorithm applied on three crops dataset offered acceptable estimation accuracy (95%). Only soil information is utilized to train the model
[4]	Random forest algorithm	Accuracy of the model predicting yield has reached upto 75%. 12 crop types data of a single district is used in model training
[5]	K-means clustering algorithm	Six clusters are effectively formed for 124 varieties of crop using HDFS system
[6]	K-nearest neighbors model	The model predicts with an accuracy of 90% by taking average rainfall and area of field into account
[7]	Deep learning-based disease detection and classification model	Provides automated plant disease detection reducing manual labor

the plant quality. This approach reduces the manual labor required for inspecting defective plants and increases crop production significantly.

Considering the previous research already done in this area, the proposed system focuses more on the number of crops of varied locations using multiple sources of data collection. The main goal is to implement a more convenient and sophisticated system that is reliable. A succinct summary of the related papers is demonstrated in Table 1.

3 Implementation

The implementation process is bifurcated into three modules so as to look into every aspect of the crop analysis. The three modules are as follows:

1. Recommendation of season and location for a crop for maximum yield.
2. Recommendation of harvesting months for a crop for maximum revenue.
3. Comparison of fruit and vegetable crops on basis of yield.

3.1 Module 1: Recommendation of Season and Location for a Crop for Maximum Yield

Based on the crop name, the seasons and the location best suited for the crop are recommended. The Open Government Data (OGD) Platform of India was used to get the dataset. The dataset refers to year-wise, crop-wise, district-wise and season-wise data on crop covered area (in Hectare) and production (in Tonnes). The seasons are kharif, rabi, the whole year, summer, winter and autumn. There are 33 states and 124 crops for which the analysis is done. The workflow is shown in Fig. 1.

3.1.1 Step 1: Data Collection

The dataset consists of 246091 data points belonging to 7 attributes. The seven attributes are name of the state, the name of the district, the year the crop was grown, the season the crop was grown, the area under which the crop was grown and the production of the crop. The data for the crops is from the year 1997 to the year 2015.

3.1.2 Step 2: Preprocessing the Data

The dataset is checked for any NULL values, and the corresponding data tuples are deleted for efficiently analyzing and obtaining accurate results. Along with this, few data rows are duplicated, and hence, they are also omitted to avoid redundancy.

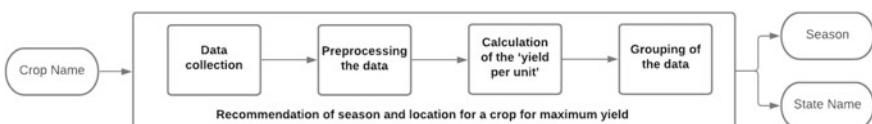


Fig. 1 Workflow for recommendation of season and location for a crop for maximum yield

3.1.3 Step 3: Calculation of the ‘Yield per Unit’

With the help of the area and the production values, the ‘yield per unit’ is calculated. The formula used is

$$\text{Yield_per_unit} = \text{production}/\text{area} \quad (1)$$

3.1.4 Step 4: Grouping of the Data

The data is still in the raw form which cannot be used to output the data in the desired form. The grouping of data is done based on the crop name, state name and the season and then ordered in the descending form of the yield. Now, the data is in the required format, and when queried by inputting a crop name, the season and location suited will be returned.

3.2 Module 2: Recommendation of Harvesting Months for a Crop for Maximum Revenue

Based on the crop name, the harvesting months best suited for the crop are recommended using the maximum and minimum prices. The harvesting months can be any of the 12 months of a year, but the output returned is the three best consecutive months to harvest a particular crop. The workflow is shown in Fig. 2.

3.2.1 Step 1: Data Collection

Two datasets are used for the purpose of analysis. Both of them are taken from the agricultural produce marketing committee (APMC) Web site which has the commodity-wise arrival reports and average rates. The former ‘yield dataset’ consists of the monthly yield of 167 crops for 12 months for the year 2013 to 2019 except for the

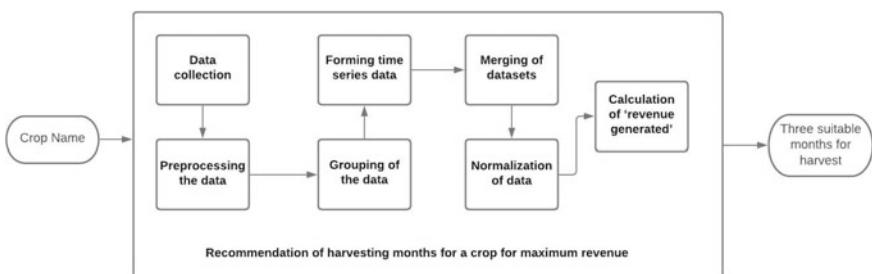


Fig. 2 Workflow for recommendation of harvesting months for a crop for maximum revenue

year 2018. The later ‘price dataset’ consists of the monthly maximum and minimum prices of the same 167 crops for 12 months for the year 2014 to 2019.

3.2.2 Step 2: Preprocessing the Data

Null values and duplicate data rows are removed from both datasets for ease of analyzing.

3.2.3 Step 3: Grouping of Data

The yield dataset has columns for the year, crop name and 12 more columns corresponding to the monthly yields. This dataset is multi-indexed to have two indexes—year and crop name. Now, using these index values, for the same year and crop, the yield is summed, and then, the indexes are reset again.

3.2.4 Step 4: Forming Time Series Data

Now for efficient analysis, the yield dataset is converted into the time series format, thereby creating a dataset with the attributes—date, crop name and yield. The same is done for the price dataset which gives a dataset with attributes—date, crop name, maximum price and minimum price.

3.2.5 Step 5: Merging of Datasets

The next step is to merge the yield dataset and price dataset in the time series format into one single dataset. This gives us information on the yield, maximum price and minimum price of a crop for all the months from the year 2013 to 2019.

3.2.6 Step 6: Normalization of Data

Since the scale at which the yield and prices differ by a huge margin, the data is normalized so as to reduce the gap and bring all the attributes to the small scale.

3.2.7 Step 7: Calculation of ‘Revenue Generated’

Once the normalization is performed, the next step is to calculate the revenue. It is calculated by using the formula:

$$\text{Revenue} = \text{yield} * \text{maximum_price} \quad (2)$$

Now, based on the revenue values, the three harvest months are outputted for the required crop where the revenue generated is the highest.

3.3 Module 3: Comparison of Fruit and Vegetable Crops on Basis of Yield

Based on the multiple crop names queried, they will be compared based on the yield, and the best crop to grow will be returned. The workflow is shown in Fig. 3.

3.3.1 Step 1: Data Collection

Two different datasets are used for analysis. One is the Indian fruit production dataset. It has data for 67 fruit crops, and the attributes of the data are crop name, year, crop area, crop production and crop yield. The data is for the years ranging from 2014 to 2019. The other is the Indian vegetable production dataset. It has data for 35 vegetable crops. The attributes are crop name, crop yield and crop area. This data ranges from 1998 to 2019. Both the datasets are taken from the Open Government Data (OGD) Platform of India.

3.3.2 Step 2: Preprocessing the Data

In the dataset to avoid redundancy, the duplicate values are removed. Along with this, the NULL valued rows are also deleted.

3.3.3 Step 3: Normalization of the Data

In both datasets, the crop yield and area values are in different ranges. To efficiently analyze the data, they are normalized to bring them on the same scale of range.

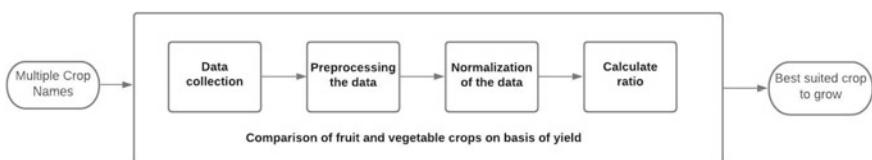


Fig. 3 Workflow for comparison of fruit and vegetable crops on basis of yield

3.3.4 Step 4: Calculate Ratio

Next, the ratio of the yield concerning the area is calculated for each crop. Now, when the multiple crop names are inputted, their respective ratios are compared, and the crop with the maximum ratio is selected.

4 Results

The data engineering is successfully carried out on varied Indian crops for a range of years. The chart analysis is created for each of the modules for all the crops supported through interactive plotting. These plots can help look at the crop analysis all at once or as selected by the user as demonstrated in Figs. 4, 5, and 6. For ease of understanding, these charts are thereafter converted to visualizations. For the first module, the locations and seasons are shown visually on the map of India shown in Fig. 7. Moreover, the second module of harvesting month recommendation is shown in the form of a calendar in Fig. 8. Lastly, the comparison is shown in the form of pie charts in Fig. 9.

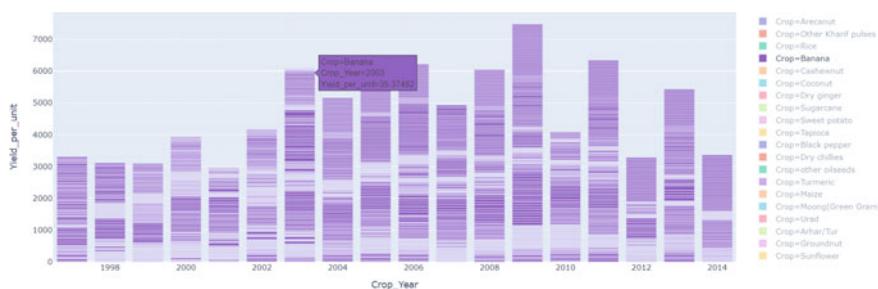


Fig. 4 Graph showing the yield for crops from 1997 to 2014

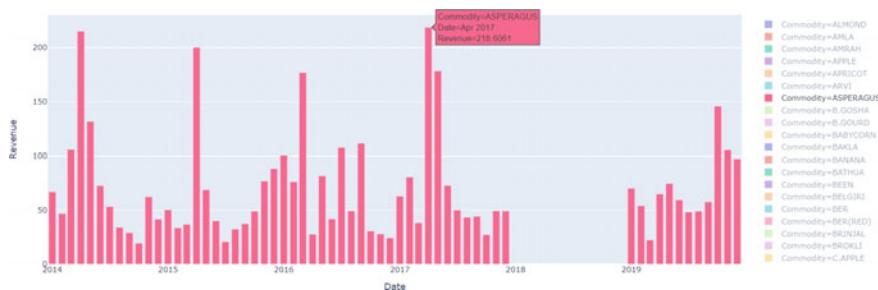


Fig. 5 Graph showing the revenue for crops for all months from 2014 to 2019

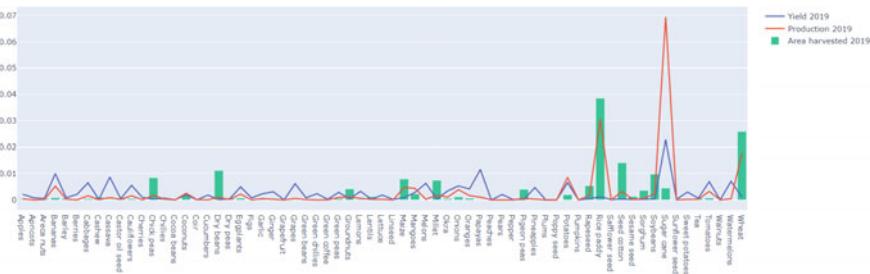


Fig. 6 Graph showing the yield, production and area of crops for the year 2019

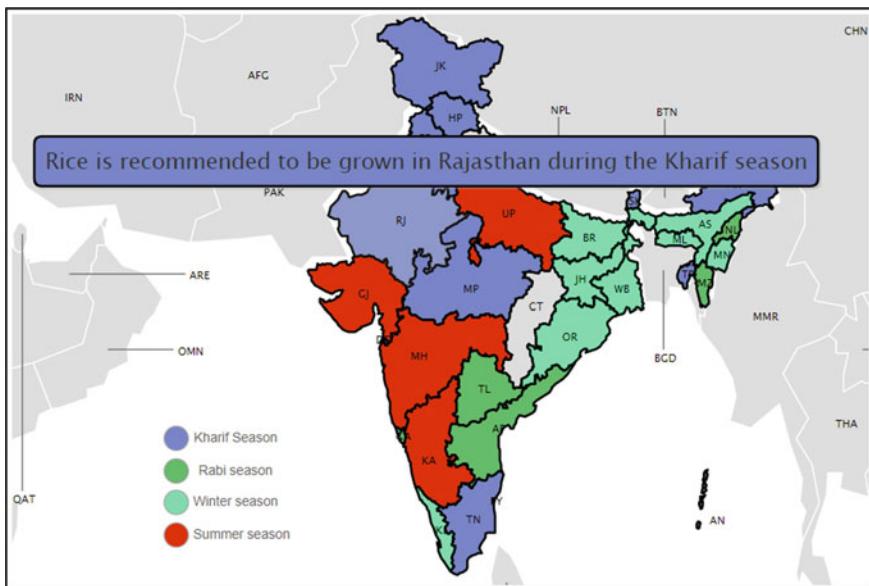


Fig. 7 Visualization for the recommendations of season and state for the crop rice

5 Conclusion

In this paper, data analysis was carried out for various Indian crops using the available datasets on Indian digital platforms. Data engineering for more than 6 years of crop data was conducted which provided critical and significant insights into the revenue generation, yield production, seasonality and location of the crop to be grown. Dividing the entire analysis into three modules aided in evaluating how varied factors decided the optimal crop to be grown. The results of this system can help various

Fig. 8 Visualization of the harvesting months for the crop almond

Harvesting Month Recommendation

ENTER THE CROP NAME AND GET THE RECOMMENDATION!

SHOW RECOMMENDATION

MAY											
1	2	3	4	5	6	7	8	9	10	11	
12	13	14	15	16	17	18	19	20	21	22	
23	24	25	26	27	28	29	30	31			

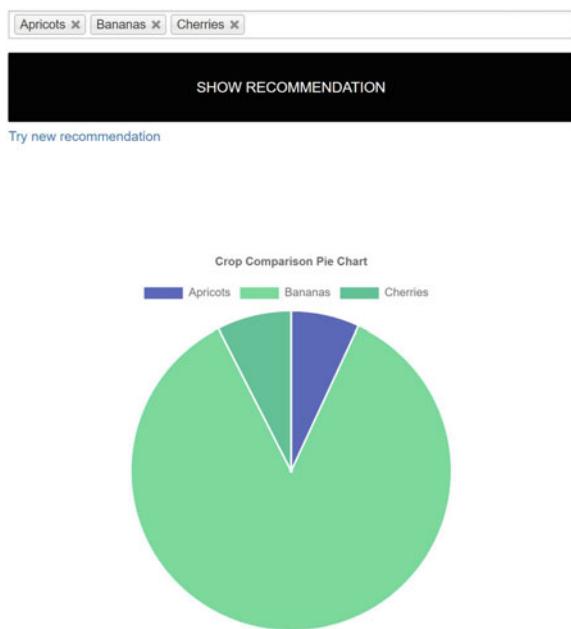
JUNE											
1	2	3	4	5	6	7	8	9	10	11	
12	13	14	15	16	17	18	19	20	21	22	
23	24	25	26	27	28	29	30				

JULY											
1	2	3	4	5	6	7	8	9	10	11	
12	13	14	15	16	17	18	19	20	21	22	
23	24	25	26	27	28	29	30	31			

agronomists and farmers with meaningful insights where the process of crop production is easily ready. This could result in higher yields, faster growth, quality yields and maximum revenue generation.

A future scope is to take into consideration the crop growing parameters like irrigation levels, pH levels and fertilizer usage along with the external climatic factors like temperature, humidity and rainfall to improve the crop health, thereby increasing the revenue generated and yield produced.

Fig. 9 Visualization of the comparison of the crops apricots, bananas and cherries



References

1. Majumdar J, Naraseeyappa S, Ankalaki S (2017) Analysis of agriculture data using data mining techniques: application of big data. *J Big Data* 4:20
2. Kumar R, Kumar Yadav S, Kumar Sharma T (2019) Estimation of major agricultural crop with effective yield prediction using data mining, 1st edn. [ebook]. Blue Eyes Intelligence Engineering & Sciences Publication, p 5
3. Ramesh D (2015) Analysis of crop yield prediction using data mining techniques. *Int J Res Eng Technol* 04:470–473. <https://doi.org/10.15623/ijret.2015.0401071>
4. Champaneri M, Chachpara D, Chandavidkar C, Rathod M (2020) Crop yield prediction using machine learning. *International Journal of Science and Research (IJSR)*. 9:2
5. Sekhar C et al (2018) Effective use of Big Data Analytics in Crop planning to increase Agriculture Production in India. *International journal of advanced science and technology* 113:31–40
6. Ms. Fathima K, Barker S, Kulkarni S (2020) Analysis of crop yield prediction using data mining techniques. <https://doi.org/10.13140/RG.2.2.14424.52482>
7. Khan ES, Saemeen K, Parveen D, Jannat P (2018) Analysis of data mining techniques for agricultural science. In: International conference on smart city and emerging technology (ICSCET), vol 2018, pp 1–6. <https://doi.org/10.1109/ICSCET.2018.8537322>

Designing an LSTM and Genetic Algorithm-based Sentiment Analysis Model for COVID-19



Poonam Rani, Jyoti Shokeen, Arjun Majithia, Amit Agarwal,
Ashish Bhatghare, and Jigyasu Malhotra

Abstract The unleashing of Coronavirus on human lives has drastically changed a lot of things. The pandemic has been a difficult time for everybody as people lost their jobs and businesses, the economy dwindled, health issues due to the virus, be it physical or mental, were prevalent. Loneliness, depression, and anxiety caused by lockdown and work from home became the new normal. It, therefore, becomes imperative to study the large amount of social media data using computational methods and gauge the sentiment of people on various policies and strategies undertaken to fight the pandemic and take decisions accordingly. We introduce a Social Media Pandemic Sentiment Model on COVID-19 Twitter dataset to study the sentimental variation in people throughout the duration of pandemic and derive useful results out of it. We also provide an extensive comparative analysis of this model with other conventional states of the art models to display the competence of our model.

Keywords COVID-19 · Sentiment analysis · Social media · LSTM · Genetic algorithm

1 Introduction

The world witnessed the worst possible outbreak of COVID-19 or Corona Virus in the year 2019 by affecting every segment of human life. This pandemic brought the world to a standstill as a large number of countries announced complete lockdown of its cities to fight against the pandemic. Every organization, be it education, economy, travel, hospitality, entertainment, sports, or food, there has been a complete change in the way they work. Other major effects of lockdown are the increasing deterioration of

P. Rani · A. Majithia · A. Agarwal (✉) · A. Bhatghare · J. Malhotra
Department of Computer Engineering, Netaji Subhas Institute of Technology,
University of Delhi, Delhi, Delhi, India

J. Shokeen
Department of Computer Science and Engineering UIET, Maharshi Dayanand
University Rohtak, Rohtak, Haryana, India

people's mental health and the difficulty in staying home throughout the day without being involved in any social activity or travel. As we are moving into this changed world, things are being done in an online-offline hybrid fashion, with organizations also taking work-from-home as a serious contender for the future. This research carries out an in-depth sentiment analysis of the people throughout the period of pandemic based upon tweets made by people on Twitter. This research work is helpful in determining the human behavior and sentimental differences caused by lockdown.

We designed a deep learning-based model to classify the sentiments of tweets. This model uses heuristic genetic algorithm to improve the model efficiency and achieve peak results. It also incorporates LSTM to learn dependencies between distant words. The main contribution of this paper is to improve the performance of classifying the tweet's sentiments by combining genetic algorithm, and LSTM model. Section 2 gives a brief description of the approaches used in the proposed model. Section 3 provides related works in this area. The proposed model is explained in Sect. 4. Section 5 provides a comparison of the proposed model with already existing competitive and latest algorithms on different performance measures in this domain. Lastly, Sect. 6 concludes the paper.

2 Models Used

2.1 LSTM

Long Short Term Memory (LSTM) is an artificial recurrent neural network (RNN) but is different from usual feed forward networks. LSTM is a feedback network and processes the entire sequence of data, not just single data points. RNNs suffer from a problem called vanishing gradient problem that impedes learning of long data sequences. Since the layers receiving small gradients stop learning over time, RNNs suffer from short memory. LSTMs resolve this issue as they have direct access to forget gate's activation to regulate information flow and hence can learn data sequences. LSTMs are used in sentiment analysis as they enable learning long distance dependency between words. LSTM models can find sentiment of a given text and classify it.

2.2 Genetic Algorithm

Genetic Algorithm (GA) is a heuristic and search optimization technique inspired from natural evolutionary process. GA is used to search the optimal solution in a given search space. The operators used to replicate principles of natural genetic evolution are selection, crossover, and mutation. The population of chromosomes is the

crux of GA. Each potential solution is represented by a chromosome. Chromosomes are initialized randomly and “survival of the fittest” ideology is used in the sense that only those chromosomes which are a better solution get the chance to reproduce. The first step involves random initialization of chromosomes in a population. Next, fitness function is used to evaluate fitness which is symbolic of the performance of chromosomes. The next step involves selection of parents for reproduction based upon their fitness. The selected parent chromosomes then undergo crossover to produce offspring. Mutation is the process by which small modification or change is done in order to create a new individual representing another solution. If this individual is more fit for survival, then it will replace the other most weak chromosome from the set. This is called survivor selection. Through mutation, the solution search space is explored for any optimal solution. Finally, it checks for convergence and determines if search space has been searched optimally.

3 Related Works

A lot of work has already been done in this area. Wang et al. [9] utilized word embedding and LSTM for finding the sentiments of social media posts. They first converted the posts into vectors using a model of word embedding and then used LSTM to find the final sentiment of the post. Xu et al. [11] propose an improved approach of word representation, which combines sentiment information with the conventional TD-IDF approach and finally, weighted word vectors are produced. To capture information about context effectively and to represent comment vectors better, weighted word vectors are input into BiLSTM, and then by using a feedforward neural network, sentiment tendency is obtained.

Chakraborty et al. [3] pointed out in their study that tweet handles for COVID-19 and WHO were unsuccessful in providing guidance to people about this pandemic. They analyzed two datasets where the first dataset shows people having negative to neutral attitudes, and the second one shows people having positive to neutral attitudes.

Xu et al. [11] propose an improved approach of word representation, which combines sentiment information with the conventional TD-IDF approach and finally, weighted word vectors are produced. To capture information about context effectively and to represent comment vectors better, weighted word vectors are input into BiLSTM, and then by using a feedforward neural network, sentiment tendency is obtained.

Arora and Kansal [2] proposed a model to predict sentiments of tweets as neutral, negative, or positive. The model uses a neural network that has text normalization with deep convolutional character level embedding. Yang et al. [12] proposed a hierarchical attention network-based model to classify documents. This model consists of two layers: first layer mirrors the hierarchical structure of a document and second one has two levels of attention mechanism, namely word level mechanism, and sentence level mechanism, to give different attention to different types of information based on their importance.

Wang et al. [10] attempted to reduce the dependency of machine learning techniques on human annotation. They combined attention mechanism and Bi-directional LSTM to propose a hybrid model called AM-Bi-LSTM. Rani et al. [4] performed a qualitative and quantitative analysis of social media platforms to study the nature of communication.

Recently, Aljameel et al. [1] developed a model to predict awareness of precautionary measures in Saudi Arabia. Several predictive models were tested on the Arabic dataset out of which SVM classifier outperformed other models. The authors observed that the south region of Saudi Arabia showed the highest level of awareness while the middle region showed the least. On the other hand, Singh et al. [8] proposed a sentiment analysis model using BERT (Bidirectional Encoder Representations from Transformers) model to classify public opinions about coronavirus based on tweets. They used two datasets for sentiment analysis in which the first dataset consists of tweets from all over the world and the second contains tweets from India only.

4 Social Media Pandemic Sentiment Model

In this paper, we propose a deep learning-based model architecture for sentiment analysis on COVID-19 Twitter dataset. Neural networks are a very powerful and efficient technique for image and text classification and can be used for a variety of applications in deep learning like recommender systems [6, 7] and stock price prediction [5]. A limitation of neural networks is that it does not have any considerations or scope of memory. As a result, the training is not influenced by past history which becomes problematic especially in case of text classification wherein the order of words plays a huge role in determining the gist and sentiment of the text. RNN solves this problem by having a feedback looping mechanism that acts as memory. LSTM goes one step further by incorporating a short-term and long-term memory component which makes it even more suitable for sequence based applications like texts. GA is an evolutionary search optimization algorithm that searches for optimal parameters to achieve peak accuracy.

We combine LSTM with GA to design Social Media Pandemic Sentiment Model. SMPSM aims to detect the sentiments during pandemic time using Twitter dataset. In the proposed architecture, we propose an LSTM model with sequential input layer and optimal number of hidden layers each of which has an optimal number of neurons. These optimal parameters are searched and chosen by GA. Softmax is used as an activation function at output layer. The aim is to predict sentiment of a tweet and classify it in positive, negative, or neutral. GA is employed to search for the most optimal window size and optimal number of units in the LSTM network. To evaluate the fitness of GA, different numbers of LSTM units and window sizes are used. The possible contender solution populations are randomly initialized and search space is investigated by the genetic operators. Binary bit encoding that represents the window and number of LSTM units is used to depict the chromosomes. GA begins the

search for the most superior solution by employing the selection and recombination operators. According to fitness function, the best solution is converged to and used for further reproduction. The fitness of a chromosome is determined using the Mean Squared Error Method and optimal solution is the subset with lowest Mean Squared Error. If the current solution of the reproduction phase satisfies the end condition, then the solution is considered final and most optimal. Otherwise, the entire process of GA repeats itself.

Recurrent Neural Networks have a feedback loop that acts as the memory component. However, LSTM can maintain the memory for a longer time. This makes LSTM more suited for linguistic applications where memory plays a huge role. However, in an LSTM network, the window size and number of LSTM units play a huge role in accuracy and are difficult to decide. Character level CNNs have the unusual ability to use word embedding and derive features from it and hence were very successful in linguistic applications. However, recent approaches treat pre-trained embeddings as fixed parameters which reduces their effectiveness. Bag of words models usually is used in applications with high recall. But, they also report a significant number of false positives which limits their usability. Bi-LSTMs prove to be very useful and instrumental in fixed sequence to sequence kinds of applications. However, they fail to give satisfactory results in cases where input and output sizes differ greatly. Figure 1 depicts the flowchart of the proposed architecture.

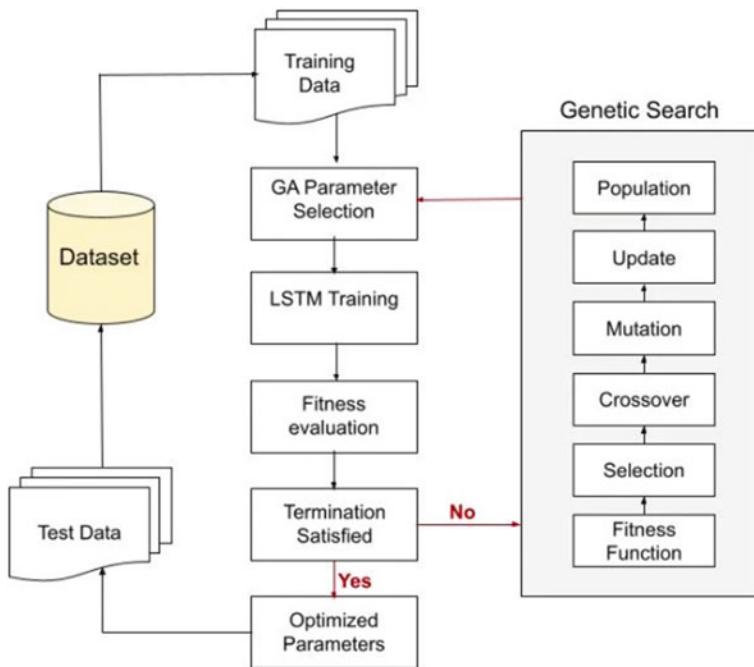


Fig. 1 Flowchart of proposed architecture

5 Experimental Work

5.1 Dataset

The dataset is built from Coronavirus (COVID-19) tweets ranging over 14 months starting from March 2020. It contains twitter id and their corresponding twitter score. The original tweets were collected using Twitter APIs. A total of 11 csv files were created, each consisting of tweets and their sentiment, with every csv file having data of a particular month. The sentiment of any tweet is categorized as positive, negative, and neutral (Table 1).

5.2 Results

This paper proposes a Social Media Pandemic Sentiment Model (SMPSTM). Figure 2 shows the month-wise time series graph of COVID-19 tweets and classifies the tweets into positive, neural, and negative. We evaluate and compare this model with already existing competitive state-of-the-art models such as RNN, LSTM, Character-level CNN, Bi-LSTM, Hierarchical Attention Networks, and Attentional Bi-LSTM. The performance parameters taken for comparison are accuracy, precision, recall, and

Table 1 Keywords used in tweets

#corona	Social distancing	#coronavirus	Wear mask
#covid19	#covididiots	#staysafe	Herd immunity
Pandemic	Vaccine	Quarantine	Face shields
Covid-19	#healthworkers	#lockdown	#coronawarriors
Hand sanitizer	#stayathome	#workfromhome	Self isolating

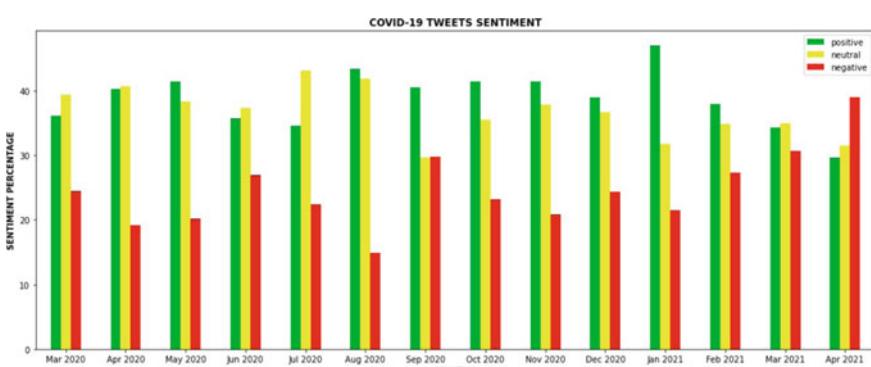


Fig. 2 Time series graph of COVID-19 tweets sentiment

Table 2 Value of parameters used by GA

Parameter	Value
Gene length	10
Crossover rate	0.4
Mutation rate	0.1
Number of generations	4
Population size	4

Table 3 Performance comparison of proposed model with different algorithms

Architecture	Accuracy	Precision	Recall	F1-score
RNN	0.82	0.76	0.85	0.80
LSTM	0.85	0.75	0.87	0.81
Character-level CNN	0.84	0.70	0.81	0.75
Bi-LSTM	0.83	0.72	0.86	0.78
Hierarchical attention net	0.85	0.78	0.82	0.80
Attentional Bi-LSTM	0.84	0.77	0.88	0.82
SMPSM	0.88	0.80	0.85	0.83

F1-score. All the models have been trained and tested on the same dataset to get a wholesome comparison. The parameters used by GA such as gene length, crossover rate, mutation rate, number of generations, and population size are very crucial and their values are shown in Table 2. After investigation by GA, the best window size turns out to be 36 and the optimal number of LSTM units is 12. The final model is trained for 20 epochs with a batch size of 32 and dropout to be 0.2. Table 3 presents the performance of different classifiers on the given dataset to give a competitive and wholesome analysis by comparing them on measures, namely, accuracy, precision, recall, and F1-score.

6 Conclusion

This paper proposed SMPSM as a deep learning-based model that outperforms all the baseline models like RNN, LSTM, etc in sentiment analysis of COVID-19 dataset. The Attentional Bi-LSTM gives higher recall values owing to its capability to capture long term dependencies better. Character Level-CNNs sometimes undergo higher perplexity due to the nature of prediction as a result of which they may produce unusual words. For future work, we would like to implement our model on the vast dataset of COVID-19 that includes tweets of whole year 2021.

References

1. Aljameel SS, Alabbad DA, Alzahrani NA, Alqarni SM, Alamoudi FA, Babil LM, Aljaafary SK, Alshamrani FM (2021) A sentiment analysis approach to predict an individual's awareness of the precautionary procedures to prevent COVID-19 outbreaks in Saudi Arabia. *Int J Environ Res Public Health* 18(1):218
2. Arora M, Kansal V (2019) Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Soc Netw Anal Mining* 9(1):1–14
3. Chakraborty K, Bhatia S, Bhattacharyya S, Platos J, Bag R, Hassanien AE (2020) Sentiment analysis of COVID-19 tweets by deep learning classifiers-a study to show how popularity is affecting accuracy in social media. *Appl Soft Comput* 97, 106754
4. Rani P, Bhatia M, Tayal D (2019) A comparative study of qualitative and quantitative SNA. In: 2019 6th International conference on computing for sustainable global development (INDIA-Com). IEEE, pp 500–504
5. Rani P, Shokeen J, Singh A, Kumar S, Raguvanshi N (2021) Stock price prediction using reinforcement learning. In: International conference on innovative computing and communication (ICICC-2021). Springer
6. Shokeen J, Rana C (2019) An application-oriented review of deep learning in recommender systems. *Int J Intell Syst Appl* 10(5):46
7. Shokeen J, Rana C (2019) Social recommender systems: techniques, domains, metrics, datasets and future scope. *J Intell Inf Syst* 1–35
8. Singh M, Jakhar AK, Pandey S (2021) Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Soc Netw Anal Mining* 11(1):1–11
9. Wang JH, Liu TW, Luo X, Wang L (2018) An LSTM approach to short text sentiment classification with word embeddings. In: Proceedings of the 30th conference on computational linguistics and speech processing (ROCLING 2018), pp 214–223
10. Wang M, Zhu Y, Liu S, Song C, Wang Z, Wang P, Qin X (2019) Sentiment analysis based on attention mechanisms and Bi-Directional LSTM fusion model. In: 2019 IEEE SmartWorld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation. IEEE, pp 865–868
11. Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* 7:51522–51532
12. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489

Machine Learning Techniques for Keystroke Dynamics



Kirty Shekhawat and Devershi Pallavi Bhatt

Abstract Conventional security mechanisms such as token-based and knowledge-based authentication mechanisms are losing importance in the present era of immense technological development in cyber threats. Password and pin are examples of these mechanisms. Keystroke biometrics is a promising solution for ensuring cybersecurity in both standalone and connected systems. Keystroke biometrics is a subset of behavioral biometrics and distinguishes users based on their typing patterns. The performance of a user authentication system utilizing keystroke biometrics depends on the extracted features and classification techniques. The objective of this paper is to compare three different learning techniques namely support vector machine, random forest and logistic regression, in the context of keystroke biometrics. Time-based features are extracted from a publicly available dataset. These features are analyzed with above mentioned machine learning algorithms, and the performance of these algorithms is compared. Hyperparameter tuning and cross-validation are performed to further enhance the performance. Experimental results demonstrate that Random forest is the most efficient with accuracy of 0.85 and F1 score of 0.74. The accuracy obtained with support vector machine and logistic regression is 0.76 and 0.63, respectively.

Keywords Biometrics · Keystroke dynamics · Machine learning · User authentication · Security

1 Introduction

The world is undergoing digital transformation. Data storage and utilization have become an inseparable part of daily activities. A lot of confidential and sensitive information is shared on the digital platforms by the users. Static usernames and passwords are the most common security mechanism used by organizations to provide confidentiality and security to consumers. Passwords are difficult to manage

K. Shekhawat · D. P. Bhatt (✉)

Department of Computer Application, Manipal University Jaipur, Jaipur, Rajasthan, India

and remember [1]. Reports of data breach and stolen passwords are increasing and hackers are coming up with advanced statistical techniques. A robust user authentication system that is not easily hackable else need of the hour. Behavioral biometrics utilizes certain unique features of the individual behavior which cannot be imitated easily by another human or machine. Keystroke dynamics is one such approach of behavioral biometrics [2].

Typing biometrics, keystroke analysis, and keystroke biometrics are different synonyms used to refer feast of dynamics in the literature. The concept of keystroke dynamics was first proposed during World War II because of emerging security concerns [3]. Operator's word trained to identify the cipher sender based on key rhythms. Since then keystroke dynamics has witness adaptations of different algorithms and techniques dynamics for the user is monitored on the basis of typing characteristics [4]. The typing characteristics refer to features such as pressure, key hold and release time, typing speed [5]. Keystroke dynamics-based user authentication systems can be static or continuous in nature [6]. As the name implies in the static keystroke dynamics system, the typing behavior of the user is analyzed only once, mostly at the time of login. In a continuous verification keystroke dynamic system, the typing behavior of the user is analyzed at different checkpoints during the entire session. The advantages of keystroke dynamics that distinguish it from other behavioral biometric techniques are non-intrusive nature, no requirement of user training transparent and inexpensive requirements (Fig. 1).

Research proffers that in a keystroke dynamics-based authentication system (KDA) 'what is typed' is not important rather attention is given to ' how it is typed'. Each person has different typing behavior due to difference in physiological features. KDA may be used with features extracted from a keyboard or touchscreen [7]. Over the past two years, researchers have given much attention to data acquisition and feature representation methods. Different publicly available datasets in the given context can be found in [8]. The use of machine learning (ML). KDA for data classification is a comparatively new field. The aim of this paper is to access the classification efficiency of three popular machine learning algorithm in the domain of user authentication system based on keystroke dynamics. Rest of the paper is organized as follows: Sect. 2 demonstrates literature review in brief. The proposed

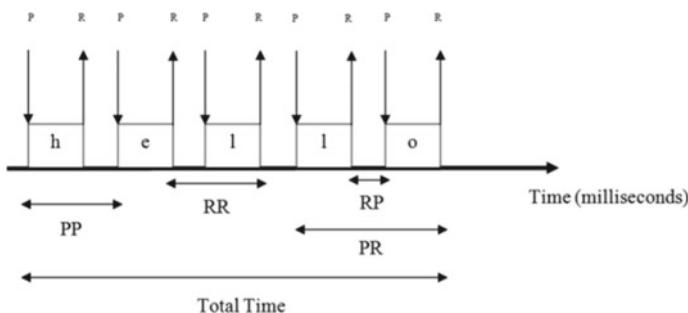


Fig. 1 Dwell time and flight time features [6]

methodology is illustrated in Sect. 3. Experimental results are discussed in Sect. 4. Conclusion and recommendations for future work are mentioned in Sect. 5.

2 Related Work

The most important aspect of a user authentication system based on keystroke dynamics is creating a unique user profile for every individual. It involves analysis of a large amount of data and consequently developing user profiles for thousands of users, while ensuring that each user profile is unique and differentiable. The easiest way to perform classification in keystroke dynamics is by using simple distance metric for such as Euclidian distance and Manhattan distance. The drawback of these approaches is the reduced performance and inability distinguish between large number of users is this technique cannot identify the correlation between features present in a data set.

Machine learning is the best technique for classification in such systems.

Different machine learning algorithms are applied in this context such as linear classifiers like Logistic classifier and Naive Bayes algorithm, support vector machine (SVM), decision trees, random forest, neural network, and nearest neighbor. The neural network is the most widely used classifier [9, 10]. Researchers preferred neural networks because of their ability to find inherent patterns even in highly complex and noisy data. The authors in [11] have used NN for predicting missing digraphs. The experimental results were FAR equal to 0.0152% and FRR equal to 4.82%. The computational cost of neural network deployment is a challenge along with cases of overfitting and underfitting. Deng and Zhong [12] applied the deep learning method to keystroke dynamics user authentication. Their study shows deep learning method significantly outperforms other algorithms on the CMU keystroke dynamics dataset. KDA is tested by using different classification algorithms for instance Genetic Algorithm [13], Gaussian Mixture Model (GMM) [14], and firefly algorithm [15].

The hidden Markov model can be used to establish a relation between noise and extracted features hidden Markov model is used in keystroke dynamics as it consists of hidden variables that have a conditional or generated dependency [16]. Unsupervised machine learning techniques such as KNN are also used in keystroke dynamics [17]. It can be seen that the performance of supervised learning techniques is better than unsupervised learning techniques.

Support vector machine reduces the generalization error by maximizing boundary, while ensuring processing choice limits. Support vector machine is used as classifiers in keystroke dynamics. Giot et al. [18] used SVM to reduce the number of entries required during the registration step, while maintaining performance of EER value 15.28%. From the reviewed literature, it can be concluded that there exists very few research work that have compared performance of different ML methods on the same data. In order to find out the best ML approach for a user authentication system based

on KD (keystroke dynamics), it is essential to test different approaches on the same dataset. The aim of this paper is to incur the same by experimental observation.

3 Methodology

The adopted methodology can be divided into four blocks as shown in Fig. 2. These process blocks are discussed in detail in subsequent sections.

3.1 Database Acquisition and Normalization

In this project, the publicly available data set of ‘keystroke dynamics challenge’ has been used [19]. The data set has been collected by gathering information from 110 test subjects. Each subject was asked to type the fixed text, ‘UNITED STATES,’ 13 times. Finally, the data was compiled in columns in which each column represents some timing information.

Data normalization refers to applying statistical techniques to data to improve its quality which leads to better classification. Data normalization is important to remove anomalies like redundancy and inconsistency. In this research, the acquired data were firstly checked for garbage or null values. Such entries were removed. The filter database was then normalized using an equalization histogram. Now, the data is ready for feature extraction and machine learning phase.

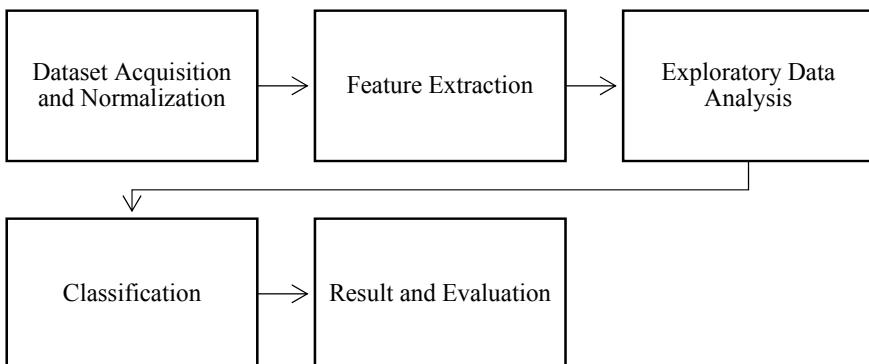


Fig. 2 Block diagram of proposed methodology

3.2 Feature Extraction

Diagraph features expected from the timing information available about consecutive key types and latency. The action of pressing two keys consequently leads to development numerous time features. The four basic time-based features used in keystroke dynamics are given below. 155 features were extracted for each user.

- Dwell Time/hold duration (hd): The time duration for which a particular key is pressed is called Dwell Time.
- Flight Time or Up-Down Time (udd): It refers to the time elapsed between releasing first key (key up event) and pressing second key (key down).
- Down-Down Time (ppd): The time elapsed between key down event of first and second key is called Down-Down Time.
- Up-Up Time (rrd): The time interval between key up event of first and second key is called Up-Up time (UU).
- Down-Up Time (prd): The time elapsed between down event of first key and up event of next key is called Down-Up Time (DU).

3.3 Exploratory Data Analysis

Exploratory data analysis refers to plotting swarm plots to visualize data and extract some information from it and assign weights to features. In Fig. 3, x axis represents user number, and y axis represents these time features values, Down-Down Time (ppd), Up-Down Time (udd), and hold duration (hd). Through it can be observed that the hold duration behavior is different from other time duration features. The next step is creating increasing order sequence of all time duration for all users and for all character written in signature, for example hold time for all different user for different keys is in incremental order. Same for other time duration features also. Taking exact value of time will not provide much information so for this therefore data is in bins (Encoded form), for example we take hold time duration data separate it into 10 bins (0–9) Fig. 4 shows the resultant swarm bins. It can be seen that for the given dataset hold duration shows maximum variation for different users and thus must be given greater importance.

3.4 Classification

The next step is to classify the extracted time features by using machine learning algorithm. Machine learning papers to the group of specialized algorithms developed for pattern recognition, categorization, and prediction. Machine learning techniques are highly efficient, and it can identify trends from highly dimensional databases. In this research work, three algorithms of machine learning are used namely support vector

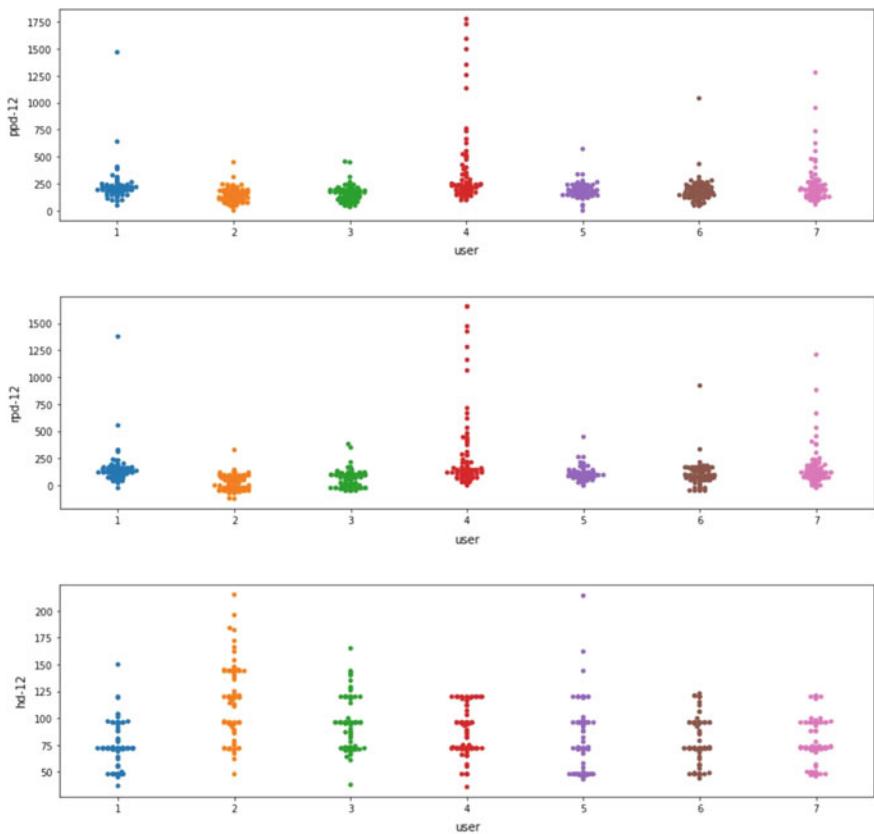


Fig. 3 Swarm plots for exploratory data analysis

machine, random forest, and logistic regression for the purpose of classification. In contrast to several machine learning algorithms, support vector machine works on small sample learning methods. The underlying principle of SVM is structural risk minimization. SVM is based on linear regression and assigns training samples one among the N categories. In other words, the training samples are grouped in N number of classes, which separated by strict boundaries. Figure 5 demonstrates a bisecting SVM hyperplane that divides the input into two categories (H_1 and H_2). The crucial task in SBI Wills identification of hyperplane and its derivative line coefficient. The optimal hyperplane b_i set the points the representing largest separation among two categories. The distance between the two categories is called a class interval.

Traditionally, logistic regression was used as a statistical method for binary classification. Over the years, logistic regression has been adopted as a machine learning model. It is different from linear regression as sinusoidal curve is used for separating

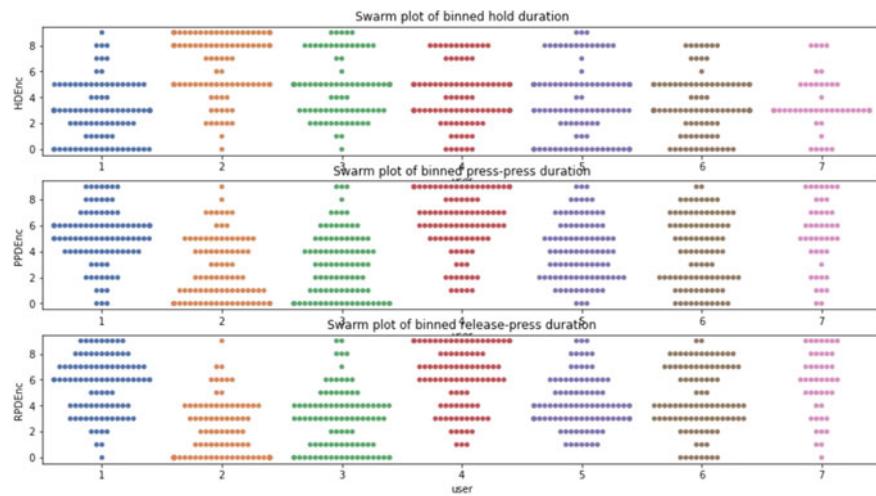


Fig. 4 Swarm bins plots

Fig. 5 Optimal separating surface in SVM [20]

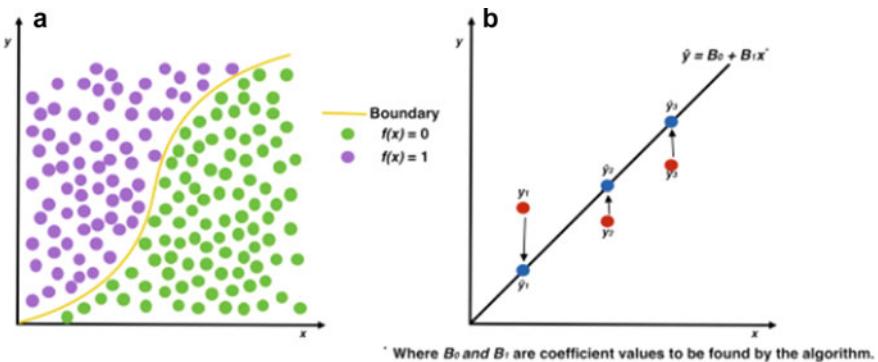
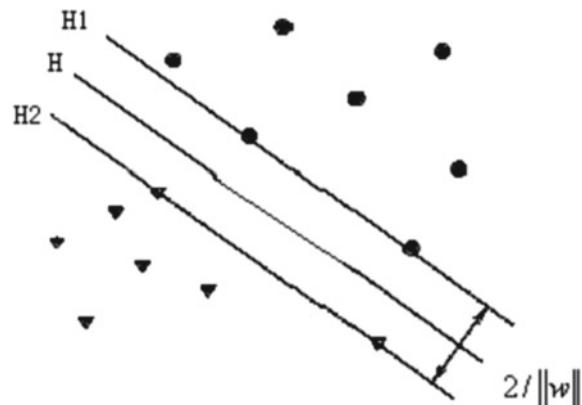


Fig. 6 Illustration of logistic regression [21]

different categories. Logistic regression involves waited transformation of the categorical data points. In Fig. 6, the logistic regression functions categorize inputs into two categories based on the coordinates.

Random forests are sometimes referred to as random decision forests. It is a form of N symbol learning methodology. Random forests use a multitude of decision trees and training stages for classification and regression. The output class is either mode or mean predicted value of individual decision trees. An individual decision tree is constructed by selecting a random sample from the available data. These decision trees are also called random subspaces. The concept of bagging is used for random sampling of training data. The prediction power of random forests is very high as the correlation between the individual trees is reduced by randomly selecting features.

4 Result and Discussion

The illustration of scatter plots (release press time vs press- press time) is shown in Fig. 7. The plot signifies that a proportional relationship exists between Up Down and Down time. The histograms of time features of a particular users, while typing the fixed text are shown in Fig. 8. It can be concluded that hold duration shows more variance than other features and that the features follow an approximate bell-shaped distribution. The training parameters are mentioned in Table 1.

Performance parameters obtained after hyperparameter tuning are demonstrated in Table 2. The performance measures used in this research are accuracy, precision, recall, and $F1$ score. From Table 1, it can be observed that Random forest gives the best results.

Accuracy denotes the number of correctly identified users. Random forest has the highest accuracy of 85%, followed by SVM. However, in a user authentication system, other performance parameter are also important as cost of false positive (identifying intruder as genuine user) can be high.

Precision is the measure of total number of correctly identified users out of total number of users identified or labeled as correct by the system. Recall is the measure of total number of correctly identified users out of total number of genuine users in

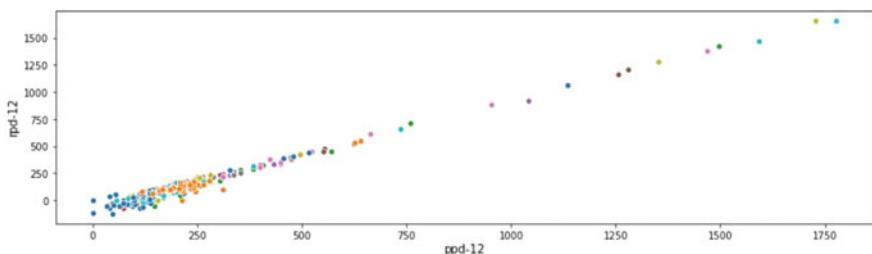
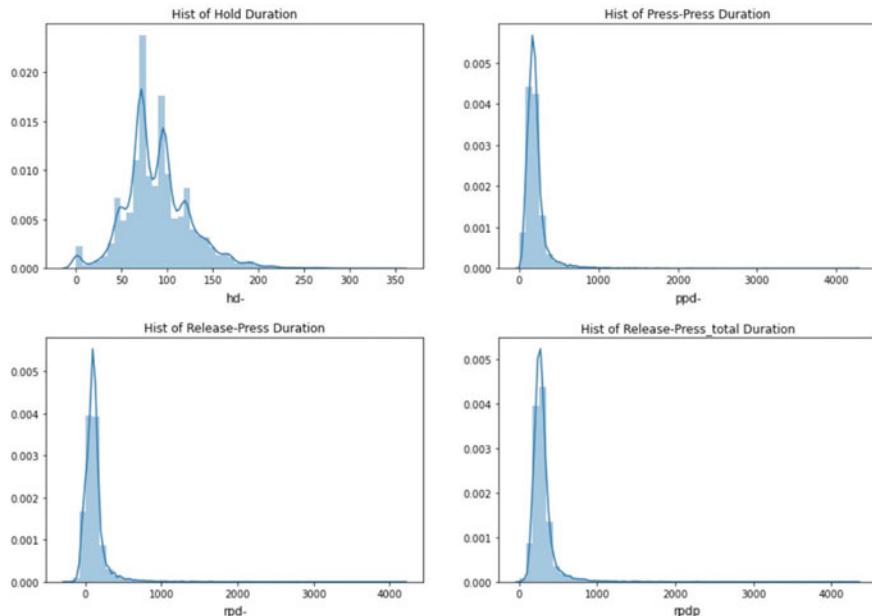


Fig. 7 Illustration of logistic regression

**Fig. 8** Histogram of time features**Table 1** Accuracy after hyperparameter tuning

Model	Training parameters
SVM	C = 1, kernel = linear
Random forest	Criterion = 'ini', max_depth = 450, max_features = 'log2', min_samples_leaf = 1, min_samples_split = 4, n_estimators = 300
Logistic regression	C = 1

Table 2 Performance matrix after hyperparameter tuning

Model	Accuracy	Precision	Recall	F1 score
SVM	0.76	0.67	0.69	0.72
Random forest	0.85	0.67	0.72	0.74
Logistic regression	0.63	0.67	0.62	0.71

database. *F1* score is harmonic mean of precision and recall. *F1* score is used when cost of false negative is high. It can be observed that Random forest shows superior performance, followed by SVM. Logistic regression has the weakest performance matrix.

5 Conclusion

Due to ever increasing number of cyber attacks, keystroke dynamics is being utilized for developing strong user authentication systems. In keystroke, dynamics developing a unique and consistent user profile is a challenging task as the value of time-based features varies with the emotional state of the user. Sophisticated feature extraction and ML based classification techniques can be used for enhancing user uniqueness and consistency. In this research SVM, logistic regression and random forest were applied on a publicly available dataset for comparative analysis. The results reveal that random forest demonstrates the best performance with accuracy close to 85% and *F1* score of 0.74. In future, authors would like to extend the model by incorporating artificial cues or sensor data in the analysis.

References

1. Jaccard J, Nepal S (2014) A survey of emerging threats in cyber security. *J Comput Syst Sci* 80(5):973–993
2. Monroe F, Rubin A (2000) Keystroke dynamics as a biometric for authentication. *Futur Gener Comput Syst* 16:351–359
3. Yadav A (2018) Comparison of learning models in behavioural biometrics using keystroke dynamics. *Int J Comput Intell Res* 14:1061–1067
4. Raul N, Shankarmani R, Joshi P (2020) A comprehensive review of keystroke dynamics-based authentication mechanism. international conference on innovative computing and communications. *Adv Intell Syst Comput* 1059
5. Umphress D, Williams G (1985) Identity verification through keyboard characteristics. *Int J Man Mach Stud* 23(3):263–273
6. El Menshawy D, Mokhtar HMO, Hegazy O (2014) A keystroke dynamics based approach for continuous authentication. In: Kozielski S, Mrozek D, Kasprowski P, Małysiak-Mrozek B, Kostrzewska D (eds) Beyond databases, architectures, and structures. BDAS 2014. Communications in computer and information science, vol 424
7. Lamiche I, Bin G, Jing Y et al (2019) A continuous smartphone authentication method based on gait patterns and keystroke dynamics. *J Ambient Intell Hum Comput* 10:4417–4430
8. Giota R, Dorizzib B, Rosenberger C (2015) A review on the public benchmark databases for static keystroke dynamics. *Comput Soc* 55(46):61
9. Gedikli AM, Efe MÖ (2019) A simple authentication method with multilayer feedforward neural network using keystroke dynamics. In: Djeddi C, Jamil A, Siddiqi I (eds) Pattern recognition and artificial intelligence. MedPRAI 2019. Communications in computer and information science, vol 1144. Springer, Cham. https://doi.org/10.1007/978-3-030-37548-5_2
10. Shanmugapriya V, Padmavathi G (2010) Keystroke dynamics authentication using neural network approaches. In: Das V.V., Vijaykumar R. (eds) Information and communication technologies. ICT 2010. Communications in computer and information science, vol 101. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15766-0_121.
11. Ahmed A, Issa T (2013) Biometric recognition based on free-text keystroke dynamics. *IEEE Trans Cybern* 44. <https://doi.org/10.1109/TCYB.2013.2257745>
12. Deng Y, Zhong Y (2013) Keystroke dynamics user authentication based on Gaussian mixture model and deep belief nets. In: ISRN signal processing. ArticleID 565183
13. Rodrigues R, Yared G (2005) Biometric access control through numerical keyboards based on keystroke dynamics. In: Zhang D, Jain A (eds) Advances in biometrics, lecture notes in computer science, vol 3832, pp 640–646

14. Hosseinzadeh D, Krishnan S (2008) Gaussian mixture modeling of keystroke patterns for biometric applications. *IEEE Trans Syst Man Cybernetics Part C: Appl Rev.* 38(6):816–826
15. Muthuramalingam A, Gnanamanickam J, Muhammad R (2018) Optimum feature selection using firefly algorithm for keystroke dynamics. In: Abraham A, Muhuri P, Muda A, Gandhi N (eds) Intelligent systems design and applications. ISDA 2017. Advances in intelligent systems and computing, vol 736
16. Dwivedi C, Kalra D, Naidu D, Aggarwal S (2018) Keystroke dynamics based biometric authentication: a hybrid classifier approach. In: 2018 IEEE symposium series on computational intelligence (SSCI), Bangalore, India, pp 266–273
17. Cho S, Hwang S (2005) Artificial rhythms and cues for keystroke dynamics based authentication. *Adv Biomet* 3832
18. Giot R, El-Abed. M, Rosenberger C (2009) Keystroke dynamics with low constraints SVM based passphrase enrollment. In: IEEE international conference on biometrics: theory, applications and systems (BTAS 2009), pp 1–6. <https://doi.org/10.1109/BTAS.2009.5339028>
19. Kaggle: Keystroke dynamics challenge 1. <https://www.kaggle.com/c/keystroke-dynamics-challenge-1/overview/description>
20. Zhang Y (2012) Support vector machine classification algorithm and its application. In: Liu C, Wang L, Yang A (eds) Information computing and applications. ICICA 2012. Communications in computer and information science, vol 308. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34041-3_27
21. Panesar S, D’Souza R, Yeh F, Fernandez. J (2019) Machine learning versus logistic regression methods for 2-year mortality prognostication in a small, heterogeneous glioma database. *World Neurosurg* 2. <https://doi.org/10.1016/j.wnsx.2019.100012>.

Detection of Denial-of-Service Attacks Using Stacked LSTM Networks



Deepa Krishnan

Abstract In spite of the usage of improved detection mechanisms, denial-of-service attacks (DoSs) and its variants continue to ravage the computing world. There are quite a few significant works that have demonstrated the efficiency of using machine learning and deep learning algorithms in detecting the attacks. Our proposed work focuses on the effect of number of layers, learning rate and momentum on the accuracy of LSTM CNN networks in detecting the different categories of DoS attacks. We have used CICIDS 2017 attack dataset and in particular the dataset involving security attacks using DoS attack tools such as GoldenEye, Slow HTTP test, Slow Loris and Hulk. The proposed research work analyzes the effect of change of hyper-parameters on various evaluation parameters like accuracy, precision, F1 score, recall, RoC score and Kappa score and has observed that LSTM with 1 layer and learning rate = 0.1 and LSTM with 2 layers and learning rate = 0.2 has given better performance measures.

Keywords Denial-of-service attacks · Deep learning · Detection · Long short-term memory · LSTM · Machine learning

1 Introduction

The last few decades have seen the growing prominence of computing infrastructure in everyday life of common people, major industrial and corporate houses, and all government offices. With increased dependence on technology in today's connected world, sensitive data of individuals and business houses and strategic data of national significance end up in servers worldwide. This leads to growing importance of maintaining the security of these servers and defending the cyber-attacks targeting them. Global spending on cybersecurity research and products is expected to exceed \$1 trillion over the five-year period from 2017 to 2021. During the time of Covid-19 pandemic, many organizations around the world depend on cloud-based services to

D. Krishnan (✉)

Department of Computer Engineering, NMIMS University (Deemed-to-be), Mumbai,
Maharashtra, India

e-mail: deepa.krishnan@nmims.edu

fuel remote workforce, spending on cybersecurity is expected to increase further. In research work by Lallie et al. [1], the authors have elaborated on the increase in number of cyber-attacks all over the world during the time of pandemic. They have illustrated with timeline of events the major cyber-attacks that have happened throughout the world and particularly in UK. The authors also bring a loose correlation between events and cyber-attacks reported and they have envisioned a predictive model for this.

Detection of cybersecurity attacks well in advance has always been at the forefront of research. There are broadly two types of detection approaches for security attacks: signature-based detection and anomaly-based detection. Signature-based techniques depend on pre-defined signatures; however, anomaly-based techniques monitor deviation from baseline features that are already established [2]. Signature-based techniques are preferred in situation where quick decision in detection is preferred and anomaly-based detection is beneficial where it is important to detect new unseen attacks. However, these systems are prone to high false alarms as it will trigger alarms on deviations from the baseline. In this context, many significant research works based on machine learning are gaining prominence. An extensive review of techniques based on machine learning for intrusion detection is done by researchers in [3]. The authors have detailed the supervised classical machine algorithms that could be used in detecting attacks with their potential benefits and challenges. They have demonstrated the efficiency of machine learning algorithms using various performance measures like accuracy, precision, recall and F1 Score. In recent years deep learning algorithms have gained attention of researchers and have demonstrated its usefulness in the field of object recognition and classification due to its potential to learn from the hierarchy of features. Many research works have compared the effectiveness of deep learning-based algorithms in detecting attacks over traditional machine learning algorithms. The role of deep learning in intrusion detection is investigated in detail by authors in [4]. They have described various deep learning methods like autoencoders, DBN, RNN, LSTM and GRU-based methods and have compared their performance in detection rate and accuracy. Some of the research works used deep learning for feature learning in pre-training and other algorithms for classification. Other group of research works used deep learning with either shallow or deep layers as a classifier.

In the proposed work, we have used LSTM (long short-term memory) based deep neural networks [5] that have shown significant improvements over convolutional neural networks. We have used the CICIDS 2017 dataset which contain normal traffic and attack traffic that resembles the true real-world data for experimental study. The major contributions and scope of the proposed research are as follows:

- We have developed a deep learning model using Deep LSTM that can detect and classify the attacks into multiple categories: GoldenEye, Slow HTTP Test, Slow Loris and Hulk with performance measures for each class of attacks.
- We have conducted experimental analysis to investigate the number of layers on the performance measures of the classification along with identifying the best

combination of hyper-parameters. The effect of change of hyper-parameters on the performance of the model are studied and analyzed extensively.

We have organized the rest of the paper as follows; in Sect. 2, we have described the related work, and in Sect. 3, we have described our proposed work. Algorithm and methodology and experimental setup are explained in Sect. 3. The result and analysis are explained in detail in Sect. 4, and our conclusion and future scope are in Sect. 5.

2 Related Work

We have reviewed various research works that have used deep learning algorithms for intrusion detection. One of the important works in intrusion detection using convolutional neural networks is done by authors Kim et al. [6]. The authors have done experiments on CICIDS 2018 dataset and have used two convolutional layers and two max pooling layers behind each convolutional layer. In addition, the authors have used ReLU as an activation function for each convolutional layer and drop out is used after each max pooling layer. The experiments indicate the accuracy of CNN is more than that of Recurrent Neural Networks. However, for some of the classes like SQL injection and brute force web even after preprocessing, the accuracy is less than 70%. Another important research work that has worked on the CICIDS 2017 dataset is by Hongpo Zhang et al. in which the authors have proposed convolutional neural network based on SMOTE and Gaussian mixture model [7]. In the proposed work the authors have considered one benign class and 14 attack classes with a highly imbalanced dataset of normal traffic accounting for 80.30% and attack traffic of 19.70%. The authors have used SMOTE oversampling technique [8] along with undersampling with clustering using Gaussian mixture model. The authors have varied the number of convolution layers in each layer and they have got the best performance when number of kernels in four convolution layers are 32, 32, 64 and 64 and optimizer is nadam. When SMOTE and GMM is used for imbalanced treatment, it gave a detection rate improvement almost reaching 1 for all classes except for Web attack Brute Force, Web Attack XSS and Infiltration where it is ranging from 0.50 to 0.82.

Adaboost methods and synthetic minority oversampling technique (SMOTE) are used on the CICIDS dataset by Yulianto et al. [9]. In this work, SMOTE technique is used for the class imbalanced problem and principal component analysis and ensemble feature selection is used for feature reduction. The experimental analysis indicates that the Adaboost classifier using PCA and SMOTE gives an AUCROC score of 92% and the Adaboost classifier with EFS and SMOTE gave an accuracy, precision, recall and F1 Score of 81.83%, 81.83%, 100% and 90.01%, respectively. OneR and REPTree have been used along with Correlation Feature Selection on the CICIDS dataset by researchers in [10]. REPTree gave accuracy of about 99% for Brute Force Attack, PortScan, DDoS attack, Botnet attack and Infiltration Attack.

However, the specificity for the infiltration attack is found to be only around 63%. An extensive study on the performance of deep discriminative models is done by authors [11] in which the detection rate for Slow HTTP test and GoldenEye are found to be 94.5 and 92.1% with deep neural networks. However, CNN is found to give better accuracy over DNN in the case of Slow HTTP test and GoldenEye. But in the case of DoS Attacks Hulk, both DNN and CNN are found to give an accuracy of only 93.33 and 94.012%.

In [12], the authors have used LSTM-RNN and train the model on KDD 99 dataset and have done extensive experiments on varying hidden layer size and hyper-parameters to confirm the false alarm rate and detection rate. The authors have found highest efficiency for hidden layer size = 80 and learning rate of 0.01. The performance of LSTM-RNN is compared with other algorithms in terms of detection rate, accuracy and false acceptance rate. In comparison with other algorithms like KNN, SVM, Bayesian, radial basis neural network, LSTM-RNN is proven to be better with an accuracy of 96.93%. The average percentage of attack detection is compared for each category of attacks. The DoS attacks are detected on average 97.81%, probe attacks 54.714% and R2L 57.83%. However, U2R category of attacks was not detected by the proposed algorithm possibly due to the less number of samples. Another important research work that has used CICIDS dataset for deep learning-based intrusion detection is by authors Fernández and Xu [13]. In this work, the authors have used autoencoders and have observed higher reconstruction error for malicious flows. However, this approach shows a very high false-negative rate of 0.760. The authors have tried to study the effect of IP address feature on deep learning algorithms as it is a dynamic feature. They propose to investigate in future whether or not using the whether or not using the IP address and port no features can contribute to reduce the false-negative rate.

Our work stands apart from the literature in that we have minimal feature processing and a very simplified LSTM architecture that renders very high performance measures.

3 Proposed Work

3.1 *Dataset Description*

This study uses the CICIDS2017 dataset published by Canadian Institute of Cyber-security [14]. This includes benign and most up-to-date and complex real-world attacks unlike the other datasets. The other datasets for intrusion detection are found to be lacking in diversity and volume of traffic. The authors have prepared many attack profiles matching with various real-world attacks. The focus of our work is one of the most prominent categories of attacks which is the denial-of-service attacks. Hence, we have used the WednesdayWorkingHours.pcap_ISCX.csv file which includes the DoS and heartbleed attacks, SlowLoris, Slowhttptest, Hulk and

Table 1 Category of attacks in the Wednesday working hours

Category	Number of samples	Percentage of each category (%)
Benign	440,031	63.5
DoS Hulk	231,073	33.4
DoS GoldenEye	10,293	1.5
DoS Slow Loris	5796	0.8
DoS SlowHTTPtest	5499	0.8
Heartbleed	11	0.002

Table 2 Dataset statistics in the Wednesday working hours

Category	Training set	Testing set
Benign	351,921	88,110
DoS Hulk	184,948	46,125
DoS GoldenEye	8266	2027
DoS Slow Loris	4642	1154
DoS SlowHTTPtest	4379	1120
Heartbleed	6	5

GoldenEye category of attacks. The distribution of each category of attacks in the WednesdayWorkingHours.csv dataset is as follows in Table 1

We have used 80% of the dataset for training and 20% for testing. The training and testing distribution of each category of attacks is given in Table 2.

3.2 Data Preprocessing

The attack categories in the dataset are Benign, DoS GoldenEye, DoS Hulk, DoS SlowLoris, DoS SlowHTTPtest and heartbleed. These are label encoded as 0, 1, 2, 3, 4 and 5, respectively [15]. We have undertaken feature scaling as there are many features like Total Backward packets, Fwd Packet Length, Max Fwd Packet Length, Min Fwd Packet Length, MeanFwd Packet Length, Std Bwd Packet Length where the minimum and maximum values range between 0 and a very high value. This difference between magnitude of feature values affects the efficient working of optimization algorithms like gradient-based algorithms. In the proposed work, we have used standard scaler that will center the values around the mean with a unit standard deviation [16]. Every feature columns are scaled using the following equation.

$$X_{std} = (X - \text{Mean}(X)) / \text{Standard Deviation}(X) \quad (1)$$

3.3 Algorithm and Methodology

3.3.1 LSTM Algorithm

Long short-term memory (LSTM) is a class of Recurrent Neural Networks that are capable to overcome the short comings of RNN. RNNs are used for modeling time series data and a hidden unit takes as input the output from the previous hidden unit and the current input. However, RNNs are not so effective in learning lengthy time period dependencies. In this case, LSTMs are proven to efficient for remembering facts for a long interval of time [17].

Each LSTM module can have three gates: forget gate, input gate, output gate.

- Forget Gate: This gate makes a decision about what need to be discarded in that instance of time. This is done using the Sigmoid function. The information from the current input $X(t)$ and previously hidden state $h(t - 1)$ are given to the sigmoid function. It generates values between 0 and 1 to conclude which part of previous input is necessary.
- Input gate: The sigmoid function makes a decision which values to permit through 0, 1, and Tanh function gives weightage to the values which might be calculated based on degree of level of importance ranging from -1 to 1 .
- Output Gate: The output gate determines the value of the next hidden state based on information on previous inputs. The values of the current state and previous hidden state are fed to the next sigmoid function. Then the new state generated is fed through the \tanh function and output from these functions are multiplied point-by-point. Finally, the network decides which information the hidden state based on the final value and this final value is used for prediction.

3.3.2 Experimental Setup and Implementation

We have performed the experiments on Google Colab environment with hardware accelerator as TPU. LSTM implementation is done using Keras, a deep learning framework available in Python. The Keras Sequential Model is used to build the linear stack of layers. The experiments are done on a LSTM network with 3 layers: input layer, hidden layer and an output layer. The input layer is a dense layer with 50 neurons and the input dimension is 78 which is the number of features. The number of hidden layers is varied from 1 to 4 and the units in input layer and hidden layer are fully connected. The output layer contains 6 neurons that helps in categorizing the attacks to corresponding 6 categories: benign, DoS Hulk, DoS GoldenEye, DoS SlowLoris, DoS Slowhttptest and heartbleed.

The accuracy and rate of detecting attacks depends significantly on the parameters used in deep networks like LSTM. The values of the hyper-parameters of the model that are used in our experimental study are as follows:

The training set and testing set NumPy array is reshaped into (554,162,178) and (138,541,178) to be fed into the LSTM network. While conducting the experiments,

the activation function used in the output layer of the neural networks is SoftMax which helps in predicting a multinomial probability distribution. As there are multiple categories of classes and each class category is label encoded, the loss function used is Sparse Categorical Cross-Entropy.

4 Results and Discussion

The entire dataset divided into training and testing dataset after reshaping are fed to the LSTM network as per the configuration mentioned in Sect. 3. The hyper-parameters are adjusted, and the performance metrics is monitored for different parameter values. The number of epochs is kept as 100 and batch size as 1000. We have conducted experiments with learning rate = [0.001–0.5], momentum = 0.9 and all the hyper-parameter values as given in Table 3.

Table 3 Hyper-parameters of proposed work

Hyper-parameters	Values/Range of values
Learning rate	[0.001–0.5]
Number of hidden layers	[1–4]
Number of epochs	100
Batch Size	1000
Dropout	0.2
Momentum	0.9

Table 4 Performance measures for LSTM network with learning rate = 0.1

LSTM Topology	Accuracy	Precision	Recall	F1 Score	ROC AUC score	Kappa score	Balanced accuracy
LSTM 1 layer	0.984481	0.775592	0.750258	0.762385	0.916804	0.968044	0.750258
LSTM 2 layers	0.985304	0.770530	0.750801	0.760303	0.909699	0.969744	0.750801
LSTM 3 layers	0.984127	0.726386	0.704230	0.714699	0.920501	0.967296	0.704230
LSTM 4 layers	0.966097	0.439111	0.396550	0.410728	0.928731	0.928330	0.396550

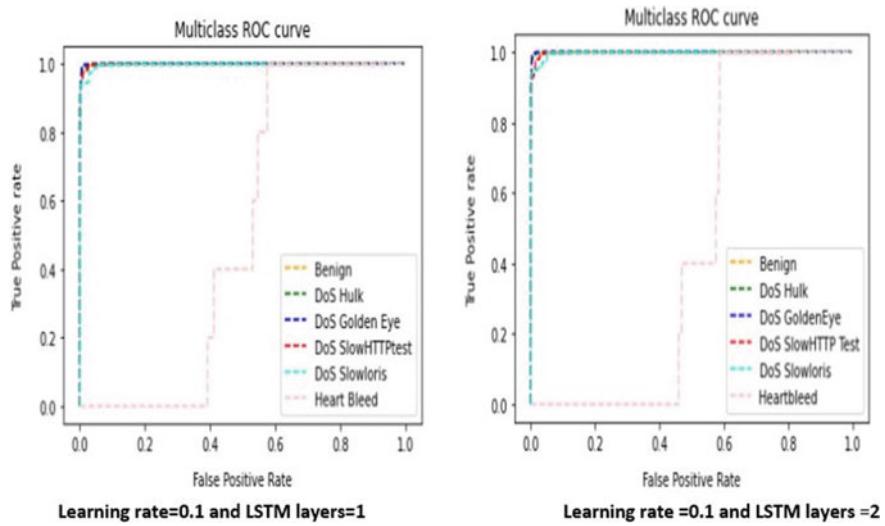


Fig. 1 ROC curve for LSTM with no of layers = 1 and 2 and learning rate = 0.1

4.1 Experimental Run with Learning Rate = 0.1

Table 4 gives the performance measures for learning rate = 0.1 and all the 4 network topologies of LSTM designed in our research study.

In Table 4, when number of layers is 1 and 2, the performance measures are better than when the number of LSTM network layers are 3 and 4. The balanced accuracy in the case of LSTM layer1 and layer 2 is 0.75. However, when the number of LSTM layers is increased to four the performance measures like Precision, Recall, F1 Score and balanced accuracy have dropped considerably while the ROC AUC score and Kappa Score remains fairly high.

In Fig. 1, we have shown the multi-class ROC curve for LSTM network with one and two layers with learning rate = 0.1.

4.2 Experimental Run with Learning Rate = 0.2

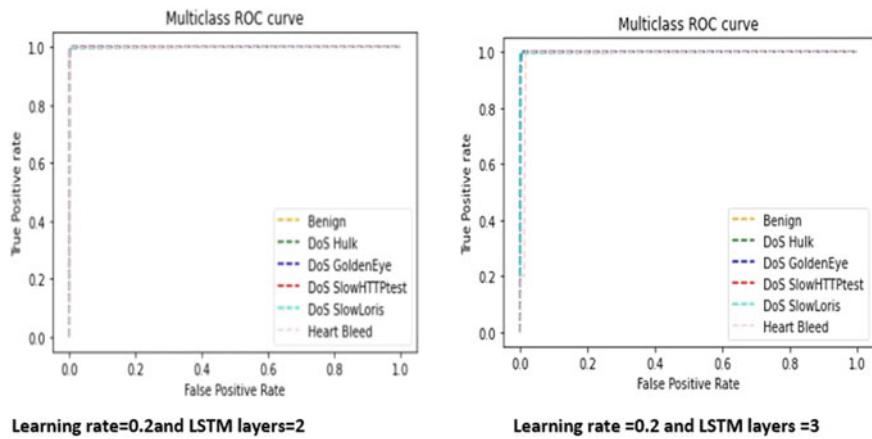
In Table 5, we have given the results of the experiments performed with learning rate = 0.02 and momentum = 0.9.

In Table 5, when the number of LSTM layers is 2, all the performance measures gave very good results. The balanced accuracy for both 2 layers and 3 layers is 0.823. However, the time taken for model execution is 1652.54 s in the case of 3 layers and 1166.97 s in the case of 2 layers.

In Fig. 2 we have shown the multi-class ROC curve for LSTM network with two and three layers with learning rate = 0.2.

Table 5 Performance measures for LSTM network with learning rate = 0.2

LSTM Topology	Accuracy	Precision	Recall	F1 score	ROC AUC score	Kappa score	Balanced accuracy
LSTM 1 layer	0.96928	0.810947	0.803778	0.806830	0.993407	0.936571	0.803778
LSTM 2 layers	0.995481	0.819564	0.823224	0.821343	0.999745	0.990690	0.823224
LSTM 3 layers	0.992573	0.818195	0.823297	0.820713	0.988386	0.988386	0.823297
LSTM 4 layers	0.990270	0.810722	0.821103	0.815832	0.984242	0.990465	0.820401

**Fig. 2** ROC curve for learning rate = 0.2and LSTM layers = 2 and 3

When we analyze the time taken for the model building, LSTM network with one layer and learning rate of 0.1 took less time of 687.95 s and the balanced accuracy score of 0.750. However, the LSTM network with 2 layers and learning rate = 0.2 took 1166.97 s and gave a balanced accuracy of 0.823. In Table 6 we have given the per class precision recall and F1 score for LSTM network with 2 layers and learning rate = 0.2 and LSTM network with one layer and learning rate = 0.1.

The most worthwhile to be noted is that both the models have failed to detect the heartbleed attacks which have only 5 samples in the testing dataset. In the case of all other categories of attacks, LSTM network with learning rate gave better performance measures as indicated in Table 6. The performance measures for LSTM 2 layers with learning rate = 0.2 gives very impressive for all the classes except the heartbleed. It is also observed during experimental runs that when learning rate is decreased from 0.1 and increased from 0.3 to 0.5, the performance measures decrease.

The highly performing two models in our experimental run are compared with [18], and the performance measures are given in Table 7. In [18], the authors have used

Table 6 Per class performance measures for high performing models

Class	LSTM Layer = 1 with learning rate = 0.1			LSTM 2 layers with learning rate = 0.2		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Benign	0.99	0.98	0.99	1.00	0.99	1.00
DoS Hulk	0.96	0.92	0.94	0.99	0.99	0.99
DoS GoldenEye	0.97	1.00	0.99	0.99	1.00	0.99
DoS SlowHTTPTest	0.86	0.78	0.82	0.96	0.99	0.97
DoS Slow Loris	0.87	0.82	0.84	0.98	0.97	0.98
Heartbleed	0	0	0	0	0	0

Bold significance are best score out of all

Table 7 Comparison of performance measures with existing work

Research work	Accuracy	Precision	Recall	F1 Score
Proposed method (LSTM with 1 layer and lr = 0.1)	0.994	0.994	0.994	0.994
Proposed method (LSTM with 2 layer and lr = 0.2)	0.995	0.995	0.995	0.995
Zhou et al. [18]	0.977	0.991	N. A	0.990

ensemble technique on the dataset used in our study. In line with our observation, the performance measures for heartbleed class are found to be not impressive by authors in [18]. In proposed method, the performance measures calculated are micro-average as it is highly imbalanced dataset and the measures are found better than the existing work.

5 Conclusion and Future Scope

With increase in the number of DoS attacks all around the world, effective mechanisms for timely detection of attacks are need of hour. In recent years, the deep learning approaches have proven to be more effective than classic machine learning algorithms. In our proposed work, LSTM network has been used with varying the number of layers and other hyper-parameters updation. Our experimental runs demonstrated the efficiency of LSTM networks in detecting multiple classes of attacks with decent performance efficiency. LSTM networks with 2 layers and learning rate = 0.02 and 1 layer with learning rate = 0.01 have given good accuracy, precision, recall, F1 score, ROC AUC score, Kappa score and balanced accuracy. The per class accuracy, precision and F1 score are found to be good for all the 5 categories with out and resampling and boosting techniques. The heartbleed attack category is found to have zero precision, recall and F1 score. In future, we will focus on detecting the smallest class with satisfactory performance measures.

References

1. Lallie HS, Shepherd LA, Nurse JR, Erola A, Epiphaniou G, Maple C, Bellekens X (2021) Cyber security in the age of covid-19: a timeline and analysis of cyber-crime and cyber-attacks during the pandemic. *Comput Secur* 105:102248
2. Tavallaee M, Stakhanova N, Ghorbani AA (2010) Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 40(5):516–524
3. Kilincer IF, Ertam F, Sengur A (2021) Machine learning methods for cyber security intrusion detection: datasets and comparative study. *Comput Netw* 188:107840
4. Aldweesh A, Derhab A, Emam AZ (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. *Knowl-Based Syst* 189:105124
5. Staude Meyer RC, Morris ER (2019) Understanding LSTM—A tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*
6. Kim J, Shin Y, Choi E (2019) An intrusion detection model based on a convolutional neural network. *J Multimedia Inf Sys* 6(4):165–172
7. Zhang H, Huang L, Wu CQ, Li Z (2020) An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput Netw* 177:107315
8. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. *J Artif intell Res* 16:321–357
9. Yulianto A, Sukarno P, Suwastika NA (2019) Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset. *J Phys Conf Ser* 1192(1):012018. IOP Publishing
10. Singh Panwar S, Raiwani YP, Panwar LS (2019) Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset. In: International conference on advances in engineering science management & technology (ICAESMT)-2019. Uttarakhand University, Dehradun, India
11. Ferrag MA, Maglaras L, Moschouyannis S, Janicke H (2020) Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. *J Secur Appl* 50:102419
12. Kim J, Kim J, Thu HLT, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 International conference on platform technology and service (PlatCon). IEEE, pp 1–5
13. Fernández GC, Xu S (2019) A case study on using deep learning for network intrusion detection. In: MILCOM 2019–2019 IEEE military communications conference (MILCOM). IEEE, pp 1–6
14. Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSp, pp 108–116
15. Hancock JT, Khoshgoftaar TM (2020) Survey on categorical data for neural networks. *J Big Data* 7:1–41
16. Brownlee J (2016) Machine learning mastery with python. Mach Learn Mastery Pty Ltd 527:100–120
17. Sherstinsky A (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys D: Nonlinear Phenomena* 404:132306
18. Zhou Y, Cheng G, Jiang S, Dai M (2020) Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput Netw* 174: 107247

Concept of Hybrid Models in Background Subtraction: A Review of Recent Trends



Saumya Maurya and Mahipal Singh Choudhry

Abstract Background subtraction (BGS) is a widely used technique in the field of computer vision for non-stationary object identification and tracking, especially in video surveillance. Hybrid models are one of the many types of approaches that can be found in the BGS literature as a result of extensive ongoing studies. This paper provides a comprehensive analysis of some of the most recent hybrid models in the BGS literature. Hybrid models are created by combining two or more models, allowing them to benefit from each other's strengths while overcoming the weaknesses of the original models. In this paper, some of the recently developed hybrid models like Hierarchical Modeling and Alternating Optimization (HMAO), randomized dynamic mode decomposition (rDMD), Adaptive Motion Estimation and Sequential Outline Separation (AME + SOS), etc. are reviewed based on their algorithms, datasets, challenges, limitations, and advantages. Descriptive analysis is done using a tabular form of review for a clear and easy understanding in addition to the comparative analysis which is performed based on f-m values of the models for a video sequence from the very popular CDnet dataset. Concluding remarks point towards the future direction of research.

Keywords Background subtraction · Hybrid models · HMAO · rDMD

1 Introduction

Object tracking and its detection are the two most significant tasks in Computer Vision finding its applications in a number of activities like Robotics, surveillance of videos, facial recognition, behavioral recognition, scene and image analysis, etc. Detection of objects is performed to identify the varied contour from the background of the scene. To achieve the mentioned task a number of methods like filtering using spatio temporal features, Optical flow and BGS can be employed. Among these methods, the most common approach followed by researchers is the BGS method.

S. Maurya (✉) · M. S. Choudhry

Department of Electronics and Communication Engineering, Delhi Technological University, Delhi 110042, India

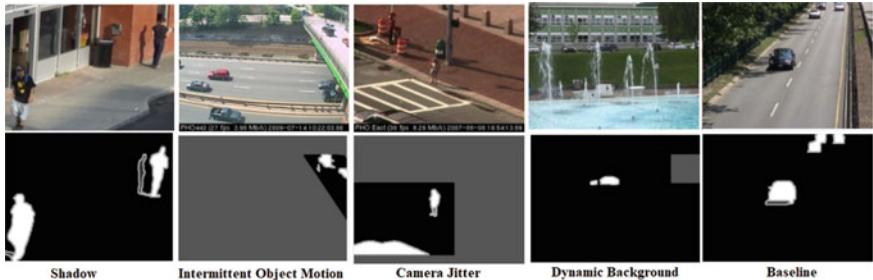


Fig. 1 Video frames with the ground truths of a few videos from different challenge categories of CDnet (2014) dataset (www.changedetection.net)

To separate the background from foreground a simple three step process involving background initialization, moving object detection using intraframe difference and background maintenance is the basic idea followed by researchers incorporating different methods.

In perfect conditions, BGS is rather an easy task to perform but in real world scenarios the presence of various challenges like intensity variation, moving background, absence of background in some frames makes the task a little challenging. There are many challenges that have been identified [1], some being simple like Dynamic Background (DB), noise, big moving object, etc. and some being complex ones like camouflage, complex dynamic background, moving camera, etc. Thus, a supercilious BGS model must be more immune to all these challenges. Figure 1 gives a pictorial representation of various challenges with their ground truths taken directly from CDnet (2014)¹ dataset.

Since BGS is such a fascinating and critical area of study, a plethora of models in various categories, using a variety of methodologies and techniques, have emerged in recent years. These categories are defined beautifully by Bowmans in [1], including statistical models like Gaussian based, support vectors, tensors, and hybrid models. In this paper, a detailed review of some recently developed Hybrid Models namely Adaptive Motion Estimation and Sequential Outline Separation (AME + SOS), Hierarchical Modeling and Alternating Optimization (HMAO), randomized Dynamic Mode Decomposition (rDMD), Gibb's Markov Random Field with maximum o' posterior probability and Gray level co-occurrence matrix (GMRF + MAP + GLCM), Singular Value Thresholding with Alternating Direction Multiplier's Method (SVT + ADMM) and Linear Spectral Clustering with non-convex Robust Principal Component Analysis (LSCNC + RPCA) is done based on their algorithms, datasets, challenges, limitations, and advantages. Hybrid models are developed by merging two or more models, thus gaining from the advantages of each other to overcome the weakness of the existing models.

The related work done in the field of BGS techniques review started with the comparison of the models in BGS category conducted by Ivor [2] by comparing

¹ <http://www.changedetection.net>.

9 techniques developed early in BGS literature. Following the trail, Piccardi [3] contrasted some methods depending on accuracy, memory requirements, and speed in 2004. Credit of doing first survey category wise goes to Cheung and Kamath [4], they divided models into non-recursive and recursive categories. Authors in [5] did the same using singular and multi monocular sensors. A survey [6] including statistical categories came into picture in 2013 that focused on all types of statistical models. Review of RPCA techniques [7] was carried out again in the same year. First comprehensive survey of all available methods is presented in [1]. Category of surveys dealing with Low rank models [8] and Neural network models [9] was conducted in the years 2016 and 2019 respectively, the latter being the latest categorical survey present in the literature.

The remaining parts of the paper are divided into three sections. A summary of the most recent hybrid models present in the literary texts of BGS with the important mathematical equations is presented in Sect. 2. Section 3. presents comparative findings in terms of f-m values for a particular video sequence from CDnet dataset along with tabular representation of data including a brief description of model algorithms, remarks, advantages, etc., and lastly Sect. 4. deals with the concluding remarks giving insights for future direction of research.

2 Recent Hybrid Models

2.1 *Integration of Local Information with Fuzzy Markov Random Field (GMRF + MAP + localGLCM)*

In the year 2016 Badri et al. [10] proposed a praiseworthy technique that detaches the in-motion objects from their casted shadows in the background. The proposed method has two main steps—subtraction of background and detection of shadows. To perform first step the background model is built with the help of median calculated using the temporal direction pixel values at a desired pixel position. Model can be mathematically described as follows-

$$Z_{(t-1)}(x, y) = \text{median}\{y_k(x, y), \quad k = 0, 1, 2, \dots, (t-1)\} \quad (1)$$

$Z_{(t-1)}(x, y)$ defines the background model's pixel value at position (x, y) till $(t-1)$ th instant of time. Authors took the advantage of RGB features including ten more local features with it at individual pixel position in both modeled reference frame and target frame to diminish the effects of color frequency changes and frequent illumination variation. Once done with the features different image can be obtained between reference and target frames as shown below-

$$d_j^f(x, y) = |p_i^f(x, y) - Z_{(t-1)}^f(x, y)| \quad (2)$$

In the above equation, f denotes the number of features and p and Z denotes the frames. To further find the non-stationary region in the scene MRF based fuzzy clustering is performed followed by maximum a' posterior probability (MAP) to divide the in-motion object with shows and backgrounds in different groups. Usage of shadow's rg color chrominance property further separation of cast shadows is done from the moving objects. The mathematical equation for shadow detection is given as-

$$r_{(t+1)}(x, y) = \begin{cases} 1; & \text{if } \Delta(x, y) \geq k_2 * \mu_\Delta \\ 0; & \text{otherwise} \end{cases} \quad (3)$$

where $r_{(t+1)}(x, y)$ denotes feature-based shadow processing, μ_Δ is calculated using average value of Δ matrix of size $M * N$ and k_2 ranging from 0.5 to 1.5 is a constant. The method's key flaw is that the value of features can be easily modified even with a small shift in significant bits' binary level.

2.2 Randomized Low Rank Dynamic Mode Decomposition (rDMD)

To overcome the problem of inflated computational time to process the high-resolution videos, Ericson and Donovan [11] proposed a model based on data driven method DMD fusing PCA and Fourier Transform. The given technique handled the dynamic background issue in video processing with the help of a randomized matrix algorithm for fast computing the low rank DMD. Usage of probabilistic SV decomposition (SVD) algorithm utilizing swiftly decomposing singular values of given data increases the computational savings of the presented method in comparison to traditional SVD algorithms significantly by 10–30 times as mentioned by authors. Moreover, rDMD is 2–3 times faster than the DMD infused with deterministic SVD. SV decomposition is given by the equation:

$$Z = X \Sigma Y^* \quad (4)$$

where Σ is a diagonal matrix and X, Y are orthogonal matrices. For improved computational time randomized SVD is used by the authors which is nothing but a linear algebra problem made up of projections and random sampling. For background modeling, main focus is given to the low rank features of the given video stream. Expression of Low rank DMD can be defined as- $A \approx \phi R V_{\text{and}}$, ϕ denotes the dynamic modes of the given data matrix, R is an amplitude diagonal matrix described as

$$R = \begin{pmatrix} r_1 \\ & r_i \\ & & \ddots \\ & & & r_k \end{pmatrix} \quad (5)$$

V_{and} is a Vandermonde matrix of eigen values

$$V_{\text{and}} = \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_k & \dots & \lambda_k^{n-1} \end{pmatrix} \quad (6)$$

The difference between background and original video data gives the identification of foreground objects. The main advantage is that dynamic modes can vary according to the requirement of video data.

2.3 Hierarchical Modeling and Alternating Optimization (HMAO)

The method put forward by Li et al. [12] joins hierarchical modeling with alternating optimization to separate the foreground and background of video sequence. Background is hierarchically represented by breaking the sequence into high and low frequency counterparts ($z = z^l + z^h$) to deal with dynamic background problems in a better way. Principal components belonging to the residuals are taken into account for detailed background instead of wasting them by outlier treatment for finer details. For better robustness towards noise, first, the foreground is detected at the low resolution and then move on to further resolutions. Furthermore, graph cut method ensures the joint enforcement of l_1 norm foreground constraint and background's rank-1 constraint.

Final joint estimation problem formed as per the defined background and foreground models above is given in (7) which is further solved using Alternating Direction Multipliers Method (ADMM).

$$\begin{aligned} \min_{X, F, \Omega} & \left\{ \sum_G f(a) \|\rho_{i,j,n} - \rho_{x,y,z}\| + \gamma D(\rho) + \beta \|\rho\|_0 \right. \\ & \left. + \sum_{i,j} \|X_{i,j,:,:} - X_{i,j,:,:}^{l*} J_{i,j,*4} O - X_{i,j,:,:}^h\| \right\} \\ \text{s.t.rank} & \left(X_{\text{vec}}^{l*} \right) = 1, \quad \|O\|_2 = 1, \quad D = P_{\bar{\rho}} X + Z \end{aligned} \quad (7)$$

O denotes the changing tendency; X is the background and $D(\rho)$ represents the prior knowledge of foreground, (i, j) represents the pixel location. Model can successfully differentiate regularly appearing background objects from noise.

2.4 Generalized Singular Value Thresholding Operator (GSVT) Based Non-convex Low Rank and Sparse Decomposition (LRSD)

Since the nuclear norm minimizes the singular values in one go, this limitation of it for approximation of rank function is solved by the researchers in [13] by exploitation of a surrogate function to solve the low rank matrix and they came up with the idea of performing non-Convex LRSD using GSVT operator and solving the formulated problem using ADMM. The GSVT operator can be defined as follows-

$$\text{Prox}_g^\sigma(A) = \arg \min_L \sum_{j=1}^{x_1} g(\sigma_i(L)) + \frac{1}{2} \|L - A\|_h^2 \quad (8)$$

where g is a non-convex surrogate function, A is the complete matrix, low rank matrix being L , and σ denotes L 's singular value. ADMM function in mathematical terms is given as

$$\mathcal{L}(L, B, C, \mu) = \sum_{j=1}^{x_1} g(\sigma_i(L)) + \lambda \|S\|_1 - \{C, L + B - M\} + \frac{\mu}{2} \|L + B - M\|_h^2 \quad (9)$$

μ controls the penalty, C is Lagrangian multiplier, λ being trade off parameter, M is known data matrix, L being low rank matrix, B is sparse matrix and $\{\cdot\}$ is inner product of matrix. Logarithmic penalty is used here by the researchers. The model fails to perform efficiently in presence of complex scenarios.

2.5 Adaptive Motion Estimation + Sequential Outline Separation (AME + SOS)

Thenmozhi and Kalpana [14] used AME approach long with SOS to put forward an effective object detection technique that efficiently handles issues like sudden light change, closer view gap, and ghosting. The method proposed by authors performs foreground detection and BGS to detect in-motion objects followed by separation of territory of interest from the constructed background. The condition taken into account is that the background is completely stationary devoid of any moving object.

The whole process is divided into four stages comprising initialization of background, and its modeling, subtraction of background and AME followed by SOS. For modeling of background BGS strategy based on dual motion position is exploited to point out the differences between current image and background image. AME segmentation helps in identifying frame consisting of a moving object. In SOS process, the refinement of the collection with the help of vectors in motion is studied. Conversion of video frames into gray ones is carried out as given in Eq. (10) to reduce the cost of the process.

$$Z_t = 0.125A_t(a, b) + 0.7512B_t(a, b) + 0.072C_t(a, b) \quad (10)$$

A, B and C denotes the color combinations of each frame. Background modeling is performed according to the given equation below in Eq. (11).

$$X_1(a, b) = \begin{cases} X_r(a, b), & \text{if } X_r(a, b) \leq P \\ X_{r-1}(a, b), & \text{if } X_r(a, b) > P \end{cases} \quad (11)$$

X_{r-1} represents previous image background, X_r is current background frame and P is the current frame sequence. The threshold deciding a pixel as a current object or background is defined as:

$$Y(x, y) = q(x, y) + k\sigma(x, y) \quad (12)$$

q is the local mean of the pixel, σ being its standard deviation controlled by variable k . Presence of pixel energy factors within the threshold of group pixel energy factors ensures the assignment of that pixel to the cluster. Non-stationary region of images is calculated using difference of image sequence between consecutive frames.

2.6 Non-convex Rank Approximation Robust Principal Component Analysis and Super-Pixel Motion Detection Using Linear Spectral Clustering (LSCNC-RPCA)

In this approach [15] traditional RPCA technique is infused with motion detection using super pixels to perform background separation, and the model is further solved with the help of Lagrange's augmented alternating direction technique. Segmentation method performed using super pixels uses a particular area of image having some visual importance formed by a group of nearby pixels holding some similar features like texture, color, etc. Authors performed segmentation in initial step to extract the grouping matrix consisting of super pixels. The pixels in the possession of the j th super pixel is marked as j ($j = 1, 2, \dots, k$) and a grouping matrix $H \in R^{m*n}$ is obtained by arranging each image frames' information as a column vector, which is later used for the extraction of sparse foreground. Moving forward, improved version

of traditional RPCA (1) is incorporated for obtaining sparse matrix.

$$A^{k+1} = \arg \min_A ||A||_Y + \frac{\mu^k}{2} \left\| A - \left(C - D^k - E^K + \frac{F^t}{\mu^t} \right) \right\|_F^2 \quad (13)$$

A being low rank matrix, C the sequence of visual frequency, μ being the parameter of penalty, D and E being sparse foreground and dynamic backgrounds and F is Lagrange multiplier. In the final steps, results of above two steps are combined, forming a motion mask in return for identifying the video's moving objects. The threshold value t deciding the mask result is calculated as follows-

$$t = \arg \min_g (\alpha_0(a)\sigma_0^2(a) + \alpha_1(a)\sigma_1^2(a)) \quad (14)$$

$\alpha_0(a)$ and $\alpha_1(a)$ denotes moving pixels probability having pixel value less than t , and σ represents the variance between classes. Authors used LSC with non-convex RPCA to develop the proposed LSCNC-RPCA model. The mentioned method has an upper hand in terms of complex computation and storage due to the inclusion of Linear Spectral Clustering in it during segmentation.

3 Comparative and Descriptive Analysis

The most common approach used in BGS techniques to comment on efficiency is by taking the F -measure value calculated on various video sequences taken into account. The F -measure is characterized in terms of precision and recall, which are respectively defined as correctly classified positive samples compared to total samples and true positive specimens compared to total positive classified cluster. Mathematical expressions of these terms are given below

$$\text{Precision} = \frac{A_p}{A_p + B_p}; \quad (15)$$

$$\text{Recall} = \frac{A_p}{A_p + B_n}; \quad (16)$$

$$F - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}; \quad (17)$$

Here, A_p being true positive, B_p being false positive and B_n being false negatives. The F-measure of different models calculated on video sequence of Highway from baseline category of CDnet (2014) dataset is shown in Fig. 2. The plot shows that rDMD model has the highest Fm among all the reviewed models. Table 1 presents a tabular representation of important aspects of the above discussed models.

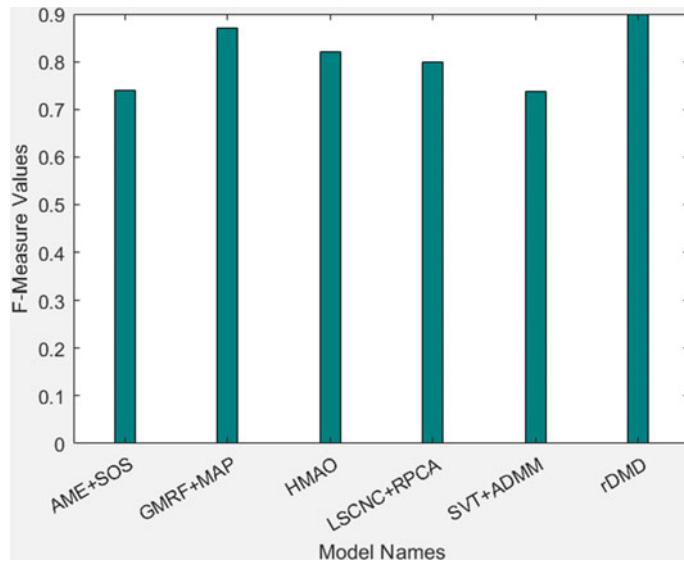


Fig. 2 Plot of Fm values of discussed models for baseline category (Highway) of CDnet dataset

Table 1 Overview of above discussed models in terms of various parameters

Model No.	Algorithm used	Issues addressed	Remarks	Limitations	Dataset used	Authors (Year)
2.1	GMRF + MAP + localGLCM	Cast Shadows on moving objects	<ul style="list-style-type: none"> Shadows are detected using GLCM dependent local features 	<ul style="list-style-type: none"> Noise sensitivity Failure in handling sudden illumination change and data recorded by in-motion camera 	CDnet, BMC	Badri et al. (2016) [10]
2.2	rDMD	Illumination change, Bootstrapping, Camouflage, Dynamic Background	<ul style="list-style-type: none"> High computational speed having 180 frames per second as frame rate 	<ul style="list-style-type: none"> Cannot perform well with sleeping foreground objects 	CDnet, BMC	Erichson and Donovan (2016) [11]

(continued)

Table 1 (continued)

Model No.	Algorithm used	Issues addressed	Remarks	Limitations	Dataset used	Authors (Year)
2.3	HMAO	Dynamic and Complex backgrounds with illumination change, Noise	<ul style="list-style-type: none"> Residual principal components are not treated as outliers but are used to obtain detailed background 	<ul style="list-style-type: none"> Limited performance while dealing with night videos, camera jitter, clutter, and irregular dynamic motion in background 	CDnet, I2R	Li et al. (2019) [12]
2.4	SVT + ADMM (Non-LRSD)	Baseline effect, Noise in frames	<ul style="list-style-type: none"> High accuracy and processing speed with 156 f/s frame rate 	<ul style="list-style-type: none"> Low efficiency with complex scenes and camouflaged videos 	CDnet, I2R	Yang et al. (2019) [13]
2.5	AME + SOS	Closer view gap, Ghosting, Sudden lighting change	<ul style="list-style-type: none"> Classification method based on frame difference is used to detect object motion High accuracy of 97.45% 	<ul style="list-style-type: none"> Model struggles in performance when encountered with complex dynamic background scenes 	CDnet	Thenmozhi and Kalpana (2020) [14]
2.6	LSCNC-RPCA	Complex dynamic background	<ul style="list-style-type: none"> Super-pixel blocks can be adjusted according to the moving objects' size 	<ul style="list-style-type: none"> Fails to extract tiny moving objects present in foreground 	CDnet, I2R	Wang et al. (2020) [15]

4 Conclusion

This paper has presented a detailed review of some recently developed hybrid models in the field of BGS. The review shows that hybrid models are successful in addressing most of the difficult issues such as complex background, camouflage, etc. It is evident that algorithms formed using randomized matrix can also be used for other linear algebra-based models based especially those using SVD. Moreover, the use of DMD is an interesting topic as it can give fast results but at the cost of high precision and it can be presented as RPCA's fast approximation. The issues like complex videos and shaking cameras can be easily dealt with by AME + SOS with moderate accuracy. Issue of insufficient boundary observation and holes in extraction of foreground can be handled effectively by LSCNC + RPCA. Most of the issues with dynamic background is solved by HMAO with a performance better than many available states of art methods. The review has shown that although the models reviewed here are latest in the literature and are successful in addressing some critical issues, they are not enough to deal with night video, thermal and turbulence, or situations with multiple critical challenges. Future work may include development of a more robust model handling a greater number of issues along with the critical ones as well.

References

1. Bouwmans T (2014) Traditional and recent approaches in background modeling for foreground detection: an overview. *Comput Sci Rev* 11–12:31–66. <https://doi.org/10.1016/j.cosrev.2014.04.001>
2. Ivor M (2000) Background subtraction techniques. In: International conference on image and vision computing, New Zealand, IVCNZ 2000, Nov 2010
3. Piccardi M (2004) Background subtraction techniques: a review. In: IEEE international conference on systems, man and cybernetics (IEEE Cat. No.04CH37583), vol.4, The Hague, Netherlands, pp 3099–3104
4. Cheung S, Kamath C (2005) Robust background subtraction with foreground validation for urban traffic video. *EURASIP J Adv Signal Process* 726261. <https://doi.org/10.1155/ASP.2005.2330>
5. Cristani M, Farenzena M, Bloisi D, Murino V (2010) Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP J Adv Signal Process*. <https://doi.org/10.1155/2010/343057>
6. Bouwmans T, El Baf, Vachon B (2010) Statistical background modeling for foreground detection: a survey. *Handbook of pattern recognition and computer vision*, vol 4, issue 2. World Scientific Publishing, pp 181–199
7. Bouwmans T, Zahzah EH (2014) Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance. *Comput Vision Image Understand* 122:22–34. <https://doi.org/10.1016/j.cviu.2013.11.009>
8. Bouwmans T, Sobral A, Javed S, Jung SK, Zahzah EH (2017) Decomposition into low-rank plus additive matrices for background/foreground separation: a review for a comparative evaluation with a large-scale dataset. *Comput Sci Rev* 23:1–71. <https://doi.org/10.1016/j.cosrev.2016.11.001>

9. Bouwmans T, Javed S, Sultana M, Jung SK (2019) Deep neural network concepts for background subtraction: a systematic review and comparative evaluation. *Neural Netw* 117:8–66. <https://doi.org/10.1016/j.neunet.2019.04.024>
10. Subudhi BN, Ghosh S, Cho SB, Ghosh A (2016) Integration of fuzzy Markov random field and local information for separation of moving objects and shadows. *Inf Sci* 331:15–31. ISSN 0020-0255
11. Erichson NB, Donovan C (2016) Randomized low-rank dynamic mode decomposition for motion detection. *Comput Vision Image Understand* 146:40–50
12. Li L, Hu Q, Li X (2019) Moving object detection in video via hierarchical modeling and alternating optimization. *IEEE Trans Image Process* 2(4):2021–2036. <https://doi.org/10.1109/TIP.2018.2882926>
13. Yang Z, Fan L, Yang Y, Yang Z, Gui G (2019) Generalized singular value thresholding operator based nonconvex low-rank and sparse decomposition for moving object detection. *J Franklin Inst* 356(16):10138–10154
14. Thenmozhi T, Kalpana AM (2020) Adaptive motion estimation and sequential outline separation based moving object detection in video surveillance system. In: *Microprocess Microsyst* 76:103084
15. Wang Y, Wei H, Ding X, Tao J (2020) Video background/foreground separation model based on non-convex rank approximation RPCA and superpixel motion detection. *IEEE Access* 8:157493–157503. <https://doi.org/10.1109/ACCESS.2020.3018705>

Artificial Neural Network Approach for Multimodal Biometric Authentication System



M. J. Sudhamani, Ipsita Sanyal, and M. K. Venkatesha

Abstract The engrossing development in biometric authentic system renders secured access to civilian and public information. Substantial evidences in the literature defeats the unimodal biometric authentication system and entails the significant research on multimodal biometric authentication system. High fidelity, non-invasive and undetecting properties of finger vein features are amalgamated with face features in the proposed work. Employment of Convolution Neural Network (CNN) waives off the time-consuming Region-of-Interest (ROI) extraction process. The model was developed and trained on GPU enabled online cloud service platform. The effectiveness of the CNN architecture is escalated with the aid of fine-tuning the hyperparameters like insertion of dropouts and minibatch selection. Minimal preprocessing step involving min–max normalization contributes prominently to reduce computation time. The normalized ensembles are then fed into the CNN model with the softmax classifier for feature extraction. This work also highlights the novelty of achieving a potent authentication model with minimal features, outperforming the existing state-of-the-art authentication systems. Various classifiers are used to evaluate the model and a minimum Equal Error Rate (EER) of 0.46% is accomplished.

Keywords Finger vein · Face · Feature level fusion · Multimodal biometrics · Convolution neural network · Overfitting · Underfitting · True positive rate · False positive rate · EER · Support vector machine

M. J. Sudhamani

Department of Computer Science Engineering, RNS Institute of Technology, Bangalore, India

I. Sanyal (✉)

Department of Electronics and Communication Engineering, RNS Institute of Technology, Bangalore, India

M. K. Venkatesha

RNS Institute of Technology, Bangalore, India

1 Introduction

An imperative obligation of information security emphasizes on authentication system. Conventional methods of authentication relying on passwords, pins, and tokens are liable to attacks, hence biometric authentication has emerged. The field of authentication is widely automated by employing Biometrics. Incessant evolution in this field has led to the accomplishment of promising consequences. The conventional authentication systems using pins and ids in various applications are extensively replaced by biometric authentication. Biometrics is generally apportioned into two classes: unimodal biometrics and multimodal biometrics. Unimodal biometrics is restricted to authenticate subjects using single cues. The decision-making process in multimodal biometrics includes the integration of diverse traits of a subject. Unimodal biometrics have undeniable limitations like inter-class resemblance, distortion in the extracted data, and intra-class disparity [1] hence, vulnerable to unauthorized access. Assimilating multimodal biometrics momentously improves the reliability and performance of the single cue-based authentication system. Furnishing precise results is a computationally intensive and time-consuming operation. The primary challenge in developing a robust multimodal biometric system is the assortment of features and the level of fusion [2]. The notion of automated feature extraction from the input ensemble, and autoencoding scaling down of a large input image to a significantly low dimensional attribute space, proposed by Hinton and Salakhutdinov [3] led to the extensive application of Deep Neural Networks (DNN) and its variants in the field of pattern recognition [4]. Convolution Neural Networks (CNN) is a variant of DNN which automates the process of extracting distinctive features from the raw input images [5]. CNN's are also immune to translation, rotation, and scaling of the input ensemble [6]. This work embraces the deployment of CNN for feature extraction from two different modalities, i.e., face and finger vein and their fusion at feature level which is then subjected to various discriminative learning techniques for classification, hence overcoming the critical limitations of reliability, performance, and time.

The paper is arranged as follows, Sect. 2 deals with the state-of-the-art deep learning methodologies used in the field of multimodal biometrics, Sect. 3 provides an in-depth information of the proposed methodology by giving a brief explanation of preprocessing and normalization, training of CNN model, minin batch and optimization, feature extraction and fusion and classification. Section 4 deals with the experimental results, and Sect. 5 concludes the work proposed through this paper.

2 Literature Survey

Evolution in the field of multimodal biometrics authentication system has led to remarkable results. The state-of-the-art deep learning perspectives of multimodal biometric systems using CNN architecture are revisited in this section.

AI-Waisy et al. [7] proposed an identification system using the face and iris as the biometric trait. CNN and DBN were used as feature extractor and the traits were fused at score level. The classification accuracy ranged from 99.19 to 100% over the various datasets like NIST, CASIS V1.0, MMU1 and SDUMLA-HMT. Ding and Tao [8] proposed an eight-layered CNN. Feature extraction was carried out incorporating a three-layered stacked autoencoder. The facial dataset of Labeled Faces in the Wild (LFW) and CASIA-WebFace facial dataset used in the work furnished an accuracy of 76.53% and 99% respectively. Veluchamy and Karlmarx et al. [9] proposed an identification system based on feature level fusion of finger knuckle and finger vein traits. The work comprises of IIIT Delhi finger knuckle dataset and SDUMLA-HMT finger vein dataset. Algorithm combining ANN and SVM classifiers was employed rendering an accuracy of 96%. A fused convolutional-subsampling four-layer CNN was proposed by Radzi et al. [10] for authentication using finger vein which is a modified version of stochastic diagonal Levenberg–Marquardt algorithm. This algorithm reduces the convergence time of the Neural Network. In [11], an identification system was established using mouse gesticulation and keystroke dynamics, fused at feature level. The K-Neural Network classifier furnished an accuracy of 68.8%. Li et al. [12] proposed CNN architecture like Alexnet and VGG16 for the feature extraction from the dorsal hand vein trait. Classification of the extracted features was done through KNN classifiers. In [13], the authentication was based on dorsal hand vein patterns, which were extracted using the clustering-based segmentation and mathematic morphology and rotation invariant Hough transform. The patterns were fed to a neural network for classification and an EER (Equal Error Rate) of 0.83% was recorded. A comparative study has been carried out in [14] using two CNN fusion architectures namely bilinear architecture and fully connected architecture. Face and iris are fused at feature level and are classified using the softmax classifier. The Bilinear and fully connected architecture achieved an accuracy of 99.85% and 99.65% respectively. The finger vein authentication projected in [15] utilizes two vivid architectures namely LightConvolutioNueralNetwork with triplet loss function and CNN with supervised discrete hashing technique. An EER of 13.16 and 9.77% is recorded for both the techniques. Hong et al. [16] proposed transfer learning for classification of finger vein using pretrained model like VGG-16, VGG-19 and VGG Face classifiers, an EER of 6.115 and 0.804 were obtained for low quality and high quality images respectively. The work of Al-Waisy et al. [17], proposes a real-time multimodal biometric authentication system called IrisConvNet by the fusion of left and right irises at rank level resulted in an identification rate of 100%. The 3 layer CNN architecture is employed for feature extraction and classification using the softmax classifier. Sayed et al. [18] proposed human authentication through gait sequence classification where the features from the gait sequence were extracted using a simplified fuzzy CNN architecture and encompassed a softmax classifier which recorded an efficacy of 98.7%. The proposal mentioned in [19] utilizes the finger vein images which were preprocessed by normalization and adaptive contrast enhancement and an accuracy of 98.33% was obtained through Contrast Limited Adaptive Histogram Equalization (CLAHE) of the given images. The datasets like SDUMLA, FVUSM, HKPU and UTFVP with various qualities of images were used

in the experimentation. The UTFVP dataset outperformed with an accuracy of 95%. In 2019 Alay and Al-Baity [20] recommended an end-to-end CNN architecture for feature extraction. Face and iris traits were fused at feature and score level. The classification accuracy at feature level was recorded as 99.22% and fusion at score level achieved an accuracy of 100%. Song et al. [21] addressed the problem of noisy images using densely connected neural network (DenseNet) architecture for feature generation from finger vein images. The difference between the generated features was the score estimates. In [22], three CNN architecture Unet, Segnet and RefineNet were proposed for feature extraction from the finger vein images, and a softmax classifier was used for classification. UTFVP database was used and Unet furnished remarkable EER of 0.322%.

3 Proposed Methodology

The proposed model contemplates the face and finger vein images from the SDUMLA-HMT dataset [27] involving 106 subjects. Image preprocessing and normalization for all the images of face and finger vein are carried out. The preprocessed images are labeled according to their respective classes and are split into training, validation and testing set. The features of the face and the finger vein are extracted from their respective trained CNN model. The extracted features from the two modalities are fused at feature level. The min–max normalization technique is employed for normalization of the fused features. The normalized features are further classified using various classification algorithm and the results are compared to get the best suited classification algorithm for authentication. The proposed methodology is illustrated in Fig. 1.

3.1 Preprocessing and Normalization

The RGB finger vein and face images in the SDUMLA-HMT dataset are of dimensions 320 * 240. Prior to training, the images are converted to grayscale. In order to facilitate training time reduction, the images are resized to a dimension of 64 * 64. Preprocessed images are normalized by dividing each pixel of the image by 255. The normalization step prevents the gradients from assuming a larger value during the time of backpropagation. Subsequently, image augmentation is done for increasing the training data. The proposed model uses various augmentation techniques on finger vein images, namely rotation, shearing, cropping, resizing and translation. Figure 2 illustrates the sample preprocessed and augmented images from the dataset.

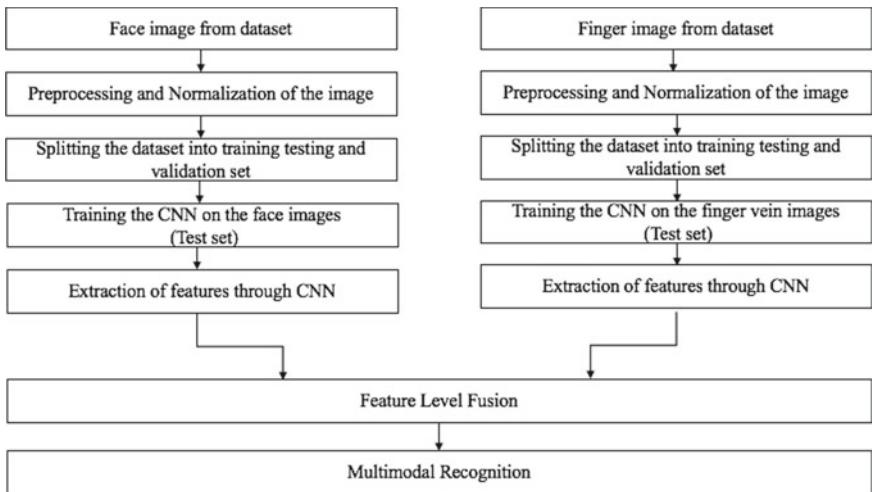


Fig. 1 Methodology of the proposed model

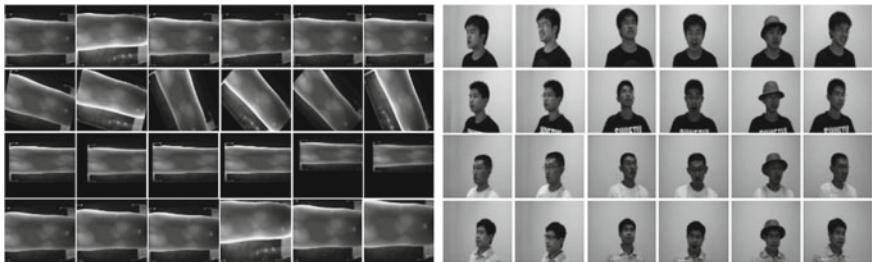


Fig. 2 Sample dataset images illustrating finger vein image augmentation and sample dataset images illustrating preprocessed face images

3.2 *Training of CNN Model*

Features of the image are accentuated or depreciated by filtering in spatial domain. Linear filtering in digital image processing is attained using Convolution. Hence, the computation of the linear function with the convolution operator in a neural network is termed as Convolution Neural Network. Layers of CNN can be contrived from any of the three fundamental layers, namely convolution layer, pooling layer or fully connected layer.

Validation accuracy is decisive in determining the value of bias and variance of the neural network during classification. Prior to feature extraction, the CNN must be trained on the preprocessed image database. The validation accuracy can be computed by splitting the data set into test, train and the validation set. In the proposed work the database is split into 70% of training set, 15% of validation set and 15% of testing set.

This work proposes a nine-layered CNN with seven hidden layers. The input layer consists of the preprocessed labeled images of dimension $64 * 64$, which is invariably the input to the first hidden layer of the CNN. The first hidden layer consists of the convolution operation, followed by the activation function. Max-pooling layer is the second hidden layer. The proposed work uses a $3 * 3 * 32$ dimension kernel in the first hidden layer for performing convolution.

The dimension of an image after convolution operation is found using Eq. (1). A $64 * 64$ dimension image is converted to $62 * 62 * 32$ feature space.

$$(L, W, D) = \left[\frac{(N_h + 2p - F)}{S} + 1, \frac{(N_w + 2P - F)}{S} + 1, N_c \right] \quad (1)$$

where L is the length, W is the width and D is the depth of the output image, N_h is the height and N_w is width of the input ensemble, dimension of the filter is F , the padding parameter is P , stride parameter is S and N_c is the number of filters used for convolution.

The activation functions are used to insert nonlinearities to the output of each neural network layer. Proposed work uses the Rectified Linear unit activation function represented in Eq. (2), as it is applicable for multi-class classification problem. The output value can be elucidated as the estimation of nearness of the given input ensemble to the multiple classes used in classification. There are various activation functions namely sigmoid, tanh, leaky-ReLu, Elu, etc. Out of the aforementioned activation functions, vanishing gradients is the major limitation of tanh and sigmoid functions. Moreover, tanh and sigmoid functions are not applicable for multiclass classification problems. This justifies the suitability of ReLu activation function for the proposed work.

$$R(p) = \begin{cases} 0 & \text{if } p < 0 \\ p & \text{if } p \geq 0 \end{cases} \quad (2)$$

The second hidden layer is the max-pooling layer, the dimensions of the output obtained from pooling layer is calculated using Eq. 1. The model includes four convolution layers where the output of the 1st, 3rd and 4th convolution layer is given to a max-pooling layer and the 2nd convolution layer has a dropout associated with it. Each of the convolution layers is assigned with the ReLu activation function. The flatten layer output is input to the fully connected 6th layer. The flatten layer converts the multidimensional feature space to 1D feature space. The 7th fully connected layer is referred as the feature descriptor. The features are extracted from this layer after training the model. The output of the last fully connected layer is given to the softmax layer represented in Eq. (3) which is instrumental in computing the probability distribution of the input image to all the existing classes. The proposed CNN model configuration is tabulated in Table 1.

$$f(x) = \frac{e^{x_j}}{\sum_{i=1}^K e^{x_i}} \quad (3)$$

where x_j is the j th input vector and K is the number of classes.

Owing to the large number of trainable parameters in CNN the issue of high bias and variance is inevitable. High variance termed as overfitting, is characterized by a model performing effectively on training set but poorly on validation set. High bias or underfitting occurs due to the poor performance of the model on both training and validation set. The measure of the bias and variance is the measure of a model's effectiveness in classification. Low values of bias and variance is a criterion for a model to be accurate in its predictions. The cause for high values of bias and variance are diverse and complex but it can be rectified by calibrating the hyperparameters, by the proper selection of optimization algorithm, increasing size of dataset, varying the learning rate α , increasing the number of hidden layers, L1, L2 regularization, dropout regularization, etc.

Dropouts can be cited as one of the regularization techniques in which the units of neural network layers are temporarily deactivated at random leading to the creation of thin neural network [23]. They are applied both in feed-forward and backpropagation of neural networks. It addresses the issue of overfitting. Exponential coalition of

Table 1 Configuration of the proposed CNN model

Types of layer	# of filters	Feature map size (Height * Width * Channel)	Kernel size
Image input layer	–	64 * 64 * 1	–
1st Convolution layer	32	62 * 62 * 32	3 * 3
Relu-1			
Max-pool 1	1	31 * 31 * 32	2 * 2
2nd Convolution layer	64	29 * 29 * 64	3 * 3
Relu-2			
Dropout (0.5)		29 * 29 * 64	
3rd Convolution layer	128	27 * 27 * 128	3 * 3
Relu-3			
Max-pool 2	1	13 * 13 * 128	2 * 2
4th Convolution layer	128	11 * 11 * 128	3 * 3
Relu-4			
Max-pool 3	1	5 * 5 * 128	2 * 2
Flatten 5th layer	0	(3200, 1)	–
Fully connected layer 6	–	(512, 1)	–
Fully connected layer 7	–	(100, 1)	–
Softmax layer 8	–	–	–
Output	–	100 Class	–

the different neural network architecture is also efficiently supported using dropouts [23]. This work proposes the implementation of dropouts for the prevention of overfitting. Through experimentation, it was found that the best result was achieved by introducing the dropouts in the 2nd hidden layer of the proposed CNN structure. The feed-forward operation with dropouts is as shown in Eqs. 4–6.

$$\tilde{y}^{(i)} = r^{(i)} * y^{(i)} \quad (4)$$

$$Z_j^{(i+1)} = W_j^{(i+1)} \tilde{y}^i + b_j^{(i+1)} \quad (5)$$

$$y_j^{(i+1)} = f(Z_j^{(i+1)}) \quad (6)$$

The thinned output layer in the i th layer of the neural network is $\tilde{y}^{(i)}$ as given in Eqs. 4 and 5 and $r^{(i)}$ is a vector of Bernoulli random variables [23] each of which is associated with a probability p . In Eq. 6, $y^{(i)}$ is the output of the i th layer, for $i = 0$ $y^{(0)}$ is the input to the neural networks, $Z_j^{(i)}$ is input vector to the activation function of the i th layer of the j th unit, f is the activation function. For the proposed work, the dropout is introduced in the second convolution layer hence i assumes a value of 2 and the probability p is 0.5, which is an optimal choice for a wide range of neural networks [23]. In this work, the validation accuracy is computed by inserting the dropouts at various layer of the proposed CNN architecture. The computed validation accuracy is as shown in Table 2. In the absence of dropouts the model yields comparatively low validation accuracy than the one with dropouts. Insertion of dropouts in the 2nd convolution layer outperforms the other combinations with an observed validation accuracy of 97.03% and 96.76% for finger vein and face modalities respectively. The rectified linear unit (ReLU) is employed as the activation function f . Similarly, in backpropagation the neural networks will be thinned out since some units of a layer will be temporarily dropped at random.

Table 2 Validation accuracy to demonstrate optimal choice of dropout layer position

Dropout layer position (i)	Validation accuracy (%)	
	Finger vein	Face
1st layer	95.85	95.32
2nd layer	97.03	96.76
1st and 2nd Conv layer	96.14	94.24
No dropout	92.88	90.65

3.3 Minibatch and Optimization

Accurate classification is realized by training the convolution neural network. Training the neural networks over a large training set is computationally intensive leading to increased learning time. Hence, the entire training set is randomly divided into groups of smaller size, generally to numbers equal to the power of 2. These groups are labeled as minibatch and the best default size of a minibatch is 32 [24]. A comparative study has been carried out to determine the best batch size based on execution time for the minibatch size of 32 and 64. Table 3 illustrates the elapsed training time for both face and finger vein traits by varying batch size. It is observed that the minibatch size 64 outperforms compared to minibatch size of 32 incurring 7.7 h of training time.

Magnitude of the loss function is reduced by gradient descent. The loss function is formulated by the categorical cross-entropy, the formula is as represented in (7). Categorical cross-entropy is the ideal choice in the case of multiclass classification where the output of the CNN must only relate to a single class out of the many labeled classes.

$$L(y, \hat{y}) = \sum_{i=0}^M (y_i * \log(\hat{y}_i)) \quad (7)$$

where y_i is the true value of the i th class, \hat{y}_i is the predicted value of i th class and M is the total labeled classes.

There are a variety of gradient descent optimization algorithms available till date like Adagrad, RMSprop, Adadelta, AdaMax, Momentum-based SGD (Stochastic Gradient Decent), etc. Proposed work uses Adam (Adaptive Moment Estimation) as the gradient descent optimization algorithm for training CNN. Adams is more efficient and versatile [25] than the other optimization algorithm. It is based on RMSprop and momentum SGD. It combines the advantages of RMSprop [25] which has a good performance with non-stationary and online settings and Adagrad which works efficiently with sparse gradient [25, 26]. The hyperparameters of Adams algorithm are step size or learning rate α , decay rate β_1 , β_2 and a small number ϵ to forbid division by zero during implementation, generally the default values of these hyperparameters, i.e., $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ & $\epsilon = 10^{-8}$ are the default settings

Table 3 Comparison of training time based on minibatch size

Traits				Total training time (h)
Face		Finger vein		
Minibatch size	Time taken (h)	Minibatch size	Time taken (h)	
32	4.17	32	4	8.17
64	3.4	64	3.3	7.7

that yield optimal results [25]. Nevertheless, these hyperparameters can be tuned accordingly by ones need. The proposed work retains the default values of these hyperparameters.

3.4 Feature Extraction, Fusion and Classification

The CNN is trained separately on the images of the two modalities. After training the CNN model, all the images (10,200 finger vein and 8400 face images) of the two modalities are given to their respective previously trained CNN model and the features are extracted from the 7th hidden layer (viz the fully connected layer). The finger vein modality yields 10,20,000 features (102 images per subject * 100 subjects * 100 features per subject) and the dimension of the feature space being 100 * 10,200. Similarly, from face 8,40,000 features (84 images per subject * 100 subjects * 100 features per subject) are extracted and the dimension of the features space is 100 * 8,400.

These extracted features are fused at feature level by vertical concatenation of the two feature spaces. The fused features are then subjected to the min–max normalization technique. It is instrumental in scaling the features from the modalities to an identical range of values (i.e., between 0 and 1). The formula for min–max normalization is as shown in Eq. 8.

$$\text{norm}(f_i) = \frac{f_i - \min(f)}{\max(f) - \min(f)} \quad (8)$$

where f_i is the i th feature value, $\min(f)$ is the minimum value in the feature vector, $\max(f)$ is the maximum value in the feature vector. Following the feature level fusion, the fused features are split into testing set and training set by means of K-fold cross-validation, which prevents the problem of overfitting up to a certain extent. Classification algorithms namely Support Vector Machine (SVM) with linear, cubic, quadratic kernel and KNN are used. Performances of the classifiers are graded by computing the True Positive Rate (TPR), False Positive Rate (FPR) and Equal Error Rate (EER) as discussed in Sect 4.

4 Experimentation and Results

The multimodal biometric authentication is modeled using the face and finger vein cues from SDUMLA-HMT dataset. The face images in the dataset include 106 subjects with varied poses, illumination conditions and accessories. The finger vein database contains left as well as right-hand index, middle, ring finger vein images for each subject. Proposed work incorporates 100 subjects for training. There are 6 finger vein images per subject (of size 240 * 320 * 3) which is scaled up to 102 images

Table 4 TPR, FPR & EER of the classifiers

Classifier	Kernel	TPR (%)	FPR (%)	EER (%)
Weighted KNN	–	96.7	3.5	3.4
Cosine KNN	–	97.3	2.7	2.7
SVM	Linear	97.3	0.14	1.42
SVM	Cubic	98.7	0.15	0.72
SVM	Quadratic	99.18	0.11	0.46

using image augmentation. The number of face images per subject used in this work is 85(of size 480 * 640 * 3). Each image is converted to grayscale, normalized and subsequently resized to a dimension of 64 * 64 which eventually will contribute to a swifter training.

The proposed work is evaluated on GPU enabled Google Collaboratory cloud environment with a 12 GB, 875 MHz GPU memory. Each of the two CNN models for face and finger vein images are trained for 50 epochs. The input to the CNN model is a 64 * 64 grayscale image and the output is a 100 * 1 feature vector, where each row represents the feature of the corresponding subject. Proposed work is carried out using 100 subjects with 102 finger vein and 84 face images per subject. The dimension of the feature vector is 100 * 100 * 102 and 100 * 100 * 84 for finger vein and face traits respectively. The feature vector from both the modalities are fused at feature level (concatenated vertically) and the size of the resulting feature space is (10,000 * 186) 18,60,000. The feature space is then labeled accordingly and separated into testing and training set by cross-validation. This work evaluates the model performance incorporating various classifiers like SVM with linear, cubic and quadratic kernel, weighted KNN and cosine KNN. The true positive rate, equal error rate and false positive rate of the various classifiers are as shown in Table 4. From the table, it is observed that SVM with quadratic kernel has outperformed the rest of the classifiers with a TPR, FPR and EER of 99.18%, 0.11% and 0.46% respectively.

The works in [13, 19] include a multitude of preprocessing and processing steps for feature extraction. The extracted feature is then given to neural network classifiers for classification. Hence, the overall procedure in the work is computationally intensive and time-consuming whereas our work requires minimal preprocessing and uses a single CNN structure for feature extraction and classification which has a less computational requirement and furnishes EER and accuracy better than the mentioned state-of-the-art methods. In [21, 22] deep convolution neural network architecture like VGG-16, VGG-19, Unet and dense net were used which require abundant computational power, although these models are the de-facto standards when it comes to image processing they furnished accuracies which were less compared to the accuracy achieved in the proposed work. In terms of accuracy and EER, the proposed model also outsmarted the accuracies and EER furnished in [9, 11, 15].

5 Conclusion

This work attests the efficacy of reduced features in achieving highly secured multimodal biometric authentication system. Incorporating deep layered CNN structure in state-of-the-art work [13, 16, 19, 21, 22] enforces exhaustive computational model. Inclusion of reduced feature set diminishes the aforementioned obligation in the present work.

Comprehensive experimentations are carried out to reveal the perks of multimodal biometric authentication system. The experiments are conducted employing finger vein and face images from the benchmark SDUMLA_HMT multimodal database. 18,60,000 fused features vector is obtained by training the CNN model with 18,600 images (i.e., 100 subjects 102 finger vein images/subject and 84 face images/subject). The feature vector is fed to the SVM quadratic classifier. The classifier attains a TPR of 99.18% and an EER of 0.46%.

The result demonstrates that the model is invariant to illumination, pose changes and accessories for face images and transformation invariant on finger vein images. This work also explores the fine-tuning of hyperparameters like the selection of minibatch size and addition of dropouts. The work evidences the reduction in the learning time and escalated efficiency. The SVM classifier with quadratic kernel records highest TPR of 99.18%, outperforming the state-of-the-art authentication techniques.

References

1. Jain AK, Ross A (2004) Multibiometric systems. Commun ACM 47:34–40
2. Fernandez FA (2008) Biometric sample quality and its application to multimodal authentication systems. Ph.D. thesis, Universidad Politecnica de Madrid (UPM)
3. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
4. Fischer A, Igel C (2014) Training restricted Boltzmann machines: an introduction. Pattern Recogn 47(1):25–39
5. Al-Waisy AS, Qahwaji R, Ipson S, Al-Fahdawi S, Nagem TAM. A multi-biometric iris recognition system based on a deep learning approach.
6. Yu D (2013) Deep learning methods and applications. Signal Process 28(3):198–387
7. Al-Waisy AS et al (2017) A multimodal biometric system for personal identification based on deep learning approaches. In: 2017 seventh international conference on emerging security technologies (EST). IEEE, pp 163–168. <https://doi.org/10.1109/EST.2017.8090417>
8. Ding C, Tao D (2015) Robust face recognition via multimodal deep face representation. IEEE Trans Multimedia 17(11):2049–2058. <https://doi.org/10.1109/TMM.2015.2477042>
9. Veluchamy S, Karlmarx LR (2017) System for multimodal biometric recognition based on finger knuckle and finger vein using feature-level fusion and k-support vector machine classifier'. IET Biometrics 6(3):232–242. <https://doi.org/10.1049/iet-bmt.2016.0112>
10. Radzi F, Khalil-Hani M, Bakhteri R (2016) Finger vein biometric identification using convolutional neural network. Turk J Electr Eng Comput Sci 24:1863–1878. <https://doi.org/10.3906/elk-1311-43>
11. Panasiuk P, Szymkowski M, Marcin D (2016) A multimodal biometric user identification system based on keystroke dynamics and mouse movements. In: IFIP international conference

- on computer information systems and industrial management. Springer, Cham, pp 672–681. <https://doi.org/10.1007/978-3-319-45378-1>
- 12. Li X, Huang D, Wang Y (2016) Comparative study of deep learning methods on dorsal hand vein recognition. In: Lecture notes in Chinese conference on biometric recognition. Springer, pp 296–306
 - 13. Belean B, Streza M, Crisan S, Emerich S (2017) Dorsal hand vein pattern analysis and neural networks for biometric authentication. *Stud Inf Control* 26(3):305–314. ISSN 1220-1766
 - 14. Talreja, V., Valenti M, Nasrabadi NM (2017) Multibiometric secure system based on deep learning. <https://doi.org/10.1109/GlobalSIP.2017.8308652>
 - 15. Xie C, Kumar A (2017) Finger vein identification using convolutional neural network and supervised discrete hashing. https://doi.org/10.1007/978-3-319-61657-5_5
 - 16. Hong H-G, Lee M-B, Park K-R (2017) Convolutional neural network-based finger vein recognition using image sensors. *Sensors* 17(6):1–21
 - 17. Al-Waisy AS, Qahwaji R, Ipson S et al (2018) A Multibiometric iris recognition system based on a deep learning approach. *Pattern Anal Appl* 21:783–802
 - 18. Sayed M (2018) Performance of convolutional neural networks for human identification by gait recognition. *J Artif Intell* 11:30–38. <https://doi.org/10.3923/jai.2018.30.38>
 - 19. Das R, Piciucco E, Maiorana E, Campisi P (2018) Convolutional neural network for finger-vein-based biometric identification. *IEEE Tran Inf Forensics Secur* 1–1
 - 20. Alay N, Al-BAity HH (2019) A multimodal biometric system for personal verification based on different level fusion of iris and face traits. *Biosci Biotech Res Commun* 12(3)
 - 21. Song J, Kim W, Park K (2019) Finger-vein recognition based on deep densenet using composite image. *IEEE Access* 1–1. <https://doi.org/10.1109/ACCESS.2019.2918503>
 - 22. Jalilian E, Uhl A. Hand book on vascular biometrics chapter 8 improved CNN-segmentation-based finger vein recognition using automatically generated and fused training labels
 - 23. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
 - 24. Bengio Y (2012) Practical recommendations for gradient-based training of deep architectures. *Neural networks: tricks of the trade*
 - 25. Kingma DP, Ba J (2014)“Adam: a method for stochastic optimization. CoRR abs/1412.6980
 - 26. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint [arXiv: 1600.04747](https://arxiv.org/abs/1600.04747)
 - 27. Shandong University machine learning and applications—Homologous multi-modal traits (SDUMLA-HMT), Shandong University, China. <http://mla.sdu.edu.cn/sdumla-hmt.html>

Malware Classification and Defence Against Adversarial Attacks



Aayush Kamath, Vrinda Bhatu, Tejas Paranjape, and Rupali Sawant

Abstract In today's world, new kinds of malicious codes are being created every day. This malware have the potential to compromise our systems and cause tremendous loss, may it be hardware based, or data oriented. We felt that there was a need to design a robust mechanism that can defend our systems against such malicious attacks. Traditional methods can't cope up with new kinds of malware. Machine learning would not only help in detecting the known kinds of malware, but also in identifying new kinds of malware. So, we implemented various machine learning techniques to make accurate detection of malware. We proposed the use of a hybrid classifier for further improving the accuracy of malware classification in comparison to the existing machine learning techniques. Although machine learning does provide a solution, it is still vulnerable to adversarial attacks. Hence, we created adversarial examples and analyzed their impact on machine learning classifiers. Then, as a defence method, we implemented adversarial training where the machine learning classifiers were trained on the adversarial examples along with the original samples. Adversarial training enabled the classifiers to become robust against potential adversarial attacks.

Keywords Malware · Adversarial training · Machine learning · Classification · Adversarial attacks · Neural networks

1 Introduction

When computers were invented, it was a very difficult task to breach the security of the computer due to the lack of networking. After the advent of the internet, people began

A. Kamath (✉) · V. Bhatu · T. Paranjape · R. Sawant

Department of Information Technology, Sardar Patel Institute of Technology, Mumbai, India
e-mail: ayush.kamath@spit.ac.in

V. Bhatu
e-mail: vrinda.bhatu@spit.ac.in

R. Sawant
e-mail: rupali_sawant@spit.ac.in

to play with concepts like viruses and worms to infect computers remotely. However, in those days, such malware was not very commonly found. Furthermore, after the computer boom of the 2000s, the number of people using personal machines grew exponentially. Unfortunately, so did the number of malware trying to hack into these machines to do various malicious tasks. Moreover, if we take a look at today's online landscape, we can easily see a number of malware trying to infect our systems, held out by many security systems in place. These security systems not only use hashes to verify if software is malware or not but also are starting to use machine learning models to classify them. These machine learning models give us the probability of software being malicious, and if it is above a certain threshold, then the user is notified of the same.

Unfortunately, as machine learning started to weed out malicious code, hackers came up with different technology. By using adversarial examples, hackers ensured that the model was trained in such a way that the number of false positives would be increased. This essentially confuses the model, and it allows harmful code to be executed by the system. This is the problem that we have solved in this research paper.

Firstly, we have compiled a dataset that comprises numerous features of 5000 data samples. These samples were then fed into a machine learning model using several existing algorithms and found out the best of them. Further, we have identified the features that prominently impact the classification of the software. Using these features and the model, we created a hybrid classifier that was able to classify malware samples with greater accuracy as compared to existing algorithms.

Lastly, we fed the model adversarial examples. This caused the accuracy to fall, as predicted. However, after training the model against such adversarial examples, our model became better at dealing with potential obfuscations in the features on which it was trained.

This paper is structured such that in Sect. 2, we have reviewed the existing research related to our topic in order to get accustomed to the domain in question. In Sect. 3, we began with the dataset description in Sect. 3.1. In Sect. 3.2, we proposed a solution where we looked at the existing machine learning models used for malware classification and we chose the ones which are commonly used, i.e., Naive Bayes (NB), Random Forest (RF), K-Nearest Neighbors (KNN), and selected the best two based on the results obtained after testing on our dataset. Later, for further increasing the accuracy we implemented a hybrid classifier using deep neural networks along with RF and KNN. In Sect. 3.3, for reducing the vulnerabilities of machine learning models we trained our models by adversarial examples thus making it robust against adversarial attacks. In Sect. 3.4, we analyzed the results obtained through the implementation of our proposed solution. In Sect. 3.5, we summarized our learnings and identified a few areas where future work can be conducted.

2 Literature Review

For this research, we looked up the existing literature in this area. The following summarizes our findings in a concise format. In Rathore et al. [1], the creation of a feature vector space is mentioned as a critical aspect of any machine learning algorithm. They have used a static analysis approach and found out that the Random Forest technique gave the best accuracy. Through Liu et al. [2] and Aslan and Samet [3], we were provided a comprehensive overview of various approaches for malware detection such as signature-based, machine learning based, deep learning based and cloud based to name a few. They shed light on the features used, models and algorithms used and the results achieved by various kinds of malware detection approaches.

In Sayadi et al. [4], we learned of a clear trade-off between the performance of standard ML classifiers and the number of HPCs (Hardware Performance Counters) available in modern microprocessors. We have reached an optimal level of epochs considering the above research. Teenu et al. [5] and Huang et al. [6] have given an in-depth explanation about gradient-based, white box, black box, score based, and other attacks, conducted on machine learning models. The conclusion through these papers is that training against adversarial examples is essential to improve the accuracy of the classifiers. Firdausi et al. [7] talks about different classifiers and compares their accuracies while Grosse et al. [8] extends this research and talks about the effect of adversarial training upon the classifiers. Paper [9] and paper [10], Ibitoye et al. and Zhang et al. respectively have provided deep insights on malware in specific domains. The former provides information about the network security aspects of malware and its application in adversarial training, while the latter gives us valuable information regarding a particular virus, transmitted through pdf files. For a better understanding, we recommend reading paper [11], Serban et al. where the entire construction of an adversarial example has been explained in detail. This paper gave us a very sound understanding of what adversarial training is all about. Finally, Ranveer et al. [12] has covered static, dynamic, and hybrid methods of feature extraction for malware.

3 Proposed Solution

See Fig. 1.

3.1 Dataset Description

The dataset we used for training our machine learning models is the ‘top 1000 PE (Portable Executable) imports dataset’ [13]. Obtained through static analysis, it contains the top 1000 imported functions which were extracted from the ‘PE imports’ elements from the Cuckoo Sandbox Reports. With a total of around 47,000

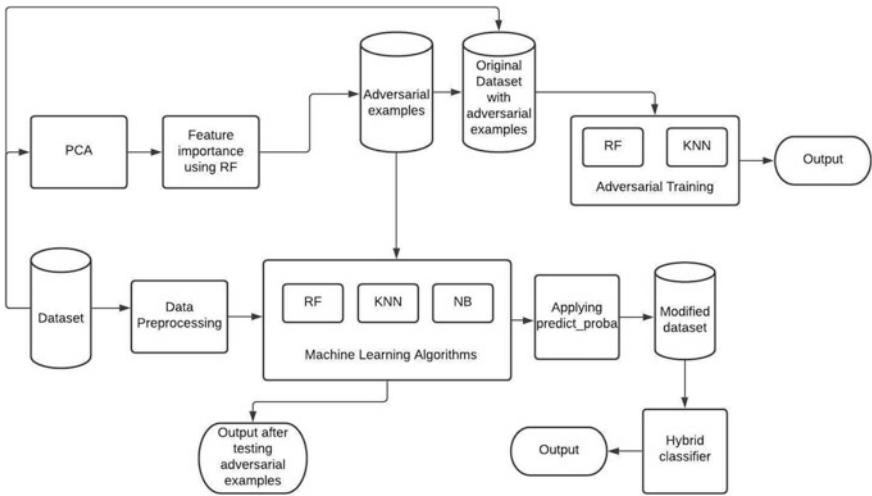


Fig. 1 Block diagram

samples, 45,000 of those belonged to the malware class, and close to 2000 were benign. Since the ratio was quite skewed, there was a risk of overfitting on the part of machine learning classifiers. Hence, we decided on using 5000 samples in total for training and testing with the bifurcation being around 3000 malware and 2000 benign samples. Each row represented a sample file. For every sample file, 1002 column values existed with the first column being its MD5 Hash value and the last column being its label, i.e., whether it is malware or benign. The rest of the 1000 columns were the most common import functions invoked by all the sample files that were run in the cuckoo sandbox environment [13], as mentioned earlier. Each of these columns had either 0 or 1 as their value. ‘0’ signified that the particular function wasn’t invoked by the given sample file. Whereas, ‘1’ indicated that the particular function was invoked by the given sample file.

3.2 Machine Learning Classifiers

We began our modeling phase by identifying 3 different machine learning classifiers for our malware dataset. They are Random Forest, K-Nearest Neighbors, and Naive Bayes. Each model was trained and subsequently tested on our malware dataset. Accuracy parameters for each of the classifiers were analyzed. Based on the results, we observed that Random Forest and KNN outperformed Naive Bayes significantly in terms of accuracy. To further bolster the accuracy in order to better classify malware samples we decided on a hybrid classifier that would leverage the benefits of both KNN and Random Forest.

A deep neural network was used alongside KNN and Random Forest for building the hybrid classifier. We made use of the predict_proba function where for each sample, the probability value of labels 0 and 1 for KNN as well as Random Forest was obtained. So if for KNN, the predict_proba values were 0.92 and 0.08 for labels 0 and 1 respectively, it would mean that KNN is 92% certain that the given sample is benign. With 2 values each for KNN and Random Forest, we had 4 new features that we could use. These 4 output variables were added to the dataset of the 1000 PE import variables. These 1004 features were fed to a deep neural network with 7 layers in total. The last layer being the output layer. We used stochastic gradient descent as our optimizer and varied the learning rate, momentum, and decay parameters of our neural network to arrive at the most optimum state of our hybrid classifier. We trained the neural network over 200 epochs and validated it with our test data.

3.3 Adversarial Examples and Adversarial Training

Adversarial Examples are generated by attackers in order to fool a machine learning classifier. They are small inputs provided to machine learning models that are intentionally designed by the attacker to induce an error on the part of the model. They are essentially features created out of small perturbations to the actual input features. In order to make our machine learning models robust against adversarial attacks, we decided to generate certain adversarial examples and as a defence technique make use of Adversarial Training.

To make changes in a dataset with a size of 1000 features would have been a huge task. Hence, we applied Principal Component Analysis (PCA) to our dataset to reduce the dimensionality of the dataset. By applying PCA, we made sure that all necessary data is retained, but at the same time, the dimensionality is reduced. After reducing the features to 100, we calculated the feature importance score for all features. Based on the scores, we chose the 10 most important features out of 100 and created adversarial examples by changing the sign for the values of those features for 20% of the samples. As expected, accuracy for Random Forest and KNN drastically reduced when tested with this data. As a defence against adversarial attacks, we decided to use Adversarial Training. In adversarial training, apart from the original data, the obfuscated data, i.e., the adversarial examples are also used for the training phase. This is done in order to make the machine learning model robust against perturbations in the data.

Hence, the adversarial examples were then merged with the original data and adversarial training was performed for the defence part. Doing so, helped the Random Forest and KNN classifiers to reach close to their initial accuracies. Random Forest reached an accuracy of 89.73% and KNN reached an accuracy of 89.42%. Hence, the adversarial training proved to be an effective defence method as Random Forest and KNN regained their accuracy for the most part.

3.4 Results

We found out that our hybrid classifier outperformed Random Forest, which had the highest accuracy so far, by about 0.6%. When the deep neural network was used without any parameters related to KNN and Random Forest, it performed better than KNN but still gave a lesser accuracy than Random Forest. The highest accuracy was obtained by our hybrid classifier (Tables 1 and 2).

The graph in Fig. 2 shows the relationship between the accuracy of the hybrid classifier and the loss function over 200 epochs. With 94.05% our hybrid classifier has shown substantial improvement over existing classifiers.

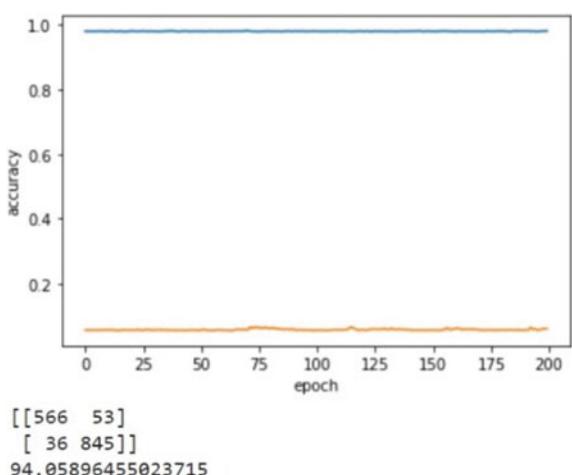
Table 1 Comparative results (Machine Learning Classifiers)

Accuracy (%)	Model
93.47	Random forest
89.59	K-nearest neighbors
72.29	Naive Bayes
91.23	Deep neural network
94.05	Hybrid classifier (DNN with RF + KNN)

Table 2 Comparative results (Adversarial examples and adversarial training)

Accuracy with PCA (%)	Accuracy with adversarial examples (%)	Accuracy after adversarial training	Model
93.03	60.76	89.73	Random forest
90.95	56.74	89.42	K-nearest neighbors

Fig. 2 Accuracy after using hybrid classifier



The accuracy of Random Forest went from 93.03 to 60.76% and KNN went from 90.95 to 56.74% after subjecting them to adversarial examples. The varying accuracy (93.03 and 90.95% for RF and KNN as compared to 93.47 and 89.59% earlier) is due to the application of PCA. As a defence method, adversarial training was used and the machine learning classifiers recovered in terms of their classification accuracy with 89.73 and 89.42% for Random Forest and K-Nearest Neighbors respectively.

3.5 Conclusion and Future Work

Combining the attributes of K-Nearest Neighbors and Random Forest along with a deep neural network we were able to create a hybrid classifier that gave us an accuracy of 94.05%, which is close to 0.6% better than the next best classifier, Random Forest with an accuracy of 93.47%. Without the predict_proba parameters, the neural network only returned an accuracy in the range of 91–92%. Hence, it is pretty clear that the hybrid classifier implemented by us helps in better classification of malware samples. We also made our classifiers robust against adversarial attacks by training it against adversarial examples. Adversarial training made sure that there isn't a very large drop in accuracy once adversarial examples are introduced to the machine learning classifiers.

For future research, different methods such as Generative Adversarial Networks could be explored for generating adversarial examples. Exploring a new method for creating adversarial examples would provide deeper insight into the ways in which obfuscations to features affect the machine learning classifiers. Also, new kinds of classification algorithms could be looked at, apart from the ones implemented, in order to further boost the accuracy of malware classification.

References

1. Rathore H, Agarwal S, Sahay SK, Sewak M (2018) Malware detection using machine learning and deep learning. In: Mondal A, Gupta H, Srivastava J, Reddy P, Somayajulu D (eds) Big data analytics. BDA 2018. Lecture notes in computer science, vol 11297. Springer, Cham
2. Liu L, Wang B, Yu B et al (2017) Automatic malware classification and new malware detection using machine learning. *Frontiers Inf Technol Electron Eng* 18:1336–1347
3. Aslan OA, Samet R (2020) A comprehensive review on malware detection approaches. In: IEEE Access, vol 8, pp 6249–6271
4. H. Sayadi, N. Patel, S. M. P.D., A. Sasan, S. Rafatirad and H. Homayoun, "Ensemble Learning for Effective Run-Time Hardware-Based Malware Detection: A Comprehensive Analysis and Classification," 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, CA, 2018, pp. 1-6
5. John, Kallivayalil TT, Tony (2019) Adversarial attacks and defenses in malware detection classifiers. <https://doi.org/10.4018/978-1-5225-8407-0.ch007>
6. Huang Y, Verma U, Fralick C, Infantec-Lopez G, Kumar B, Woodward C (2019) Malware evasion attack and defense. In: 2019 49th Annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W), Portland, OR, USA, pp 34–38

7. Firdausi I, Lim C, Erwin A, Nugroho AS (2010) Analysis of machine learning techniques used in behavior-based malware detection. In: 2010 Second international conference on advances in computing, control, and telecommunication technologies, Jakarta, pp 201–203
8. Grosse K, Papernot N, Manoharan P, Backes M, McDaniel P (2017) Adversarial Examples for malware detection. In: Foley S, Gollmann D, Snekkens E (eds) Computer security—ESORICS 2017. ESORICS 2017. Lecture notes in computer science, vol 10493. Springer, Cham. https://doi.org/10.1007/978-3-319-66399-9_4
9. Ibitoye, Abou-Khamis O, Matrawy R, Ashraf Shafiq M (2019) The threat of adversarial attacks on machine learning in network security—A survey
10. Zhang J (2018) MLPdf: an effective machine learning based approach for PDF malware detection. *Secur Cryptogr*
11. Serban AC, Poll E, Visser J (2019) Adversarial examples—A complete characterisation of the phenomenon’, 17th Feb 2019
12. Ranveer S, Hiray S (2015) Comparative analysis of feature extraction methods of malware detection. *Int J Comput Appl* 120(5)
13. Oliveira A (2019) Malware analysis datasets: top-1000 PE imports. IEEE Dataport. <https://doi.org/10.21227/004e-v304>

A Novel Ensemble Machine Learning Model for Prediction of *Zika Virus* T-Cell Epitopes



Syed Nisar Hussain Bukhari, Amit Jain, and Ehtishamul Haq

Abstract Zika virus belongs to the genus Flavivirus and causes Zika fever in humans. World Health Organization (WHO) declared its outbreak as Public Health Emergency of International Concern in 2016. Currently, there is no approved vaccine for clinical use to combat the *Zika Virus* infection and its epidemic. The *in-silico* approach to T-cell epitope prediction of Zika virus is useful to save biologist's time and efforts for vaccine development. The authors have proposed a novel ensemble machine learning model to predict Zika virus T-cell epitopes using physicochemical properties of amino acids. The model has been designed by fusing the top two performing classifiers from among the six machine learning classifiers (base classifiers). The peptide sequences consisting of experimentally determined T-cell epitopes and non-epitopes of Zika virus were collected from Immune Epitope Database and Analysis Resource (IEDB). The authors have verified the model through Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). The proposed model achieved an accuracy of 92.11% on the test dataset. We validated the model using a separate validation set and the model achieved an accuracy of 93.47%, Gini of 0.968, AUC of 0.996, the sensitivity of 0.964, and specificity of 0.961. To check the robustness of the model, a technique called repeated k-fold cross-validation was used, and an average accuracy of 91.7% was recorded. The predicted epitopes would undoubtedly play a critical role in designing vaccines to save lives across the globe from this deadly virus.

Keywords Ensemble · Machine learning · TOPSIS · Zika virus · Stacking · Epitope · T-cell · Immunoinformatics

S. N. H. Bukhari (✉) · A. Jain

University Institute of Computing, Chandigarh University, N95, Chandigarh-Ludhiana Highway, Mohali, Punjab 140413, India

E. Haq

Department of Biotechnology, University of Kashmir, Srinagar, Jammu and Kashmir 190006, India

e-mail: haq@uok.edu.in

Acronyms Used

WHO	World Health Organization
IEDB	Immune Epitope Database and Analysis Resource
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
ZIKV	Zika virus

1 Introduction

ZIKV is an enveloped virus that belongs to the genus Flavivirus and family Flaviviridae. It is almost similar to yellow, dengue fever, and West Nile virus, which is spread through a bite by an infected mosquito belonging to Aedes species (*Aedes aegypti* and *Aedes albopictus*) [1]. It was identified in Rhesus monkeys first in Zika Forest in 1947 in Uganda and 1952 in humans in the Republic of Tanzania and Uganda. Its outbreaks have also been recorded in Asia, Africa, the Pacific, and the Americas. In 2016 in Brazil, which is considered as the hotspot of an epidemic, around 216,207 cases were reported, and 8604 babies with malformations were born. In February of 2016, the outbreak was declared as Public Health Emergency of International Concern. To date, the shreds of evidence of ZIKV disease have been reported from 86 countries cum territories [2]. The majority of the people infected with ZIKV are asymptomatic. Generally, symptoms are mild fever, conjunctivitis, joint and muscle pain, malaise, and headache, which lasts for 2–7 days, and an incubation period of 3–14 days has been estimated. The ZIKV infection can pass to the fetus of a pregnant woman and is the main reason for microcephaly, other congenital abnormalities in the developing fetus and the newborn.

ZIKV is a single-stranded, non-segmented positive-sense RNA virus. It has a genome of 10.7 kb, which can be translated directly into one long protein. The protein can encode three structured proteins (capsid–C, envelope–E, membrane protein–M) as well as seven non-structured proteins (NS1, NS2A, NS2B, NS3, NS4A, NS4B, and NS5) [3, 4]. The principal and primary antigenic determinant is envelope (E) glycoprotein which mediates the fusion and binding at the time of virus entry [5–8]. Therefore, this envelope-E glycoprotein acts as a primary research target to develop antiviral therapeutics and vaccine candidates. The high similarity between the ZIKV and other Flaviviruses has facilitated vaccine development research a lot [9–12]. Although ZIKV infection is a severe and fatal disease, currently there is no effective vaccine and a specific medicine to combat the ZIKV infection and its epidemic. However, people need to follow certain precautions to prevent this infection such as taking enough water to stop dehydration, use of paracetamol or analgesics, acetaminophen, and taking rest [12–14]. Nevertheless, these measures are not enough to prevent this infectious disease. So to have an effective and viable vaccine

against the different strains of ZIKV, it is essential to select the number of epitopes that are antigenic as epitope-based vaccines are considered as a safe platform for the development of vaccines [15–17].

1.1 Background

The immunoinformatics approach has emerged as a promising field for epitope prediction [15]. Highly immunogenic and conservative T-cell epitopes on virus antigens act as potential targets for vaccine [16]. It is pertinent to mention here that most of the existing methods like NetMHC-a web-based tool (versions: 2.2, 2.3, 3.0, 4.0) which are based on support vector machine (SVM) and neural network classifiers only provide binding capacity estimation of a peptide (not direct prediction as a discrete-valued output, i.e., 0 or 1) [17–23]. However, the CTLpred server predicts epitopes directly, but it takes peptides of length up to 9-mers only. Here in this study, an ensemble model has been proposed which like the CTLpred server predicts directly whether a peptide sequence is ZIKV T-cell Epitope or not and it also takes variable-length peptides as shown in Fig. 1. The classifiers used for the proposed ensemble model have been trained on the physicochemical properties of amino acids.

The rest of the paper is organized as follows: Material and methods, feature extraction, feature selection, partitioning of data, training and selection of base classifiers

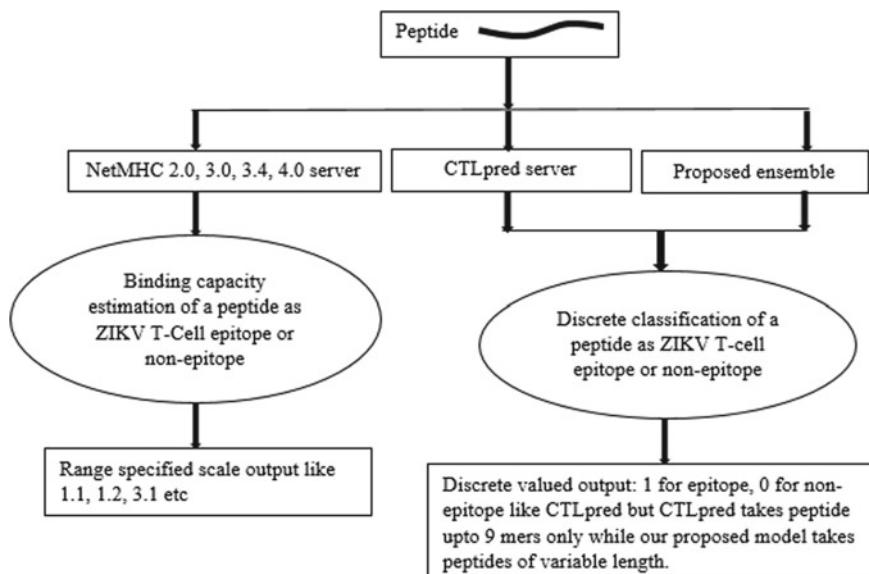


Fig. 1 Prediction outputs from existing methods and the proposed ensemble model

are detailed in Sect. 2. The proposed ensemble model is explained in Sect. 2.6. Results including model evaluation on various parameters, model validation and model comparison are discussed in Sect. 3. Discussion is done in Sect. 4. Conclusion and future work are described in Sect. 5.

2 Materials and Methods

The experimentally determined ZIKV T-cell epitopes were collected from the IEDB [24]. Dataset consists of 3534 peptide sequences (epitopes and non-epitopes) out of which 1757 sequences are non-epitopes and the rest are epitopes. Fifteen (15) redundant epitopes were discarded. The epitope length is in the range of 8 to 30-mers and non-epitopes 6 to 28-mers. The glimpse and structure of the dataset are shown in Table 1 where column SL denotes sequence length and CL denotes class, i.e., 1-epitope and 0-non-epitope.

2.1 Feature Extraction

Feature extraction is an important step to improve performance and enhance model effectiveness. In the current study, physicochemical properties of amino acids have been used to extract features from peptide sequences. The features have been extracted using R packages *peptider* and *peptides* [25]. Table 2 presents physicochemical properties used in the current study, the necessary R packages, and functions inside the package along with the notations used in the study.

2.2 Feature Selection

Choosing the right set of features and feeding them to the classifier is an essential step in machine learning problems to expect great results. *Caret* package in R offers a function called *varImp()* which was used to calculate the feature importance of

Table 1 Sample dataset of ZIKV T-cell epitopes and non-epitopes

Peptide sequence	SL	F1	F2	-	F10	F11	CL
GLDFSDLYY	9	65.2	7.55	-	-0.173	8	1
AAMLRIINARKE	12	23.7	-1.93		-0.345	3	0
RFLEFEALGF	10	16.65	4.01		-0.875	8	1
RPRVCTKEEF	10	34.64	7.34		-0.325	3	0
TEQRKTFVEL	10	78.67	7.05		-0.046	4	0

Table 2 Physicochemical properties of amino acids used in the current study

S. No.	Property name	Package	Function	Notation
01	Aliphatic index [26]	Peptides	aIndex(seq)	F1
02	Potential Protein Interaction (Boman) Index [27]	Peptides	boman(seq)	F2
03	Instability index of a protein sequence [28]	Peptider	instaIndex(seq)	F3
04	Probability of detection of a peptide sequence	Peptides	ppeptide (x, libscheme, N)	F4
05	Hydrophobic moment 1. Rotational angle a-helix = 100 2. Rotational angle b-sheet = 160 [29]	Peptides	hmoment (seq, angle)	F5_1, F5_2
06	Molecular weight 1. Monoisotopic = FALSE 2. Monoisotopic = True [30]	Peptides	mw(seq,monoisotopic)	F6_1, F6_2
07	Theoretical net charge at 9 pKa scales	Peptides	charge	F7
08	Hydrophobicity index	Peptides	Hydrophobicity	F8
09	Isoelectric point	Peptides	pI	F9
10	Kidera factors	Peptides	kideraFactors	F10
11	Amino acid composition	Peptides	aaComp	F11

all features. In the present study, *varImp ()* function was used with *random forest algorithm* which returned weights or importance score corresponding to the entire feature set. The scores represent *Mean Decrease in Gini* by importance measure and convey how much does a feature contributes to homogeneity in data. In the present study, the complete dataset has 13 features, and after applying feature selection, only ten top-ranked features returned by *VarImp ()* function were selected to train classifiers. Features with their importance score are shown in Table 3, their plot in Fig. 2 and the formula for classifier training for all six (06) classifiers in Eq. 1.

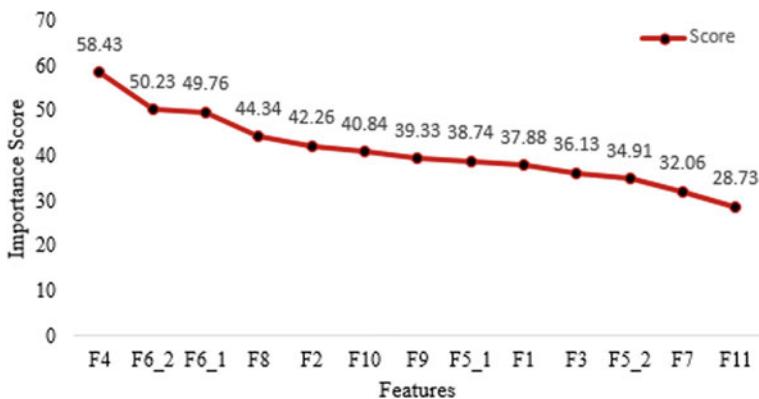
$$\text{CL} \sim f(F4, F6_2, F6_1, F8, F2, F10, F9, F5_1, F1, F3) \quad (1)$$

2.3 Data Partitioning

In this study, we divided the dataset into 80:20 ratios for training and testing set, respectively. The validation set, which is 20% of the training set was kept for validation purposes to validate and check its accuracy. Figure 3 shows the data portioning process used in the current study.

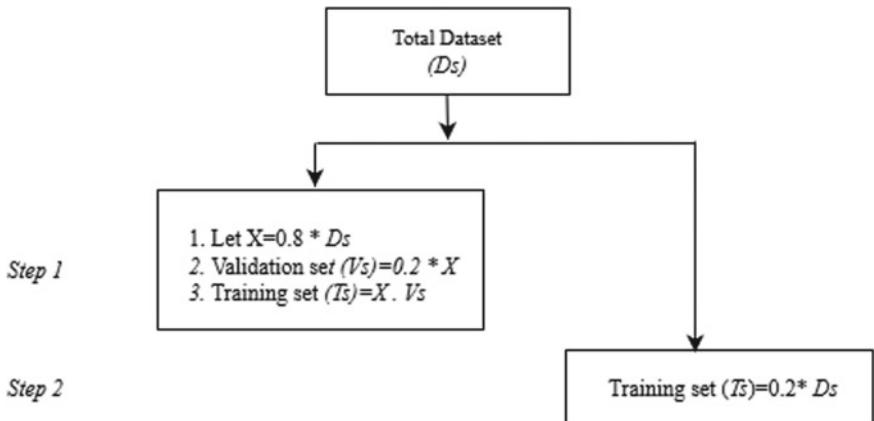
Table 3 Feature importance

Feature	Score
F4	58.43
F6_2	50.23
F6_1	49.76
F8	44.34
F2	42.26
F10	40.84
F9	39.33
F5_1	38.74
F1	37.88
F3	36.13
F5_2	34.91
F7	32.06
F11	28.73

**Fig. 2** Feature importance plot

2.4 Training of Base Classifiers

Machine learning, a subset of artificial intelligence, is an area of computational science focusing on analyzing and interpreting patterns in data to enable learning, reasoning, and decision making outside of human interaction. A machine learning classifier is a mathematical expression representing data in the context of a problem. Table 4 shows machine learning classifiers we have used in the current study along with their R packages, and functions with essential tuning parameters. We trained all the classifiers individually, and Table 5 shows their Gini, AUC, sensitivity, specificity, and accuracy scores on the test set.

**Fig.3** Data partitioning process**Table 4** Machine learning classifiers used in the current study

Classifier	R package	Tuning parameters
Decision Trees [31]	Package rpart	params = list(split = "information"), control = rpart.control(usesurrogate = 0, maxsurrogate = 0))
Neural Network [32]	nnet	size = 10, linout = TRUE, skip = TRUE, MaxNWts = 10,000, trace = FALSE, maxit = 100
Support Vector Machines (SVM) [33]	Kernlab (ksvm)	kernel = "rbfdot", prob.model = TRUE
adaBoost [34]	ada	Control = rpart: rpart. Control (maxdepth = 30, cp = 0.01, minsplit = 20, xval = 10), iter = 50)
Random Forest [35]	randomForest	ntree = 500, mtry = 2
Linear Model [36]	nnet	trace = FALSE, maxit = 1000

Table 5 Performance evaluation parameters of all trained classifiers

Model	Gini	AUC	Sens	Spec	Accuracy
Decision Tree	0.685	0.842	0.861	0.795	82.84
Linear Model	0.521	0.761	0.701	0.643	67.30
Neural Network	0.501	0.751	0.672	0.653	66.26
SVM	0.830	0.915	0.813	0.862	83.70
Random Forest	0.928	0.964	0.859	0.928	89.29
AdaBoost	0.927	0.963	0.904	0.890	89.67

2.5 Selection of Classifiers for the Proposed Ensemble Model

As can be seen, the top two performing classifiers are Random Forest and AdaBoost in terms of accuracy, i.e., 89.29% and 89.67%, respectively. Figures 4 and 5 show receiver operating characteristics (ROC) and receiver operating characteristic convex hull (ROCH) curves of random forest and AdaBoost, respectively.

To make sure that the selected top two performing classifiers are the best fit for our proposed ensemble model, we used a technique called TOPSIS. It is a decision analysis technique based on multi-criteria originally developed by [37]. R programming framework provides a package called *topsis* for this. For TOPSIS, three things are required: *Performance table*, *criteria weights*, and *criteria MinMax*. Table 6 is the *Performance table* of six trained classifiers for the *topsis* package. *Criteria weights* is a vector that contains weights for each criterion, and in the current study a uniform weight of 1 (default) has been set for this parameter.

Criteria MinMax is a vector that contains preference direction for each criterion with plus “+” sign indicating the criteria have to be maximized and minus “-” sign indicating the criteria have to be minimized. Here Gini, AUC, AUCH, accuracy criteria are to be maximized, and Minimum Cost-Weighted Error Rate (MWL), Minimum Error Rate (MER) criteria are to be minimized. As can be seen from Table 7, after applying TOPSIS, rank 1 and 2 are achieved by AdaBoost and Random

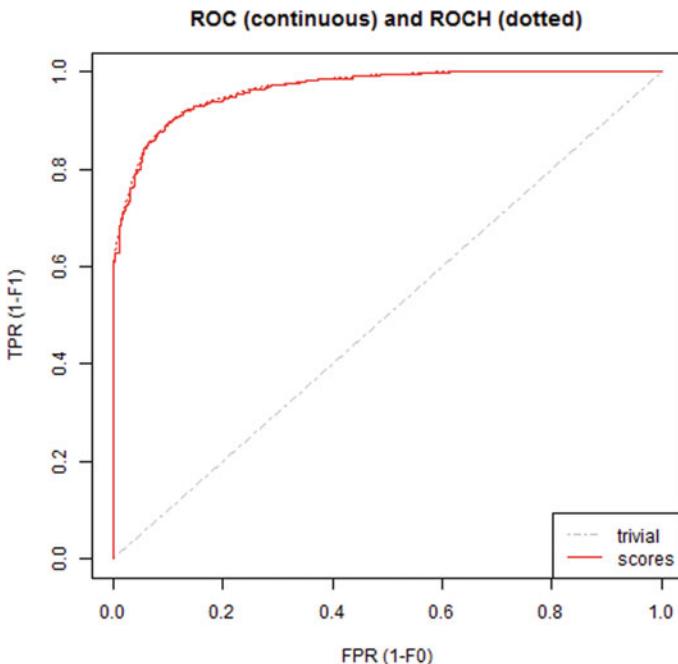


Fig. 4 Random forest ROC and ROCH curves

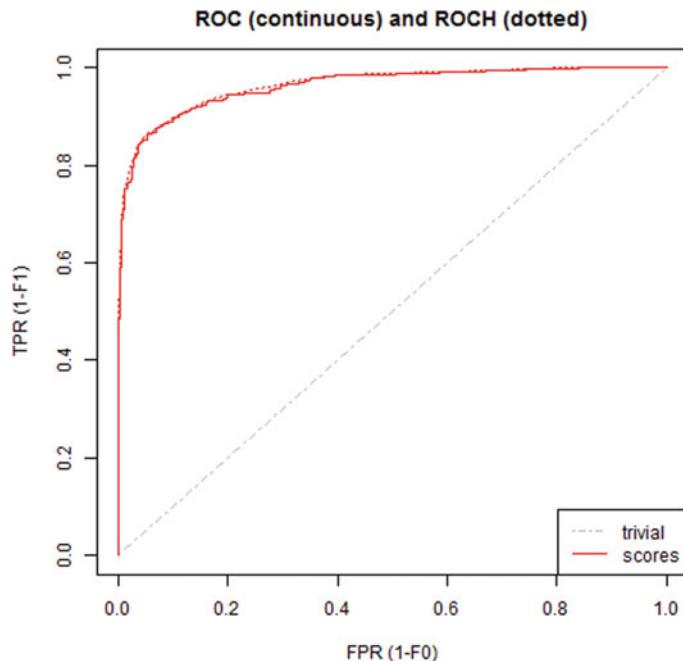


Fig. 5 AdaBoost ROC and ROCH curves

Table 6 Performance table of six trained classifiers for TOPSIS

Row Id	Model	Gini	AUC	AUCH	MER	MWL	Accuracy
1	Decision Tree	0.685	0.842	0.846	0.172	0.172	82.84
2	Linear Model	0.521	0.761	0.775	0.309	0.309	67.3
3	Neural Network	0.501	0.751	0.763	0.319	0.322	66.26
4	SVM	0.83	0.915	0.92	0.159	0.159	83.7
5	Random Forest	0.928	0.964	0.966	0.101	0.101	89.29
6	AdaBoost	0.927	0.963	0.966	0.095	0.094	89.67

Table 7 Performance table of six trained classifiers with TOPSIS score and rank

S. No.	Classifier	TOPSIS Score	TOPSIS Rank
1	Decision Tree	0.61901178	4
2	Linear Model	0.05065414	5
3	Neural Network	0.00000000	6
4	SVM	0.72387091	3
5	Random Forest	0.97391279	2
6	AdaBoost	0.99894588	1

Forest, respectively. Hence, after applying TOPSIS it is clear that Random Forest and AdaBoost are the top two performing classifiers and prime candidates for the proposed ensemble model.

2.6 Proposed Ensemble Model

Prediction in Machine Learning is achieved through a powerful technique called classification. There are several classification techniques with few having satisfactory prediction accuracy while others exhibit limited accuracy. This paper presents a novel machine learning-based ensemble model obtained by combining two well-performing classifiers obtained in Sect. 2.5 to classify a peptide as the ZIKV T-cell Epitope or non-epitope. To blend them, we used a stacking ensembling approach as shown in Fig. 6. The beauty of stacking ensemble is that it learns how to combine in better way predictions from several well-performing machine learning classifiers. The proposed ensemble approach consists of the following steps:

Step 1: When applying a stacking ensemble approach, predictions of selected well-performing models (from level-0 models) should have a low inter-model correlation (usually < 0.75). The inter-model prediction correlation was computed using *modelCor()* and *resamples()* function provided by the *caret* package of R. The two models were compared with *resamples()* and its output was given as an input to *modelCor()* whose output is inter-model prediction correlation between Random Forest and AdaBoost. We used the *Spearman correlation* method, and we found that there is a very low correlation between the two selected models.

Step 2: In this step, the model fitting was done by combining the two selected classifiers. A new dataset was formed consisting of predictions for random forest and AdaBoost and class variable from the test dataset. We used the *Stochastic*

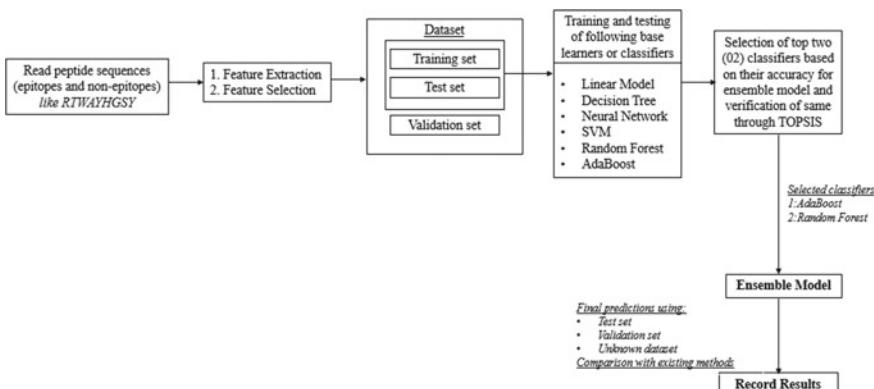


Fig. 6 Proposed ensemble mode

Table 8 Evaluation results on the testing set

Classifier	Gini	AUC	Sensitivity	Specificity	Accuracy
Random forest	0.929	0.943	0.855	0.910	89.29
AdaBoost	0.940	0.928	0.894	0.923	89.67
Proposed ensemble	0.954	0.973	0.891	0.932	92.11

Table 9 Evaluation results on the validation set

Classifier	Gini	AUC	Sensitivity	Specificity	Accuracy
Random forest	0.931	0.968	0.862	0.931	89.32
AdaBoost	0.947	0.987	0.915	0.943	89.78
Proposed ensemble	0.968	0.996	0.964	0.961	93.47

Gradient Boosting algorithm to fit the new model (Level-1 model or Metamodel) to relate these two predictions from two classifiers to the class variable.

Step 3: In this step, we used new model (proposed model) to make predictions on new samples. An accuracy of 92.11% was achieved when the proposed model was evaluated on a test set which is overall high than the accuracies of both Random Forest and AdaBoost (89.29% and 89.67%, respectively) as shown in Table 8.

Step 4: Using the testing set does not provide a good representation of out of sampled error. So the validation dataset was used which is 20% of the total training to assess and measure the accuracy of the proposed ensemble model. Here, predictions of Random Forest and AdaBoost were again created on the validation set. The data frame was created containing these two predictions and then predicted using the proposed ensemble model on predictions in the validation set. The results are shown in Table 9.

Step 5: The repeated k-fold cross-validation was used to check the model's robustness. The proposed model was also compared with existing systems using a new (unknown) dataset.

Step 6: Finally, result analysis was done to conclude that the proposed ensemble model is more accurate and robust compared to existing methods.

3 Results

3.1 Model Evaluation Results

The primary motive of model evaluation is to conclude how well the model has learnt from the data and what is the accuracy of the predictions that the proposed ensemble model produces. To analyze and evaluate the performance of individual classifiers and the proposed ensemble model, we have used the following parameters in the current study.

3.1.1 Area Under the Curve (AUC)

AUC is an area under the receiver operating characteristics (ROC) curve. A model is of good quality and better as compared to others if its AUC value is high. The value of AUC is between 0 and 1. A perfect model will have an AUC value equal to 1.

3.1.2 Gini Coefficient

Gini coefficient refers to inequality in distribution whose value is between 0 (equality) and 1 (inequality). The equation to calculate the Gini coefficient is:

$$\text{Gini} = 2 \times \text{AUC} - 1 \quad (2)$$

3.1.3 Accuracy

This measure tells us how correctly model predicts and is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Data}} \times 100 \quad (3)$$

3.1.4 Sensitivity

It is also called true positive rate, which is the proportion of the actual positives to correctly identified positives by a model and is calculated as:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

3.1.5 Specificity

It is also called a true negative rate and tells the ability of the model to identify negative results, which is calculated as:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}} + \text{FP} \quad (5)$$

where TN-*True negative*, TP-*True positive*, FP-*False Positive*, FN-*False negative*.

Overall, the proposed ensemble model resulting from combining classifiers shows higher accuracy than individual classifiers and there is an improvement in all the

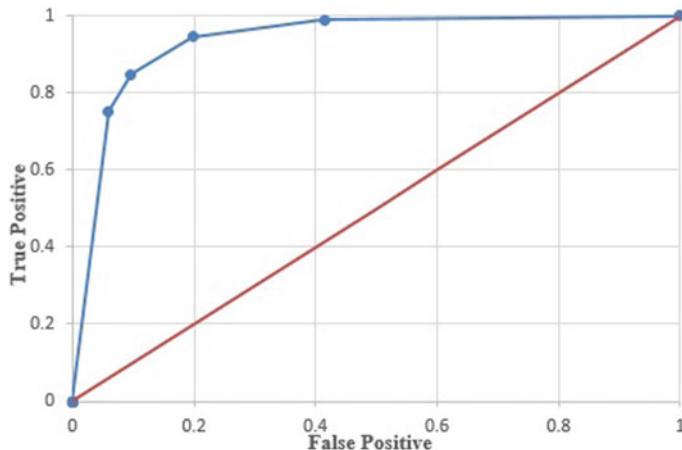


Fig. 7 ROC curve of the proposed ensemble model

evaluation parameters. Our model achieved an accuracy of 92.11% and 93.47% on the test set and validation set (which is 20% of the training set), respectively, as shown in Tables 8 and 9. So, this model can surely be used to predict novel ZIKV T-cell epitopes uniquely.

As shown in Table 8, on the validation set the proposed model achieved a Gini of 0.968, AUC of 0.996, a sensitivity of 0.964, and a specificity of 0.961 which is more than what has been achieved in the existing methods. The ROC curve of the proposed ensemble model is shown in Fig. 7.

3.2 Model Validation Results

During the model training process, problems like overfitting, biasness, and underfitting might occur. To address these issues and to check whether the proposed ensemble model is robust enough, we used a statistical method called repeated k-fold cross-validation. In the k-fold cross-validation technique, we divide the dataset into k subsets. For each subset is held out while the model is trained on all other subsets. This procedure is completed until accuracy is determined for each instance in the dataset. To increase the chances of occurrences of each portion in the test dataset, a repeated k-fold cross-validation technique is preferred, which involves repeating the cross-validation method multiple times and then recording the mean result of all folds from all the runs. In the current study, we created ten (10) folds of the dataset, executed each fold five (05) times and recorded an average accuracy of 91.7% as shown in Fig. 8.

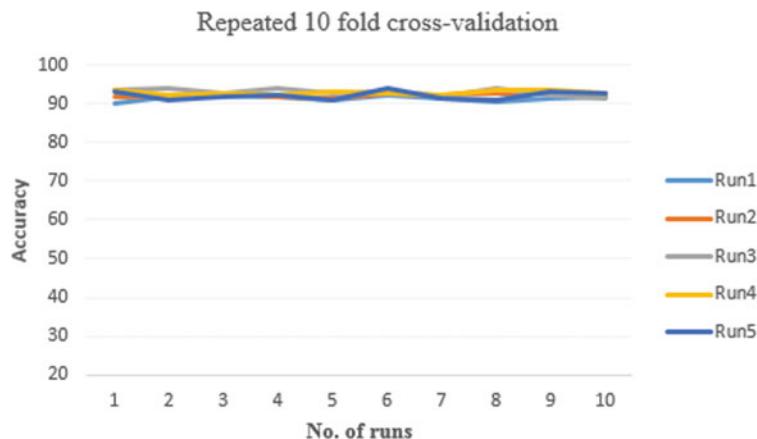


Fig. 8 Repeated tenfold cross-validation

3.3 Model Comparison with Existing Methods Using an Unknown Dataset

The proposed model was also compared with the existing system using an unknown dataset. In the current study, this unknown dataset consists of eight (08) T-cell epitopes of the ZIKV, which were collected from the literature [16, 38]. Five (05) non-epitopes were taken from the existing MHCBN4.0 server [39]. This unknown dataset consisting of thirteen (13) peptide sequences exist neither in training nor in the testing dataset. As the NetMHC series of servers only provide peptide-binding capacity, the proposed ensemble model is more efficient as it directly predicts whether a peptide is the ZIKV T-cell epitope or not as shown in Table 9. The proposed ensemble model was also compared with the CTLpred server [40] using the same unknown dataset. CTLpred server only predicts T-cell epitopes of length up to 9-mers as shown in Table 10, but the proposed model predicts peptide sequences of any length. In addition, the CTLpred server is limited to ANN and SVM, but in the proposed ensemble model, we have used more powerful and efficient classifiers. The comparison results as shown in Table 10 clearly indicate that the proposed ensemble model outperforms the existing techniques.

4 Discussion

ZIKV disease is considered as one of the devastating diseases and is affecting millions of lives globally, especially third world countries. The urgency to have preventive measures against the global threat due to the ZIKV outbreak has awakened scientists and researchers to investigate this pathogen [3]. Due to the non-availability of

Table 10 Comparison results of NetMHC and CTLpred with the proposed model

Sequence	Actual target	Predictions by NetMHC	Predictions by CTLpred	Predictions by Proposed model
NSFVVDGDT	1	49	Epitope	Epitope
VREDYSLECDPAVIG	1	25	—	Epitope
AQMAVDMQT	1	3.9	Epitope	Epitope
FVVDGDTLKECPLKH	1	2.2	—	Epitope
GEAYLDKQ	1	75	Non-epitope	Epitope
GPSLRSTTASGRVIE	1	34	—	Epitope
MEIRPRKEPESNLVR	1	65	—	Epitope
TRGPSLRST	1	7.2	Epitope	Epitope
MLRIINARG	0	3.4	Non-epitope	Non-epitope
IQIMDLGHMATC	0	56	—	Non-epitope
LVTCAKMQ	0	80	Non-epitope	Epitope
LGGFGSL	0	78	Epitope	Non-epitope
VVVLGSQERIN	0	34	—	Non-epitope

vaccine for treatment and prevention of infections due to ZIKV, vaccine development for this pathogen is an active research area. WHO in its news bulletin report [41] have reported that its global spread and recent outbreaks underline the need for research in vaccine development and its continued vigilance. In the USA, the disease is notifiable, with around 5000 notified cases, primarily of travelers returning from the affected areas. Using an experimental approach to identify epitopes is an expensive and time-consuming process. Therefore, it is high time to utilize and take advantage of rapid developments in the immunoinformatics approach. Designing vaccines based on epitopes is already showing remarkable and hopeful results, and this technology is playing a pivotal role in the treatment and prevention of cancer, bacterial, viral, and other types of diseases [42–45]. The model we proposed in the current study is showing remarkable results for the prediction of ZIKV T-cell epitopes. We compared the proposed model with existing methods, i.e., NetMHC and CTLpred servers using an unknown dataset and the results indicate that the proposed ensemble model outperforms the existing methods as the ensemble model combines the decision of multiple base classifiers with an aim to increase the accuracy as compared to individual base classifiers [46, 47].

Hence, the proposed ensemble model is more accurate and robust, and the outcome is a direct classification (deterministic approach) of a given peptide sequence as ZIKV epitope or non-epitope with high accuracy.

5 Conclusion

In this study, we proposed a machine learning-based stacking ensemble model to predict ZIKVT-cell epitopes directly (deterministic approach) based on the physicochemical properties of amino acids unlike the NetMHC series of servers, which provide the prediction of the binding capacity of peptides (probabilistic approach). We verified the classifiers selected for our proposed stacking ensemble through TOPSIS technique. The proposed ensemble model achieved a Gini of 0.968, AUC of 0.996, sensitivity of 0.964, specificity of 0.961, and an accuracy of 93.47% on the validation dataset. On test set a Gini of 0.954, AUC of 0.973, sensitivity of 0.891, specificity of 0.932, and an accuracy of 92.11% was achieved. The robustness of the proposed model was tested using a *repeated k-fold cross-validation technique* and an average accuracy of 91.7% was recorded. We also compared our proposed ensemble model with existing NetMHC and CTLpred methods using an unknown dataset. Hence, it is concluded through results as described in Sect. 3 that the proposed ensemble model is more efficient and accurate for the prediction of ZIKV T-cell epitopes. The predicted epitopes would undoubtedly play a critical role in designing vaccines to save humanity across the globe from this deadly and devastating menace. Our future work will focus on enhancing the accuracy of prediction by developing classifiers that are more powerful by exploring more machine learning classifiers and physicochemical properties of amino acids.

References

1. Report of Centers for Disease Control and Prevention, National Center for Emerging and Zoonotic Infectious Diseases (NCEZID), Division of Vector-Borne Diseases (DVBD) on Zika Transmission (2019) Centers for disease control and prevention. <https://www.cdc.gov/zika/prevention/transmission-methods.html>
2. WHO (1948) Report of World Health Organization. Indian J Pediat. <https://www.who.int/news-room/fact-sheets/detail/zika-virus>
3. Mirza MU et al (2016) Towards peptide vaccines against Zika virus: Immunoinformatics combined with molecular dynamics simulations to predict antigenic epitopes of Zika viral proteins. Sci Rep 2016(December):1–18. <https://doi.org/10.1038/srep37313>
4. Pandey RK (2018) Designing B- and T-cell multi-epitope based subunit vaccine using immunoinformatics approach to control Zika virus infection. J Cell Biochem 1–12. <https://doi.org/10.1002/jcb.27110>
5. Lindenbach BD, Rice CM (2003) Molecular biology of flaviviruses. Adv Virus Res 59(23):61
6. Zhang X, Jia R, Shen H, Wang M, Yin Z, Cheng A (2017) Structure and functions of the envelope glycoprotein in flavivirus infections. Viruses 9(338):1–14
7. Retallack H, Lullo ED, Arias C, Knopp KA, Laurie MT, Sandoval-Espinosa C, Leon WRM, Krcenik R, Ullian EM, Spatazza J, Pollen AA, Mandel-Brehm C, Nowakowski TJ, Kriegstein AR, DeRisi JL (2016) Zika virus cell tropism in the developing human brain and inhibition by azithromycin. PNAS 113(5):14408–14413
8. Meertens L, Labreau A, Dejamac O, Gressens P, Schwartz O, Axl mediates ZIKA virus entry in human glial cells and modulates innate immune responses. Cell Rep 18:324–333
9. Davis BS, Chang G-JJ, Cropp B, Roehrig JT, Martin DA, Mitchell, CJ, Bowen R, Bunning ML (2001) West Nile virus recombinant DNA vaccine protects mouse and horse from virus

- challenge and expresses in vitro a noninfectious recombinant antigen that can be used in enzyme-linked immunosorbent assays. *J Virol* 75:4040–4047
- 10. Monath TP, Guirakhoo F, Nichols R, Yoksan S, Schrader R, Murphy C, Blum P, Woodward S, McCarthy D, Mathis K (2003) Chimeric live, attenuated vaccine against Japanese encephalitis (ChimeriVax-JE): phase 2 clinical trials for safety and immunogenicity, effect of vaccine dose and schedule, and memory response to challenge with inactivated Japanese encephalitis antigen. *J Infect Dis* 188:1213–1230
 - 11. Putnak R, Barvir DA, Burrous JM, Dubois DR, D'Andrea VM, Hoke CH, Sadoff JC, Eckels KH (1996) Development of a purified, inactivated, dengue-2 virus vaccine prototype in Vero cells: immunogenicity and protection in mice and rhesus monkeys. *J Infect Dis* 174:1176–1184
 - 12. Plourde AR, Bloch E (2016) A literature review of Zika virus. *Emerg Infect Dis* 22:1185–1192
 - 13. Slenczka W (2016) Zika virus disease. *Microbiol Spectr* 4:EI10-0019-2016
 - 14. Prasasty VD, Grazzolie K, Rosmalena R, Yazid F (2019) Peptide-based subunit vaccine design of T- and B-cells multi-epitopes against Zika virus using immunoinformatics approaches. *Microorganisms*
 - 15. Alam A, Ali S (2016) From ZikV genome to vaccine: in silico approach for the epitope based peptide vaccine against Zika virus envelope glycoprotein. *Immunology*. <https://doi.org/10.1111/imm.12656>
 - 16. Babar MM, Waheed Y (2016) Prediction of promiscuous T-cell epitopes in the Zika virus polyprotein: an in silico approach. *Asian Pac J Trop Med* 9(9):844–850. <https://doi.org/10.1016/j.apjtm.2016.07.004>
 - 17. Zhao Y, Pinilla C, Valmori D, Martin R, Simon R (2003) Application of support vector machines for T-cell epitopes prediction. *Bioinformatics* 19(15):1978–1984
 - 18. Brusic V, Bajic VB, Petrovsky N (2004) Computational methods for prediction of T-cell epitopes a framework for modelling, testing, and applications. *Methods* 34(4):436–443
 - 19. Bhasin M, Raghava G (2004) Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine* 22(23–24):3195–3204
 - 20. Nielsen M, Lund O (2009) NN-align. An artificial neural network based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinform* 10(1):296
 - 21. Jensen KK, Andreatta M, Marcatili P, Buus S, Greenbaum JA, Yan Z, Sette A, Peters B, Nielsen M (2018) Improved methods for predicting peptide binding affinity to MHC class II molecules. *Immunology* 154(3):394–406
 - 22. Buus S, Lauemøller S, Worning P, Kesmir C, Frimurer T, Corbet S, Fomsgaard A, Hilden J, Holm A, Brunak S (2003) Sensitive quantitative predictions of peptide-MHC binding by a query by committee artificial neural network approach. *Tissue Antigens* 62(5):378–384
 - 23. Andreatta M, Nielsen M (2015) Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics* 32(4):511–517
 - 24. Vita R, Mahajan S, Overton JA, Dhanda SK, Martini S, Cantrell JR, Wheeler DK, Sette A, Peters B (2018) The immune epitope database (IEDB): 2018 update. *Nucleic Acids Res.* www.iedb.org
 - 25. The R Foundation R Programming. <https://www.r-project.org/about.html>
 - 26. Ikai A (1980) Thermostability and aliphatic index of globular proteins. *J Biochem* 88(6):189
 - 27. Boman HG (2003) Antibacterial peptides: basic facts and emerging concepts. *J Intern Med* 254(3):197–215
 - 28. Guruprasad K, Reddy BV, Pandit MW (1990) Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Eng* 4(2):155–161
 - 29. Eisenberg D, Weiss RM, Terwilliger TC (1984) The hydrophobic moment detects periodicity in protein hydrophobicity. *Proc National Academy of Sciences*, 1984, pp. 81(1), 140–144.
 - 30. Gasteiger E, Hoogland C, Gattiker A, Wilkins MR, Appel RD, Bairoch A (2005) The proteomics protocols handbook. Humana Press, Chicago
 - 31. Therneau T, Atkinson B, Ripley B, Ripley MB Package rpart. <https://cran.r-project.org/web/packages/rpart/rpart.pdf>

32. Ripley B, Venables W (2016) Package ‘NNET’, version 7.3-12. [Online]. Available: <ftp://tdf.c3sl.ufpr.br/CRAN/%0Aweb/packages/kernlab/kernlab.pdf>
33. Karatzoglou A, Smola A, Hornik K Package ‘KERNLAB’, version 0.9-27. <ftp://tdf.c3sl.ufpr.br/CRAN/%0Aweb/packages/kernlab/kernlab.pdf>
34. Culp M, Johnson K (2016) The R package ada for stochastic boosting. [Online]. Available: <https://cran.r-project.org/web/packages/ada/ada.pdf>
35. R. port by A. L. and M. W. Fortran original by Leo Breiman and Adele Cutler, “Breiman and Cutler’s Random Forests for Classification and Regression,” 2018. [Online]. Available: <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>.
36. Bruun J (2006) Multinomial logistic regression | R data analysis. <https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/>
37. Hwang C-L, Yoon K (1981) Multiple attribute decision making: methods and applications. Springer-Verlag, New York
38. Viedma MDPM et al. (2020) Peptide arrays incubated with three collections of human sera from patients infected with mosquito-borne viruses [version 3; peer review: 2 approved]. F1000Research 8:1–30. <https://doi.org/10.12688/f1000research.20981.3>
39. Dr Raghava’s Group MHCBN 4.0. <http://crdd.osdd.net/raghava/mhcfn/>
40. Bhasin M, Raghava GPS (2004) Prediction of CTL epitopes using QM, SVM and ANN techniques. Vaccine 22(23–24):3195–3204
41. WHO (2019) Zika: the continuing threat. Bull World Health Organ 97(1):6–7
42. Oyarzun P, Kobe B (2015) Recombinant and epitope-based vaccines on the road to the market and implications for vaccine design and production. Hum Vaccin Immunother 12
43. Steward MW (2001) The development of a mimotope-based synthetic peptide vaccine against respiratory syncytial virus. Biologicals 29(3–4):215–219
44. Almanzar G, Herndler-Brandstetter D, Chaparro SV, Jenewein B, Keller M, Grubeck-Lobenstein B (2007) Immunodominant peptides from conserved influenza proteins—a tool for more efficient vaccination in the elderly? Wien Med Wochenschr 117:116–121. <https://doi.org/10.1007/s10354-007-0393-y>
45. Olsen PAAW, Hansen PR, Holm A (2000) Efficient protection against mycobacterium tuberculosis by vaccination with a single subdominant epitope from the ESAT-6 antigen. Eur J Immunol 30(6):1724–1732
46. Alzubi OA et al (2019) An optimal pruning algorithm of classifier ensembles: dynamic programming approach. Neural Comput Appl 32(2):267–272. <https://doi.org/10.5958/0976-5506.2019.00298.5>
47. Alzubi JA, Kumar A, Alzubi OA, Manikandan R (2019) Efficient approaches for prediction of brain tumor using machine learning techniques. Indian J Public Heal Res Dev 10(2):267–272. <https://doi.org/10.5958/0976-5506.2019.00298.5>

A Deep Learning Approach for Detection of SQL Injection Attacks Using Convolutional Neural Networks



Ayush Falor, Manav Hirani, Henil Vedant, Priyank Mehta,
and Deepa Krishnan

Abstract SQL Injection attacks are one of the major attacks targeting web applications as reported by OWASP. SQL injection, frequently referred to as SQLI, is an arising attack vector that uses malicious SQL code for unauthorized access to data. This can leave the system vulnerable and can result in severe loss of data. In this research work, we have reviewed the different types of SQL Injection attacks and existing techniques for the detection of SQL injection attacks. We have compiled and prepared our own dataset for the study including all major types of SQL attacks and have analyzed the performance of Machine learning algorithms like Naïve Bayes, Decision trees, Support Vector Machine, and K-nearest neighbor. We have also analyzed the performance of Convolutional Neural Networks (CNN) on the dataset using performance measures like accuracy, precision, Recall, and area of the ROC curve. Our experiments indicate that CNN outperforms other algorithms in accuracy, precision, recall, and area of the ROC curve.

Keywords SQL-Structured query language · SQLI-SQL injection attack · ML-Machine learning · Deep learning · CNN · Web applications attacks · Detection techniques

1 Introduction

Catastrophic attacks can be carried out if there exist exploits in your web application system. Of the many types of attack that a hacker can execute—TrustWave states the commonly executed attacks are cross-site scripting (XSS), that makes about 40% of web attack, and attempts on SQL injections [1]. According to a report by Imperva: “All the vulnerabilities in 2019 (17,308) increased by 23% compared to

A. Falor (✉) · M. Hirani · H. Vedant · P. Mehta · D. Krishnan

Department of Computer Engineering, MPSTME, NMIMS University (Deemed-to-be), Mumbai, India

D. Krishnan

e-mail: deepa.krishnan@nmims.edu

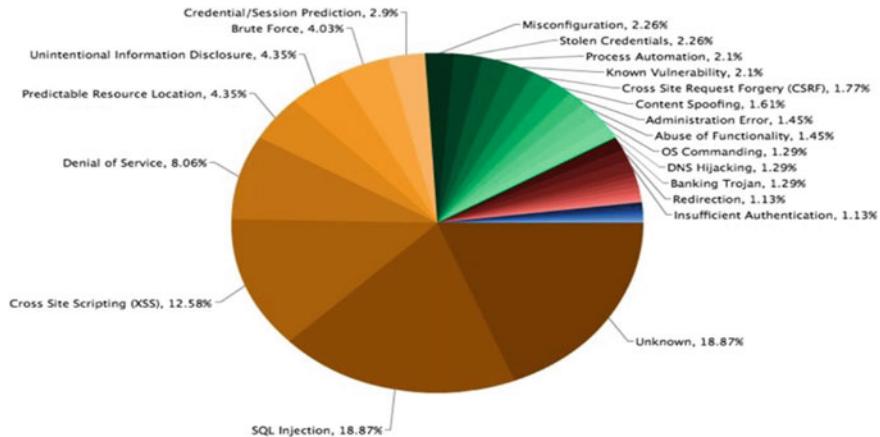


Fig. 1 Distribution of most used Exploits from 2017–2019 according to Hackmageddon [3]

2018 (14,082) and by 162% compared to 2017 (6615s). In addition, more than a third (38%) of web application vulnerabilities do not have an available solution, such as a software upgrade workaround or software patch [1]. SQL injections were used in 51% of cases by hackers found in an Independent study by AKAMI report [2].

According to Hackmageddon, the following is a distribution of all the hacking vulnerabilities and most used exploits from June 2017–2019 [3] (Fig. 1).

A SQLI attack is one of the deadliest attacks because it compromises authentication, integrity, authorization, and confidentiality [4]. This is done by injecting malicious queries into forms and getting access to the database and manipulate its data. One of the main reasons for high success rates of SQLI attacks is improper form validation which can lead to collecting data from databases and publishing sensitive content for monetary gains. Due to its high frequency and vast scope of research, a lot of work has been done in the field, but it was until late when machine learning algorithms started to give promising results.

Some of the important contributions in our proposed research work are summarized as follows:

- We have done an extensive literature survey to review the existent and possible types of SQL attacks and their defense approaches.
- We have compiled and generated a dataset covering all the known types of SQL Injection attacks
- The performance analysis of Deep learning techniques in detecting the SQL injection attacks over conventional machine learning algorithms is tested using various measures like Accuracy, Precision, Recall, and Area under the ROC curve.

The remainder of the paper is as follows in Sect. 2 Literature Survey, Sect. 3 Proposed Methodology, Sect. 4 Results and Discussion, Sect. 5 Conclusion and Future scope and finally References concludes our paper.

2 Literature Survey

The techniques for detection of SQL Injection attacks can be classified into two parts:

- Dynamic—This technique is known as dynamic detection as it uses machine learning/statistical models to classify the queries as malicious or benign and has web-code flexibility to detect and have better prevention.
- Static—This type of approach basically works on dictionary string matching approaches that use NLP to break the given query and compare it with pre-existing dictionary and check if the query is malicious or not.

For our literature survey, we have primarily focused on the Dynamic techniques proposed in studies by various researchers to detect SQL injection attacks.

This literature survey is divided into two sub-sections based on the type of learning technique that was used in the study—Supervised or Unsupervised.

2.1 *Supervised Techniques*

The supervised learning approach makes use of various models to predict the type of statement and the type of attack being performed on the system and at the same time it uses this knowledge to distinguish and label the attack in either malicious or non-malicious category and stops it from executing and all non-malicious queries are processed.

The study presented in Ref. [5] proposes a method in which the query tree is generated from SQL statement, Features are extracted from Query Tree and compared with Dataset trained using SVM classifier. The algorithms used comprised SVM Classifier and Fisher score was calculated for Feature selection. The Weka library is used for the training of vectors.

Some of the drawbacks include that approximately 6% of instances were incorrectly classified by the system. The user was marked as malicious even on execution of 1 query, which may not be feasible in case of False positives. Also, the system has been tested for an inadequate no. of instances. In Ref. [6], HTTP URL string is dissected to check for SQLi threats by comparing with rules that are mentioned. The Solution is provided to handle false positives and false negatives and improve the accuracy overtime. Naive Bayesian model is used for classification of HTTP stream as normal or malicious. The approach suggested in this approach was not tested extensively against a large database and therefore it is not clear to understand it outside the testing environment.

During Feature extraction several factors such as payload length, no of keywords, their weights, etc. are considered in Ref. [7]. And then the URL is classified as malicious or non-malicious using ANN (Artificial Neural Network) models. The Method and Algo used here are Multi-Layer Perceptron (MLP) and LSTM, both of which were Implemented using Pytorch. One of the drawbacks of using such an

approach is that using LSTM model, recognition is poor with high processing time and FPR & has lower accuracy.

Marina et al. have implemented and compared the results obtained from Support Vector Machine (SVM), Multi-layer Perceptron, and Recurrent Neural Networks with Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) classification models [8]. Here, feature extraction is done using various NLP algorithms such as TF-IDP, Word embedding, and bag of words post string analysis.

Since this is a supervised learning approach, the accuracy of system hugely depends on learning dataset and program written to extract features, classify the query and give output in the required time frame. This is a more traditional and static way of classifying attacks as dictionary method is also used to compare strings and predict if it is safe or not.

The study presented in Ref. [9] proposes a novel anomaly-based intrusion detection approach for the detection of attacks which relies on a composition of multiple models to characterize the normal behavior of web-based applications when accessing the database. They have used a probability-based model based on Bayes' theorem to train the data. The system can detect the attacks, though with few false positives and considerable CPU overhead.

Four machine learning models for predicting that a SQL injection attack might occur and also successfully preventing it [10]. The various models that were compared were Support Vector Machine (SVM), Boosted Decision Tree, Artificial Neural Network, and Decision Tree. They have proposed a framework using compiler platform and ML to detect SQL injection in Queries which are illegal and logically incorrect Queries on server-side scripting. For the dataset, 1100 samples of vulnerable SQL commands were trained in the ML models. The models were evaluated in terms of probability of detection, probability of false alarm, precision, accuracy, and processing time. Decision Jungle was the best in terms of performance.

Melody et al. have proposed multi-stage log analysis architecture, which combines both pattern matching and supervised machine learning methods [11]. They have used Kibana for pattern matching and Bayes Net for machine learning. They have evaluated their system against 10,000 web application logs generated using the Log4j framework. The results showed that the combined system of Bayes Net followed by Kibana was able to achieve 95.4% accuracy for detection of SQL injection. But since Bayes Net requires an offline training phase, it would not be possible to implement the system in real time.

Neha et al. have implemented the Naïve Bayes algorithm for deciding whether the query is malicious or not, while using MVC framework [12]. The classifier generates feature vectors from the data received from the dataset by blank separation and tokenizing and learns it by the machine learning method. The results thus obtained have not been analyzed thoroughly and FAR/FRR has not been calculated to establish the efficiency.

Solomon et al. has applied predictive analytics for SQLIA detection and prevention in context of big data [13]. The approach which is suggested here is utilized in a big data context which is lacking in existing works performed on SQLIA. Support Vector Machine (SVM) classifier has been used to make a call on the incoming

request to be safe or unsafe. Multi-class classifiers have not been deployed which limits the system in identifying and grouping the different SQL injection attacks that are predicted which help us improve the efficiency of the system.

2.2 *Unsupervised Techniques*

One of the features of Ref. [14] was that they had developed a Graphical UI/UX for user interaction which gives the user a better understanding as to what is going on behind the execution window. The researchers have applied SQL Parse tree validation and Tokenization.

But this was yet another traditional approach that does not possess code flexibility for malicious inputs. Also, it uses string checking only for select/where clauses so attackers may try to exploit the web application using other queries or strings.

In Ref. [15], Queries are initially normalized considering common techniques used by attackers to bypass detection, HMMs are trained. Only tail end of query is considered and then clustered for making a decision. System works at the database firewall layer to protect multiple websites in a shared hosting environment. Dataset is compiled from log obtained after running automated SQLi tools.

The unsupervised learning approach is a more sophisticated and complicated approach towards the same problem, with reliable results and better accuracy we can use this approach to produce highly efficient results and make the system more stable. This approach has an advantage over supervised learning and that is the system predicts and keeps learning on its own with each new input, new patterns are found and that helps the system to produce better results.

SVM, support vector machine, is used for classification and besides that Naïve Bayes, parse tree, and tokenization are extensively used. Some of them even make use of multi-class classifiers.

Another study proposed a method in which the value of an SQL query attribute from web pages are compared with already existing parameters [16]. This method uses a combined and hybrid approach which is a mixture of static and dynamic analysis. It focuses on comparing static SQL queries with dynamically generated queries after removing the attribute values. The system has the same performance as another method proposed in Ref. [17] with primary difference being they are implementing an automatic method instead of partially automatic. It also requires the web pages to be pre-analyzed (statically and dynamically).

The study presented in Ref. [18] introduces the concept of String checking based on NLP. The method is dependent on sequentially extracting the intended user input from the dynamic query string to check for any malicious input. Here, the select query approach is applied only for where clauses are partial in limiting the problem as it does not cover union queries. Also, the blank space recognition can be manipulated by attackers. The performance of the proposed model when tested with attack vectors shows room for improvement in FAR and GAR percentages as well.

In Ref. [19], the authors have implemented K means clustering as one of the means to predict the attack and to detect invalid users by maintaining an audit record. A central manager is used which helps in detecting attacks as it has clustered neighbors with attributes of records. The results have not been analyzed appropriately to calculate the accuracy, precision, or recall which limits in analyzing the system in comparison with similar methods.

3 Proposed Methodology

3.1 Dataset

One of the primary objectives of our project was to prepare an exhaustive dataset including all the possible types of payloads and queries that can be used for SQL injection attacks. This would robustly help during training of the model so that the system can be highly secured against all types of queries that even professional attackers could use.

A comprehensive dataset containing a range of all possible SQL Injection attack payloads was thus created by compiling payloads for each type of SQL Injection such as generic, blind, error-based and union-based SQL injection. Queries specific to different databases such as Oracle, MS-SQL, MySQL, and Postgres were also included in the dataset. These were taken from different open-source libraries such as Kaggle. These along with the edited versions of these statements were added that would eventually help in creating a flexible system that detects user-defined malicious queries as well.

3.2 Pre-processing

The cluttered text documents containing the payload queries were first broken up into separate queries based on their type. These were then classified based on their type of injection, and labeled as 1 if the payload is malicious, or 0 if it is not malicious. All payloads were then compiled into a data frame with columns as payload query, the type of injection, and whether the payload is malicious or not.

Next, several other such txt documents were compiled together into the data frame. Following this, the data was sorted and then further pre-processed by removing duplicate payloads, and then removing empty and nan datapoints from the table.

The data points which may have been wrongly labeled as malicious were also removed here. Finally, these compiled, sorted and pre-processed payloads were taken and added to a new csv file.

The dataset was thus made presentable, clean, and usable for the algorithm to read and understand.

Fig. 2 Compiled dataset before the cleaning and wrangling operations

The image below shows the compiled dataset before the cleaning and wrangling operations were performed on them (Fig. 2).

After performing cleaning and wrangling, the dataset was obtained as shown in Fig. 3.

3.3 Algorithms

- Decision Tree

Decision tree classification technique is one of the most popular data mining techniques. It applies a straightforward idea to solve the classification problem. The structure has a root node, however, the root node consists of no input and it generates multiple output, a single input gives multiple inner nodes, multiple output, and leaf nodes with input but no output. The decision tree accuracy increases as we keep training the data and is perfect for handling large data in short time. [20]

- Naïve Bayes

Bayes theorem is a probabilistic classifier with strong and naïve independence assumptions. It simplifies learning by assuming that features are independent of

Fig. 3 Dataset after cleaning and wrangling

14011	tomograph	0	LEGAL
14012	sUEn.: sele	1	SQL
14013	boardingh	0	LEGAL
14014	4.25E+15	0	LEGAL
14015	Darlington	0	LEGAL
14016	Calle+Moli	0	LEGAL
14017	roustabou	0	LEGAL
14018	rotondo	0	LEGAL
14019	Calle+Jose	0	LEGAL
14020	Hurtado+E	0	LEGAL
14021	2.24E+15	0	LEGAL
14022	envoys	0	LEGAL
14023	1; USE ma:	1	SQL
14024	marsie	0	LEGAL
14025	ho7AIAndC	0	LEGAL
14026	ll	0	LEGAL

given class. The naive Bayesian model uses a simple classification technique that approximates the class-conditional probability by estimating that features are conditionally independent. This can be applied successfully as we have a binary classification problem of identifying the regular queries as legitimate requests class and malicious ones as SQLi requests class. [6]

- **CNN**
Convolutional Neural Network is a deep learning algorithm which assigns weights and biases to differentiate various aspects/objects from one another. The pre-processing for a CNN is least as compared to all the other models in this paper. A CNN paired with an IDS gives us the ability to monitor traffic and make informed decisions. This increases accuracy, improves results and the number of false alerts can be significantly reduced.
- **SVM**
Support Vector Machine or SVM is a Supervised Learning algorithm. The more common approach of using SVM is for classification problems due to its accurate results. If we feed labeled training data to the SVM model for each category, they are able to categorize new text. Hyperplane is the deciding boundary which separates the two classes of data. SVM chooses the extreme points/vectors that help in creating the hyperplane. SVM then divides the given data into two sides of the hyperplane and this categorization helps us generate results. The outliers in SVM cases are called as support vectors, and that's how the name of the algorithm is termed as Support Vector Machine.

- KNN

KNN is a non-parametric lazy learning algorithm. KNN is another algorithm which is extensively used for classification and regression analysis. KNN algorithm makes use of data and classifies new data points based on the similarity measures (ex. Distance function). The classification approach is done by a majority of its voting done by the Neighbors. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point to improve the efficiency we must increase the nearest neighbors that the node has [21].

3.4 Performance Measures

The performance metrics and measures that we have used in this our study are discussed below:

Accuracy: It represents the correctness of the system.

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

Precision: It indicates the number of times the result is accurate when repeated.

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

Recall: The rate of the predicted request to the total sample.

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

ROC curve: A model which shows performance at various classification values. If the value is closer to 1 it means an ideal solution, closer to 0 means lower in efficiency.

4 Results and Discussion

In this section, we have presented our results that Fig. 4 presents the results for the five most classifiers that were used in this study namely CNN—Convolutional Neural Network, GNB—Naïve Bayes, SVM—Support Vector Machine, KNN—K-Nearest Neighbors, and DT—Decision Tree.

As we see from the graph that while GNB provides the highest Accuracy and SVM provides the greatest precision, but despite this when compared across all 3 metrics we observe that CNN provides the greatest consistency and best results.

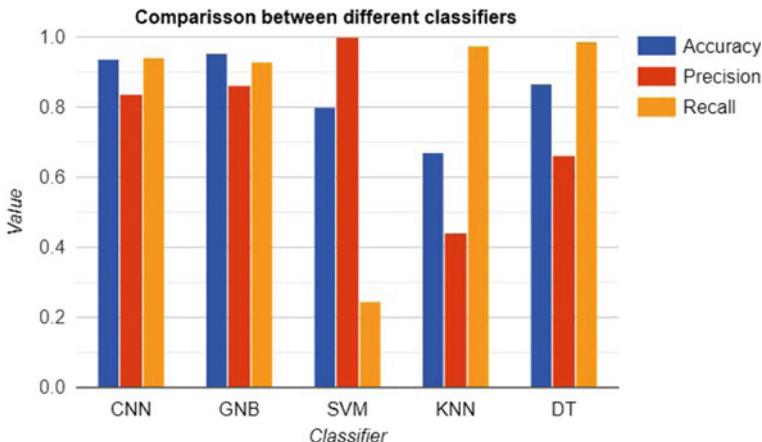


Fig. 4 Comparison between different classifiers

This is further validated by the AUC Value of CNN being highest across all five algorithms, hence we believe CNN is the best Classifier to use for this problem.

Figure 5 shows: False Positives Rate (FPR) is denoted on the x-axis against recall or True Positive Rate (TPR) on the y-axis (sensitivity). On the x-axis higher the value, poorer the performance and the vice-versa for y-axis which has been achieved and shown in the graph.

5 Conclusion and Future Scope

In this paper, we have studied the various techniques used for detection and preventing SQL injection attacks. A comprehensive dataset was created considering all the types of SQLi attack queries as well as the queries used for attacking specific databases. We have then compared five different models used for classification and evaluated their performance. We found that CNN displays the best results with respect to the performance metrics as it showed consistent well-balanced performance with high accuracy of 94.84%, precision as 85.67% and recall of 96.56%. Since these results are obtained after training the model on our exhaustive dataset, we can conclude that it is the most efficient model to be used for detection of SQL injection attacks.

Further, we would explore investigating other deep learning approaches and unsupervised algorithms to improve the performance measures. In future, we would apply feature reduction techniques to study its impact on performance measures. We will also broaden our dataset by adding other categories of SQL injection attacks as and when more types of queries come up in the future.

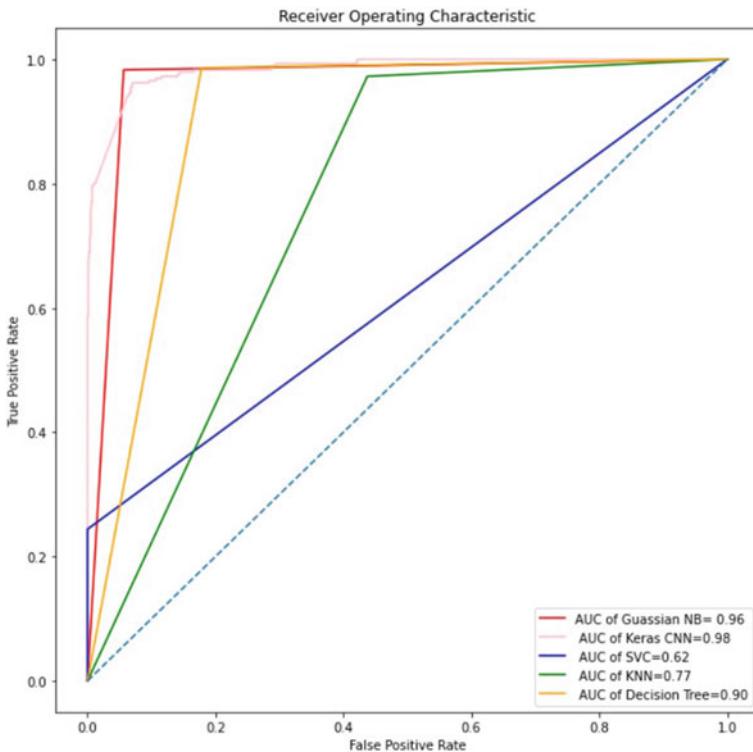


Fig. 5 ROC curves representing the different classifiers

References

1. IMPERA (2020) <https://www.imperva.com/learn/application-security/sql-injection-sqli/>
2. AKAMI <https://www.akamai.com/us/en/resources/prevent-sql-injection-attacks.jsp>
3. HACKMAGEDDON <https://www.hackmageddon.com/2021/01/13/2020-cyber-attacks-statistics/>
4. Research India Publications (2020) SQL injection attack detection and prevention techniques using machine learning. *Int J Appl Eng Res* 15(6):569–580. ISSN 0973-4562. <http://www.rip-publication.com>
5. Ladole A, Phalke MDA (2016) SQL injection attack and user behavior detection by using query tree fisher score and SVM classification. *Int Res J Eng Technol* 3(6)
6. Makio A, Begriche Y, Serhrouchni A (2014) Improving web application Firewalls to detect advanced SQL injection attacks. In: 2014 10th International conference on information assurance and security. Okinawa, Japan, pp. 35–40. <https://doi.org/10.1109/ISIAS.2014.7064617>
7. Tang P, Qiu W, Huang Z, Lian H, Liu F (2020) Detection of SQL injection based on artificial neural network. *Knowl-Based Syst* 190:105528. ISSN: 0950-7051. <https://doi.org/10.1016/j.knosys.2020.105528>
8. Volkova M, Chmellar P, Sobotka L (2019) Machine learning blunts the needle of advanced SQL injections. *MENDEL* 25:23–30. <https://doi.org/10.13164/mendel.2019.1.023>

9. Valeur F, Mutz D, Vigna G (2005) A learning-based approach to the detection of SQL attacks. In: Detection of intrusions and malware, and vulnerability assessment, pp 123–140. Available: https://doi.org/10.1007/11506881_8
10. Kamtuo K, Soomlek C (2016) Machine learning for SQL injection prevention on server-side scripting. In: 2016 International computer science and engineering conference (ICSEC). Chiang Mai, Thailand, pp 1–6. <https://doi.org/10.1109/ICSEC.2016.7859950>
11. Moh M, Pininti S, Doddapaneni S, Moh T (2016) Detecting web attacks using multi-stage log analysis. In: 2016 IEEE 6th International conference on advanced computing (IACC). Bhimavaram, India, pp 733–738. <https://doi.org/10.1109/IACC.2016.141>
12. Hande N, Bhujbal A, Maitri P, Dhiwar A, Raskar S (2018) SQL injection detection and prevention using machine learning. *Int J Sci Res Dev* 6(1):1583–1584
13. Uwagbole SO, Buchanan WJ, Fan L (2017) Applied machine learning predictive analytics to SQL injection attack detection and prevention. In: 2017 IFIP/IEEE Symposium on integrated network and service management (IM). Lisbon, Portugal, pp 1087–1090. <https://doi.org/10.23919/INM.2017.7987433>
14. Halde J (2008) SQL injection analysis, detection and prevention. Master's Projects 82. <https://doi.org/10.31979/etd.mnyq-9gq5>
15. Kar D, Agarwal K, Sahoo AK, Panigrahi S (2016) Detection of SQL injection attacks using Hidden Markov model. In: 2016 IEEE International conference on engineering and technology (ICETECH). Coimbatore, India, pp 1–6. <https://doi.org/10.1109/ICETECH.2016.7569180>
16. Lee I, Jeong S, Yeo S-S, Moon J (2012) A novel method for SQL injection attack detection based on removing SQL query attribute values. *Math Comput Model* 55(1–2):58–68 [Online]. Available: <http://dblp.uni-trier.de/db/journals/mcm/mcm55.html#LeeJYM12>
17. Huang Y-W, Yu F, Hang C, Tsai C-H, Lee D-T, Kuo S-Y (2004) Securing web application code by static analysis and runtime protection. In: Proceedings of the 13th international conference on World Wide Web (WWW '04). Association for Computing Machinery, New York, NY, USA, pp 40–52. <https://doi.org/10.1145/988672.988679>
18. Dalai AK, Jena SK (2017) Neutralizing SQL injection attack using server-side code modification in web applications. *Secur Commun Netw* 2017:1–12. <https://doi.org/10.1155/2017/3825373>
19. Singh G, Kant D, Gangwar U, Singh AP (2015) SQL injection detection and correction using machine learning techniques. In: Satapathy S, Govardhan A, Raju K, Mandal J (eds) Emerging ICT for bridging the future—proceedings of the 49th annual convention of the computer society of India (CSI), vol 1. Advances in intelligent systems and computing, vol 337. Springer, Cham. https://doi.org/10.1007/978-3-319-13728-5_49
20. Zhang Y, Liu J, Zhang Z, Huang J (2019) Prediction of daily smoking behavior based on decision tree machine learning algorithm. In: 2019 IEEE 9th International conference on electronics information and emergency communication (ICEIEC). Beijing, China, pp 330–333. <https://doi.org/10.1109/ICEIEC.2019.8784698>
21. Zhang W, Chen X, Liu Y, Xi Q (2020) A distributed storage and computation k-nearest neighbor algorithm based cloud-edge computing for cyber-physical-social systems. *IEEE Access* 8:50118–50130. <https://doi.org/10.1109/ACCESS.2020.2974764>
22. Shalini K, Ravikurnar A, Vineetha RC, Aravind Reddy D, Aravind Kumar M, Soman KP (2018) Sentiment analysis of Indian languages using convolutional neural networks. In: 2018 International conference on computer communication and informatics (ICCCI). Coimbatore, India, pp 1–4. <https://doi.org/10.1109/ICCCI.2018.8441371>

Machine Learning, Deep Learning and Image Processing for Healthcare: A Crux for Detection and Prediction of Disease



Charu Chhabra and Meghna Sharma

Abstract Machine learning has rapidly gained traction in a variety of fields, including science, healthcare, engineering, and biotechnology, in recent years owing to its effective functioning mechanism. Health care has always been a key priority for any government as the industry has made significant progress by using machine learning, artificial intelligence, and deep learning for disease prediction and diagnosis. The central aspect of this paper is to evaluate different machine learning algorithms and classification techniques in order to detect and predict different chronic diseases. The paper discusses supervised classification strategies for detecting diseases such as cancer, psychological disorders, and cardiac disorders, as well as various bioinformatics and biomedical research challenges. The comparison between classification techniques like support vector machines, logistic regression, decision trees, random forest, and Naïve Bayes classifiers has been observed in numerous diseases. The algorithms that are specifically applied in the medical applications and in healthcare sector enabled the clinical experts and physicians to watchdog, diagnose, and monitor the disease effectively and perform appropriate measures in the shortest possible duration. Decision support systems benefit the physicians for effective and timely decision-making capabilities in case of chronic diseases. The paper reviews the implementation of machine learning and deep learning which has undoubtedly contributed toward health informatics, healthcare systems including bioinformatics. After discussing the techniques and comparison of numerous classification algorithms in several diseases, the ones which have efficiently produced an appropriate result in an early detection of diseases have been highlighted. In order to emphasize the issues that must be considered while implementing the methodologies and classification algorithms for an early illness detection system, several future directions are described.

Keywords Bioinformatics · Healthcare · Chronic disease detection · Machine learning · Deep learning · Image processing · Cancer detection · Psychiatric disorder · Early disease detection · Disease prediction

C. Chhabra · M. Sharma (✉)

Department of Computer Science and Engineering, The North Cap University, Gurugram, India
e-mail: meghnasharma@ncuindia.edu

1 Introduction

Healthcare industry is an important functional industry in a nation. With respect to accomplishing a successful target for the diagnosis of diseases followed by the appropriate treatment, it is essential to develop smart healthcare set up that offers the novel and an adequate framework for the same. Jordan et al. [1] reviewed a research related to the recent progress in intelligent machines and computation intelligence which aimed at transforming the healthcare thereby making efficient systems for physicians to be able to diagnose the diseases and prognosis along with minimal cost alternatives easing out the patients. Generally, the techniques have been an aid to numerous industries thereby working efficiently in accuracy-based models in order to predict the resulting outcomes with higher approximation. Techniques such as supervised learning and unsupervised learning have undergone compelling advancement over the past decade thereby promising to provide an intelligent computing solution and environment to explore a wide area of data related problems. It is decent to state that machine learning (ML) has been proving to be promisingly effective in numerous fields of medical diagnostics. The classification, clustering, and regression analysis techniques have been provocative in data management in healthcare sector in the form of electronic health records (EHR), personalized medicines, diagnosis, and prediction in order to develop an early detection and prediction system [2]. There have been undoubtedly many significant works already exhibited in the prediction and diagnosis of diseases like cancer [3, 4], cardio-related disorders [5, 6], tumors [7], imaging related neuroimaging diseases [8], and fertility in women [9, 10] related diseases. The techniques being the most looming subfield of artificial intelligence (AI) along with an advent of computing technologies has now enabled simpler handling, analyzing and storing the humongous data that is widely used in the healthcare sector in the form of digital records. It is indeed an undeniable fact that an intelligent healthcare system is an essential and esteemed domain. The algorithms such as artificial neural network (ANN) and convolutional neural networks (CNN) are representation-based learning methods which works in multiple levels of representation that are accomplished by establishing elementary but precarious modules which converts the depiction at one level probably the raw level into a representation at a specifically higher and abstract form. Various techniques have showcased their capability to analyze the data of the patients effectively. In the areas of early disease detection and prediction in psychiatric disorders [11], cancer [12], genomics [13], and proteomics [14], support vector machines (SVM) have outperformed numerous classification algorithms. The techniques intake the experimental data to be defined as an input and further models this data to perform predictive modeling for appropriate analysis of disease using dimensionality reduction other learning parameters. Deep learning which happens to be the subset technology has been observed helpful in therapeutic field where in it credits maximum outcomes, committed learning procedure with consolidated element learning, capacity to deal with convoluted and multi-methodology information, and so on. Owing to an extensive enactment of the techniques in healthcare systems, there have been numerous audits and survey published, and one such is

Table 1 Study of the comparison of approach adopted in the past

Reference	Year of publication	1	2	3	4	5	6
Karatekin et al. [15]	2019	Y	–	–	–	–	–
Ahmad et al. [16]	2018	–	Y	–	–	–	–
Shailaja et al. [17]	2018	–	–	Y	–	–	–
Yoo et al. [21]	2017	Y	–	–	Y	Y	–
Reamaroon et al. [18]	2019	–	–	–	–	–	Y
Sujatha et al. [19]	2020	–	–	Y	–	Y	Y

1. Prediction analysis for disease pattern recognition
2. Machine learning models in smart healthcare systems
3. Clinical decision support system
4. SVM classifiers
5. Big data analytics for electronic health record
6. Supervised learning techniques for prediction

depicted in Table 1. For instance, Karatekin et al. [15] elaborated on the functionality of the illness prediction system, which analyses disease spread and detects it early. The research explored the trade-off between accuracy and interpretability of the approaches on clinical data as part of this approach. The research mainly emphasized on statistical analysis and logistic regression as a type of generalized additive model (GAM) to study the risk variables that contribute to serious retinopathy of prematurity. Ahmad et al. [16] discussed about the supervised techniques like regression for the in-depth analysis of various possibilities that could be approached for finding solutions to the issues pertaining in medical data management. Shailaja et al. [17] discussed big data analytics and the management of digital records of health care which might be used to create a clinical decision support system effectively. Yoo et al. [18] discussed the analytical and prediction-based analysis of SVM classifiers thereby enabling in classifying and helping understanding the diagnosis and further progression of the diseases. The study also laid emphasis on the scenario when there is no surety which attribute to include for disease prediction. In such a case, the researchers managed with working on an alternative instead of defining too many unimportant attributes in medical disease sample datasets, which can sabotage classification and increase the number of medical disease prediction calculations that are unnecessary. Reamaroon et al. [19] described the supervised techniques which are presently implemented for algorithmic analysis of order to exhibit disease prediction. The research model replicated the prior analysis to show the influence of alternative sampling thresholds on prediction generalizability for SVM with label uncertainty, to demonstrate the proposed sampling technique. Sujatha et al. [20] depicted the working of EHR followed by the various phases in the framework first being the representation learning, data evaluation and fitting model.

2 Proposed Methodology

In the case of chronic disease detection and prediction, feature selection and classification techniques play a vital role. Feature selection is an effective preprocessing technique used in learning algorithms which relatively works on reducing the dimensionality of the data. The reason of this peculiar step is abiding the fact that in health care, the approach for diagnosing the disease should primarily focus on identifying the possibilities of the risk factors in relation to the disease status and possible spread. Feature selection techniques of data mining helps in reducing the dimension of the data and thereby provides feature identification strategy which works on the eradication of the unnecessary data from the dataset. This in response generates better and rapid results. The other framework which the paper proposes is the deployment of the classification techniques. Classification techniques K-nearest neighbor, random forests, and SVM work well for prediction of the disease pattern at an early stage. The technique refers to predictive modeling in such a way that category or a class is labeled predicted for the respective input dataset. The section compares the use of categorization algorithms in various diseases and their promising results. In the paper, a survey for all the classification algorithms in different disease have been depicted along with the evaluation criteria and accuracy. There is a major requirement of a hybrid and novel classification technique which can ease the process of detection of the chronic and severe disease at an early stage and helps out the clinical experts to predict the possible spread and intensity of the disease in the first phase itself or prior. Feature selection method is categorized into three major categories. Filter method being the first category, wrapper method, and embedded method being the next, respectively. The best part of the methods is that the hybrid methodology has shown promising results generally in the case of detection and prediction of numerous disease such as diabetes, Alzheimer's, Parkinson's, heart disease, strokes, etc.

To fill the possible gaps in the above laying study of comparison, our paper presents the state-of-the-art in the learning techniques to solve the healthcare issues and health informatics problems. In order to summarize the major contributions of the article, the following layout is as depicted • The paper presents the execution of advanced techniques and their adaptability for endeavoring the smart health care system • It focuses more over research on adopting supervised and unsupervised methods for algorithmic analysis for early prediction of the disease • It also explains the applicability of the techniques along with big data analytics in order to demonstrate an effective electronic records management for good patient doctor relationship. • Further reviews the existing techniques adopted for healthcare in disease prediction, issues, and challenges of diagnosis and featured of prognosis of diseases.

The remaining of this paper is presented in the three sections. Section 3 describes the overview of ML techniques in the healthcare sector and proposed methodology. Section 4 presents an overview of the issues and challenges in healthcare and bioinformatics in general. This section delves into the issues that clinical practitioners and healthcare specialists encounter on a day-to-day basis. The section represents

the challenges in the health care in the form of patient's data management and workability of EHR in hospital management system. Section 5 reviews the possible applications of the numerous learning techniques in medical industry and healthcare. The section depicts the challenges in healthcare informatics and personalized medicines. Section 5.1 describes in-depth analysis of machine learning in radiology and radiotherapy. Section 5.2 represents the in-depth analysis and review of the disease prediction systems. Several techniques have been applied so far and also inculcation of ANNs, SVM, etc., have been explained which have been applied and still there is major scope of doing in the same. Section 5.3 discusses the medical imaging concepts and how the techniques have led to significant improvements in the disciplines of speech recognition and image recognition thereby contributing majorly in the field of medical imaging. The section also depicts the medical imaging applications. Section 5.4 elaborates possibilities of tracing and diagnosing the diabetes in patients using the learning techniques and hybrid concepts. In the similar section, the possible scope of improvement and gap analysis has been elaborated as well. Section 6 elaborates the cognitive intelligence and image processing for disease pattern recognition in psychiatric disorder and also depicted the crux of deep learning with image processing for disease prediction. Section 7 depicts the conclusion of the paper by portraying the generic overview of possible methods employed and counterpart for detection and on time prediction of the diseases.

3 Machine Learning in Disease Detection and Prediction

ML is an interdisciplinary discipline that is considered a subset of AI. It allows systems to adapt and learn, as well as perform with experience, without the need for explicit programming. The techniques focus on developing intelligent machines that retrieve the data and also enables the systems to learn by itself. Medical industry works on the plethora of data which could be in the form of statistical and probabilistic data, uncertain, incomplete data, noisy and unwanted data and missing data anomaly that may endanger the structure of an artifact. O.Terrada et al. [22] proposed the viability of a variety of learning approaches in medical care. The research emphasized on the use of ANN and k-nearest neighbor (KNN) in order to predict symptoms and interpret the condition of patients by simply stating whether the patient is suffering from atherosclerosis disease or not. Harimoothy et al. [23] elaborated the complex structure of the data which is observed during the implementation of techniques on relatively high dimensional data of patient followed by detection of pattern using image analysis. Owing to this functionality in health care either in the case of skin cancer detection or classification or prediction of Deoxyribonucleic Acid (DNA) and Ribonucleic Acid (RNA), these models have effectively demonstrated efficiency. Figure 1 shows the different techniques.

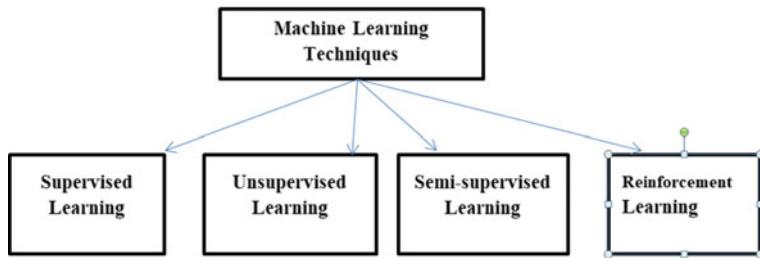


Fig. 1 Machine learning techniques and algorithms

3.1 Supervised Machine Learning

In general, the ML algorithms are categorized majorly into four categories as demonstrated above. Some of the supervised techniques that are extensively used are namely decision trees (DT) techniques, Naïve Bayes (NB) model, instance-based learning, random forests (RF), and SVM. Supervised learning approaches are used to iteratively perform predictions using training datasets. In the event of uncertainty, which is the most common problem in chronic disease prediction, the efficacy of the predictive models is remarkable, as seen by the effectiveness of pattern recognition. In general, supervised techniques deal with the concept in which the model takes in the input data and known retorts of the data along with the output followed by drilling down or training the models to certainly come up with the pragmatic predictions for the response of new data. In a very recent scenario, it has been observed that in the case of breast cancer detection [24], and the supervised techniques such logistic regression and SVM have shown to be effective in diagnosis and prognosis. The research gripped the breast cancer prediction and classification using supervised techniques. Supervised techniques are classified into two categories, namely classification and regression. Some of the classification techniques are linear classifiers, SVM, and K-nearest neighbor. Figure 2 depicts the exact working model of a supervised learning model.

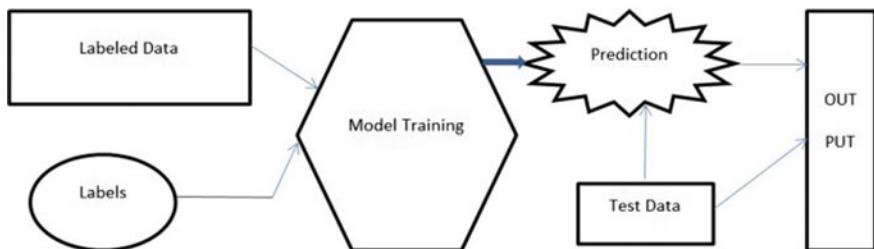


Fig. 2 Supervised learning model

3.2 Unsupervised Machine Learning

Unsupervised learning is a set of statistical algorithms that can be used in situations where there are only a few features and no goal. As a result, predictions relative becomes difficult because each observation has no related responses. Rather major emphasis is laid in locating subgroups with similar observations or creating an engaging way to visualize data. Since there is no defined target for the analysis and it is typically subjective, unsupervised learning is more analytical. Furthermore, evaluating if the collected findings are accurate is challenging because no accepted methodology for performing cross-validation or validating results on a second dataset exists because in this case real answer is unknown. The most frequently used among the unsupervised techniques is namely clustering, principal component analysis (PCA), K-means clustering and hierarchical clustering. Clustering is widely used in the applications of healthcare sector. It primarily focuses on grouping or categorizing data that is homogeneous in some way. Figure 3 depicts the working of clustering technique.

Haratay et al. [25] demonstrated an enhanced k-means clustering algorithm for pattern discovery in healthcare data. In the paper, new model was introduced which worked on new methodology for clustering very huge data, using a hybrid approach to ensemble K-means clustering approach. Bi et al. [26] demonstrated a research strategy for Alzheimer's disorder prediction and diagnosis using K-means clustering techniques. The paper unveils the algorithmic approach by initializing the number of clusters k and the cluster centers which were randomly chosen. Euclidean distance for every single point was calculated. The k-means algorithm divides a group of subjects into k clusters, and the cluster label for each subject is the method output. In return, there could be a receipt of the final prediction accuracy by comparing the cluster label with the true label that corresponds to the MRI participants. Chen et al. [27] represented the model in which genomics and metrics were utilized in order to predict the disease. In this research contribution, the paper demonstrated an efficient

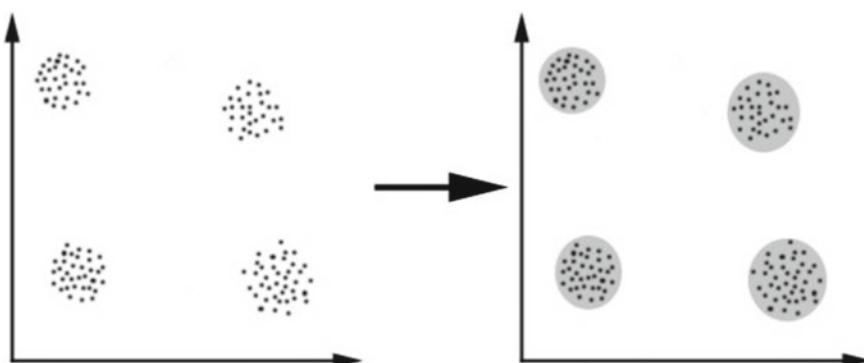


Fig. 3 Clustering unsupervised machine learning algorithm

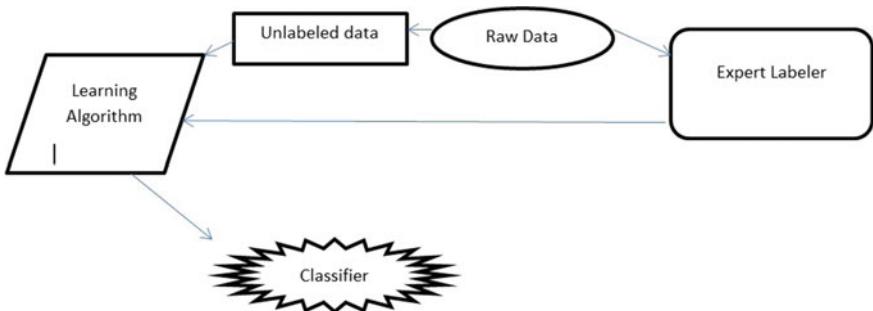


Fig. 4 Semi-supervised machine learning model

and accurate approach for linking in order to predict between genes and diseases. The paper discussed about the comparison of scAnCluster up against three existing supervised scRNA-seq clustering and annotation algorithms based on clusters their analytical pipelines and parameters by default.

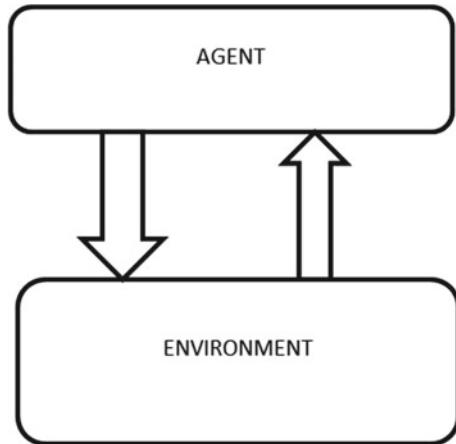
3.3 *Semi-Supervised Learning*

This section emphasizes on another type of learning methodology which is semi-supervised learning, which focuses exclusively on a framework displaying how the machine adapts the exhibition using labeled and unlabeled data. This is a machine learning technique that involves training using a small amount of labeled data and a big amount of unlabeled data. Van Engelen et al. [28] discussed about semi-supervised SVM learning technique stating that it allowed moving the large set of unlabeled data available in many use cases in cumulating them with smaller sets of labeled data. The research model demonstrated the technique worked on laying emphasis on assumptions such as smoothness assumption and low-density assumption. Several strategies were compared, including the mean teacher model, virtual adversarial training, and a wrapper method termed pseudo-label. Figure 4 elaborates the general working of semi-supervised techniques in general.

3.4 *Reinforcement Machine Learning*

Reinforcement learning is the category of intelligent techniques which basically deals with reward-based learning mechanisms. It is a branch of machine learning that studies how intelligent agents should operate in a given environment to maximize the concept of cumulative reward. Reinforcement learning, along with supervised and unsupervised learning, is one of the three main machine learning paradigms. Some algorithms in the category are Q-learning, State-Action-Reward-State-Action

Fig. 5 Reinforcement machine learning model



(SARSA) which resembles the Q-learning and Deep Q-network. Noori et al. [29] demonstrated a research model in which Q-learning was implemented for controlling the population of cancer cells. Q-learning works with finite states and actions, then selects the best action that optimizes long-term rewards. Simin et al. [30] demonstrated a research model in which the findings of the Q-learning algorithm have been compared with test networks, and the desirable performance of the reinforcement learning approach for identifying the ideal number of neurons in hidden layers has been shown. Figure 5 elaborates the conceptual framework of the working mechanism of RL.

4 Challenges in Healthcare: A Deep Dive into the Problems

Data plays a vital role in medical health care as large volume of patient's data, and case studies are available at great lengths. Considering to draw out the conclusions from the oeuvre of humongous data, it is perhaps not that easier and staggering. Instead, it is just not possible for the manual tools and techniques to cater them as it would lead to very slow, inefficient, and banal process. Presently, the scenario of the healthcare industry and bioinformatics has changed rapidly. The healthcare domain relies more on the intelligently computing systems as they are now transforming into smart healthcare systems. Sinha et al. [31] proposed the exhibition of certain techniques which contributed to extract the features that could eventually bulge into a patient oriented and disease specific therapies followed by predictive medicine. Machine learning enabled some of the major contribution in health care and health informatics thereby unearthing the deformity and abnormalities enabling the diagnosis of the disease at a very early stage and prognosis and treatment based on the same. Also, to mark, in case there is any discrepancy in the medical diagnosis, there may be a case of loss of life which is not acceptable and also in normal scenarios expensive medical

costs to patients is not admissible. Lee et al. [32] depicted a research observation wherein they proposed the utilization of the techniques contributing in automated and appropriate workability of medical data. The collection and working of medical records have thereby accredited for simpler access to patient's health database, and it return it also eases out sharing the records among hospitals thereby reducing the costs of hospital operations. Daiyaan et al. [33] elaborated the effective implementation of classification techniques and algorithms in broad range of the clinical tasks thereby supporting the prediction enabled systems. Figure 6 shows the depiction of EHR acting as the repositories for various information systems functionalities in medical diagnosis. Reddy et al. [21] described the ensemble learning methods for the prediction of diabetes retinopathy. Maniruzzaman et al. [34] explained the effectiveness of learning algorithms for the diabetes prediction system using numerous classification techniques. Several techniques like regression and classification have been utilized then for prediction. The application of classification and regression trees has also been proposed for classifying the diseases based on pattern recognition and prediction for disease. Eventually the striding and progressing feature in technology when it comes to the medical imaging, supervised and unsupervised learning techniques have done wonders in capturing and diagnosing disease like cancers, tumors and also for cancer prognosis. This indeed has been the major challenge as it involves image processing and cognitive intelligence with machine learning algorithmic approach for early disease prediction. Naresh et al. [35] described the impact of machine learning techniques in the research in gene sequencing, protein sequencing, and disease prediction on the biomarkers pattern. The paper also highlighted the details, wherein it was depicted that the possible issues and challenges in medical health care which the industry is undergoing at present is related to how the machine learning techniques can be utilized in maintaining the EHR and thereby also predicting the disease at an early stage so it does not affect the patient doctor relationship.

5 Feature Selection and Classification Techniques for the Severe Disease Prediction

The traditional feature selection approaches for learning are categorized into three major categories. Filter method, wrapper method, and embedded method being the three major categories, respectively. Filter method is the conventional method of feature selection. In this method, dimensionality reduction is exhibited in such a way that the features are regularly filtered prior applying any algorithm to the dataset. Wrapper method is another category which emphasizes on the fact of selecting the most essential and useful features followed by optimally selecting the features for learning algorithm to be exhibited. In medical healthcare, classification techniques have played a vital role in many chronic disease detections. There have even been the research observations in which the hybrid model performing clustering followed by classification for the early detection of type 2 diabetes have been concluded. In

order to conclude it emphasizes on an early prediction and detection system and also enables the personalized medicines for the patients so that it can be beneficial for the issues that at present, the industry is facing. Cojbasic et al. [36] presented a model which contributed in such a manner that it laid emphasis on unleashing the supervised learning techniques and algorithms including suitable classifiers for prediction and sampling of the disease based on pattern recognition by the virtue of image processing followed by evolving the predictive medicine and personalized medicines for the patients. Niazi et al. [37] explained the research insight which focused mainly on the patient centric medicine for cardiovascular disease. The paper discussed about the functionality of precision medicine in cardiovascular medicine has all the makings of becoming a major player in the healthcare industry. Various diagnostic tasks, such as diabetic retinopathy, cardiovascular risk calculation, and optical coherence tomography (OCT) images of the eye, have already yielded significant results for some deep learning models. One such interesting research contribution by Saeedi et al. [38] which proposed the working of the human wearables or Fitbit devices in general that enables the depiction of the disease, present stage of diagnosis and early prediction using learning techniques and prediction system. The major highlights of the research were the deployment of transfer learning algorithms in order to detect and predict the disease. This category of machine learning technique deals with creation of model for one task which is utilized as the basis for a model on a different task.

5.1 Classification Techniques for Radiology and Medical Imaging

There has been significant work in the field of radiology and radiotherapy that has contributed to the development of machine learning algorithms that can be used to detect anomalies in cancer health tissues in order to infer the ability to absorb radiotherapy during diagnosis and prognosis for effective radiation treatment. Wichmann et al. [39] reviewed perspectives to radiologists and clinical experts which accommodated techniques solely from their clinical role. Gregory et al. [40] discussed the applicability of deep learning in digital imaging and radiotherapy. The research also focused on generative adversarial networks which happens to be a promising class of DL architecture. The study revealed the capabilities of a deep learning-based computer-aided detection (CAD) model capable of assisting in the interpretation of medical pictures for brain tumor examination. Willemink et al. [41] elaborated the analysis of supervised learning techniques and image processing to recognize the pattern of disease based on imaging thereby enabling an early disease prediction system. The data is used to train and test several supervised machine learning algorithms, including nearest neighbor, multi-layer perceptron (MLP) neural network, and SVM in order to develop an intelligent classification system that will lead to an automatic lesion detection system. In a research recently, Strom et al. [42] the

major focus was on classification techniques Naïve Bayes and decision trees for pattern recognition in radiology in order to provide early disease prediction, as well as prompt diagnosis and treatment for patients. Kruthika et al. [43] presented research which included text or image deep learning system to abstract and exhibit data mining of radiology images and reports from hospital database. In this paper, a multistage classifier based on machine learning was used to identify Alzheimer's disease more accurately and effectively. It included the Naive Bayes classifier, support vector machine (SVM), and K-nearest neighbor (KNN). The research also demonstrated the application of the feature selection technique - PSO (particle swarm optimization) to many feature vectors in order to obtain the best features that represent the salient characteristics of disease. Liu et al. [44] depicted a research contribution in which they elaborated unsupervised learning techniques and CNN for liver cancer detection system and also adopted the classifiers random forests an efficient early prediction system. Cao et al. [45] proposed a plan in which they elaborated how biomedical and genetic data can be utilized for an early disease prediction system. The research model also elaborated the technique of supervised ensemble deep learning models for numerous issues in the field of bioinformatics.

5.2 Classification Techniques for an Early Disease Detection and Prognosis

Machine learning techniques stimulate effectively in disease detection systems. It enables the convenience for the doctors, physicians, and clinical experts to detect the disease in patients using the graphical user interface systems. They are unquestionably effective in the diagnosis of ailments such as diabetes, cancer, mental illness, and kidney and liver disease. Naive Bayes, classification and clustering concepts, SVM, forests, and other learning approaches have been framed out in the domain of disease detection systems. Rahimian et al. [46] proposed a model in which the numerous supervised techniques linear regression, logistic regression, and SVM were compared to the conventional statistical tools that could be used for risk prediction. The research majorly laid emphasis on classification techniques for the detection. Srivastava et al. [47] proposed a plan of methods which could trigger the applicability with respect to automate the diagnosis of disease through images obtained from gastro intestinal biopsies. In a recent study, Sevi et al. [48] expounded on the coronavirus epidemic, which unquestionably poses a daily threat to global health. The majority of their research concentrated on diagnosing disease in persons whose x-rays had been identified as probable COVID-19 candidates. The research was carried out with comparison of performance of deep learning algorithms, namely CNN and recurrent neural network (RNN) (Table 2).

Table 2 Summarized compilation of machine learning techniques used in detecting the disease in radiology and radiotherapy domain

Author	Publication Year	Disease	Approach used	Technique used	Accuracy (%)
Sathiyanarayanan et al. [49]	2019	Breast cancer	Decision tree algorithm	Supervised machine technique	99
Mushtaq et al. [50]	2019	Breast cancer	K-nearest neighbor and Naïve Bayes with sigmoid PCA	Kernel PCA-based techniques were applied and Supervised machine learning approach	99.2
Nitica et al. [51]	2020	Breast cancer	Auto- encoders, t- distributed stochastic neighbor embedding and self- organizing maps were used and deep learning classifiers	Unsupervised machine learning techniques (tSNE, AEs and self-organizing)	93.5
Wang et al. [52]	2021	Cardiac disorder	Semi-supervised learning techniques	Convolutional neural network and self-trainers	89
Ji et al. [53]	2021	Bioinformatics (disease prediction on using microRNAs)	Semi-supervised learning (variational autoencoder) and SVAEMDA	Variational autoencoder and SVAEMDA and clustering	80
Darapaneni et al. [54]	2020	COVID-19	Classification for predicting the admission to either ICU or semi-ICU etc. Biomarkers were utilized in the research and prediction	Supervised machine learning	87

5.3 Bioinformatics- Gene Sequencing and Protein Sequencing for Chronic Disease Prediction

ML and pattern recognition are both concerned with the process of making an automatic judgment. Random forests, SVM, and k-nearest neighbors are frequently used for binary/multiclass classification of test instances and need precise label definitions, whereas unsupervised methods such as PCA, k-means clustering, and semaphore are frequently used for binary/multiclass classification of test instances but instead they do not need specific explicit label definitions. The comparison of the applicability of machine learning techniques for gene prediction and protein sequencing, which are widely used to identify and forecast diseases, is shown in Table 3.

Table 3 Study of comparison of learning techniques adapted for protein sequencing and gene prediction

Author	Publication year	Approach used	Technique used	Criteria of evaluation
Rodrigues et al. [55]	2021	Prostate cancer, Image analysis, PCA ML	Semi-supervised machine learning	PCA3 detection and gene sequencing biomarkers using MLs
Sarkar et al. [56]	2021	Breast cancer survival prediction using gene expression profiling	Graph-based semi-supervised machine learning and Laplacian SVM	Ensemble of feature selection methods
Rajda et al. [57]	2019	The metagenomic data from human microbiome in many cases improves the extent of diagnosis and prognosis for multiple human diseases	Semi-supervised manifold learning and dimensionality reduction convolutional neural network yields out better performance	CNN for metagenomic data used exhibited inconsistent result so supervised linear discriminant analysis
Wang et al. [58]	2020	Computational techniques for protein–protein interactions Protein feature extractions and computational proteomics	Support vector machines and recurrent neural network	SVM with a kernel function as a predictor is utilized for the feature prediction

5.4 Classification Techniques in Diabetes Prediction

Diabetes stands out to be a deep-rooted disease and forms a major challenge in a country. The major cause of diabetes is the increased level of sugar in the blood. There have been many classifiers standardized for anticipating and prognosis of diabetes. Diabetes is undoubtedly a challenging factor in health sector, and also, the most crucial aspect of the disease is that it even cites a risk factor for the patient to develop a mental disorder by developing Alzheimer. Also, it leads to major failure of kidney failure and complete loss of vision. Mohan et al. [59] proposed a plan for diabetes prediction system using SVM classifiers. The most crucial data points in the training dataset are support vectors. The position of the dividing hyperplane would vary if these data points were deleted from the training dataset. They are also the ones that are the most difficult to categorize. Table 4 depicts the comparison of several supervised learning techniques for diabetes prediction with accuracy of the model adapted.

Prabhu et al. [60] demonstrated the research observation in which classification and ensemble techniques were used on a dataset to predict diabetes. Most commonly used were K-nearest neighbor (KNN), logistic regression (LR), decision tree (DT), support vector machine (SVM), and random forest (RF). The accuracy for every model was reported specifically.

6 Cognitive Intelligence and Image Processing for Disease Detections

Deep learning and image processing have contributed immensely in disease prediction. With the passage of decades, there has been a constant observation in increased risk for psychiatric disorder or mental disorder resulting in either life time depression or tendency to attempt suicidal in severe cases. The studies depict that the history of the depressed patients has been widely known. The most important concern is that brain functioning or neural network of human brain may in most of the cases may mediate the effects of the risk attached to depressed patient and may undoubtedly be passed on to the future generations via shared genetic factors. In many extreme circumstances, machine learning and image processing have permitted the discovery of patterns of neurological disturbance and possible proximity of prediction of a patient's ability to acquire a serious ailment. Especially in case of down syndrome prior child birth, autism detection during neonatal period, depression, Alzheimer's memory loss disease, deadly Parkinson's disease. Bahman et al. [64] explained an exploratory research in which they aimed to predict depression in the case of family history of the patient. This is beneficial and applicable when we incorporate machine learning models and image processing for effective and accurate results. In the research observation, they focused on data driven, bias, and unbiased data which was spontaneously on the check of machine learning approaches followed

Table 4 Summarized compilation of machine learning techniques used in detection and prediction of diabetes

Author	Publication year	Disease	Approach used	Technique used	Accuracy (%)
Akula et al. [61]	2019	Diabetes (Type-2)	Naïve Bayes algorithm provided better results compared to other learning methods	Supervised machine learning	85
Tayal et al. [62]	2020	Diabetes	SVM classifiers were utilized to classify labeled and unlabeled data (UI-based tool for diabetes prediction)	Supervised machine learning	89
Radja et al. [57]	2019	Diabetes	SVM, Naïve Bayes, decision trees, J48 were compared for the data set and in conclusion SVM proved out to perform better than all the algorithms	Supervised learning methods	77
Pradhan et al. [63]	2020	Diabetes	SVM classifiers over the diabetes dataset was exhibited and performance of SVM proved out to be efficient as compared to Naïve Bayes and KNN	Supervised learning	96

by deployment of classifiers and regression analysis using regression, dimensionality reduction and also SVM. Sartipi et al. [65] in the research model proposed deployment of supervised techniques with imaging analysis for autism detection which is a genetic disorder. In this research, the major emphasis was on the use of multisite data from resting-state functional magnetic resonance imaging (rs-fMRI) to differentiate autism spectral disorder (ASD) from non-ASD. Using this specific information, the major aim was to classify the patients with ASD and non-ASD. Rutkowski et al. [66] elaborated the research plan in which they employed the cognitive intelligence and deep learning models for prediction of adult dementia. Jain et al. [67] established

the working module in which they described about the concerning mental health issues which are way too common as an observation in today's era. Also elaborated that it affects the cognitive behavior, the ability of individual to sense, hampers the decision-making ability and also affects the wellbeing of an individual. Ahmad et al. [68] established a learning module in which they signified the deployment of radiomics using CNN which have proven to be useful immensely in demonstrating their efficiency for predicting type of gene or genomics based on magnetic resonance imaging (MRI).

7 Conclusion and Future Perspectives

Health care is becoming increasingly important in all economies. A large study on the medical sector has been conducted in order to find answers to many problems in new research domains, consequently enhancing and uncovering many unique solutions to various medical challenges that medical experts, clinicians, and physicians confront. Machine learning methodologies and technologies have been proved to be beneficial in the healthcare industry, according to the findings. It promotes disease growth estimates and prediction, predictive medicine, data-driven research, and diagnostic techniques, broadening the field of medical research. The paper presents a review and survey on feature selection techniques and classification categories which have resulted accurately in many severe diseases. The study shows that classifiers, which are employed as machine learning approaches, can operate successfully in hybrid and ensemble learning approaches to improve the health sector by detecting severe and chronic disease at an early stage and delivering an accurate diagnosis and prognosis to patients. Learning techniques and intelligent systems aid in smart healthcare systems, predictive medicines, personalized medicine, imaging health diagnosis, cancer detection, diabetes detection, drug discovery, and manufacturing of the same specially in maintaining the electronic health records for better clinical decision support systems. The research observations have also described that the crux of feature selection techniques, and appropriate classifiers could result in an accurate and upgraded system of disease identification at an early phase. Classification algorithms which have effectively produced the appropriate results have been of different categories. ANN, SVM, Naïve Bayes, decision trees, and random forests have extremely well depicted their characteristics and performance in many severe diseases. Though, SVM outperformed in many of the disease comparisons, but Naïve Bayes and random forests in ensemble-based ML models have also worked effectively.

References

1. Jordan MI, Mitchell MT (2015) Machine learning: a review. *Machine learning: trends, perspectives, and prospects*. Science 349(6245):255–260
2. Pavlopoulos SA (1999) Designing and implementing the transition to a fully digital hospital. *IEEE Trans Inf Technol Biomed* 3(1):6–19
3. Bazazeh D (2016) Comparative study of machine learning algorithms for breast cancer detection and diagnosis. In: 5th International conference on electronic devices, systems and applications (ICEDSA)
4. Pires NM (2016) Highly sensitive detection of human cancer antigens by an immunogold-silver assay chip coupled with a polythiophene-based optical sensor. In: 38th Annual International conference of the IEEE engineering in medicine and biology society (EMB)
5. Peng C (2020) Cardiovascular diseases prediction using artificial neural networks: a survey. In: IEEE 2nd Eurasia conference on biomedical engineering, healthcare and sustainability (ECBIOS)
6. Mandal T (2020) A comparative study of AI-based predictive models for cardiovascular disease (CVD) prevention in next generation primary healthcare services. In: IEEE International conference for innovation in technology (INOCON)
7. Hemanth G (2019) Design and implementing brain tumor detection using machine learning approach. In: International conference on trends in electronics and informatics (ICOEI)
8. Opbroek V Transfer learning for image segmentation by combining image weighting and kernel learning. *IEEE Trans Med Imag* 38(1):213–224
9. Bakkes THGF (2019) Machine learning for classification of uterine activity outside pregnancy. Annual International conference of the IEEE engineering in medicine and biology society (EMBC)
10. Sammali F (2019) Prediction of embryo implantation by machine learning based on ultrasound strain imaging. *IEEE International ultrasonics symposium (IUS)*
11. Morel D (2020) Predicting hospital readmission in patients with mental or substance use disorders: a machine learning approach. *Int J Med Inf* 139:104136. <https://doi.org/10.1016/j.ijmmedinf.2020.104136>
12. Sharma A, Rani R (2021) A systematic review of applications of machine learning in cancer prediction and diagnosis. *Arch Comput Methods Eng*. <https://doi.org/10.1007/s11831-021-09556-z>
13. Rauschert S, Raubenheimer K, Melton PE, Huang RC (2020) Machine learning and clinical epigenetics: a review of challenges for diagnosis and classification. *Clin Epigenetics* 12(1). <https://doi.org/10.1186/s13148-020-00842-4>
14. Dezső Z, Ceccarelli M (2020) Machine learning prediction of oncology drug targets based on protein and network properties. *BMC Bioinf* 21(1). <https://doi.org/10.1186/s12859-020-3442-9>
15. Karatekin T (2019) Interpretable machine learning in healthcare through generalized additive model WITH pairwise interactions (GA2M): predicting severe retinopathy of prematurity. In: Conference on deep learning and machine learning in emerging applications
16. Ahmad MA, Eckert C, Teredesai A (2018) Interpretable machine learning in healthcare. In: IEEE International conference on healthcare informatics (ICHI)
17. Shailaja K (2018) Machine learning in healthcare: a review. In: Second International conference on electronics, communication and aerospace technology (ICECA)
18. Yoo J (2017) On-chip epilepsy detection: where machine learning meets patient-specific healthcare. In: International SoC design conference (ISOCC)
19. Reamaroon N (2019) Accounting for label uncertainty in machine learning for detection of acute respiratory distress syndrome. *IEEE J Biomed Health Inform* 23(1):407–415
20. Sujatha P (2020) Performance evaluation of supervised machine learning algorithms in prediction of heart disease. In: IEEE International conference for innovation in technology (INOCON)

21. Reddy GT, Bhattacharya S, Siva Ramakrishnan S, Chowdhary CL, Hakak S, Kaluri R, Praveen Kumar Reddy M (2020) An ensemble based machine learning model for diabetic retinopathy classification. In: 2020 International conference on emerging trends in information technology and engineering (Ic-ETITE). <https://doi.org/10.1109/ic-etite47903.2020.9235>
22. Terrada O (2020) Atherosclerosis disease prediction using supervised machine learning techniques. In: 1st International conference on innovative research in applied science, engineering and technology (IRASET). Meknes, Morocco, pp 1–5
23. Harimoorthy K, Thangavelu M (2020) Multi-disease prediction model using improved svm-radial bias technique in healthcare monitoring system. J Ambient Intell Hum Comput
24. Guleria K, Sharma A, Lilhore UK, Prasad D (2020) Breast cancer prediction and classification using supervised learning techniques. J Comput Theor Nanosci 17(6):2519–2522. <https://doi.org/10.1166/jctn.2020.8924>
25. Haraty RA, Dimishkieh M, Masud M (2015) An enhanced k-means clustering algorithm for pattern discovery in healthcare data. Int J Distrib Sens Netw 11(6):615740. <https://doi.org/10.1155/2015/615740>
26. Bi X, Li S, Xiao B, Li Y, Wang G, Ma X (2020) Computer aided Alzheimer's disease diagnosis by an unsupervised deep learning technology. Neurocomputing 392:296–304. <https://doi.org/10.1016/j.neucom.2018.11.111>
27. Chen L, Zhai Y, He Q, Wang W, Deng M (2020) Integrating deep supervised, self-supervised and unsupervised learning for single-cell RNA-seq clustering and annotation. Genes 11(7):792. <https://doi.org/10.3390/genes11070792>
28. J. E. &. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, Vols. 109(2),, pp. 373–440., 2019.
29. Noori A, Alfi A, Noori G (2020) An intelligent control strategy for cancer cells reduction in patients with chronic myelogenous leukaemia using the reinforcement learning and considering side effects of the drug. Expert Syst 38(3). <https://doi.org/10.1111/exsy.12655>
30. Simin AT, Mohsen Ghorabi Baygi S, Noori A (2020) Cancer diagnosis based on combination of artificial neural networks and reinforcement learning. In: 2020 6th Iranian conference on signal processing and intelligent systems (ICSPIS). <https://doi.org/10.1109/icspis51611.2020.9349530>
31. Sinha U, Singh A, Sharma DK (2020) Machine learning in the medical industry. In: Handbook of research on emerging trends and applications of machine learning, pp 403–424. <https://doi.org/10.4018/978-1-5225-9643-1.ch019>
32. Lee G, Fujita H (2020) Deep learning in medical image analysis: challenges and applications. Springer
33. "Daiyaan , machine learning in healthcare.,," *Computational Intelligence for Machine Learning and Healthcare Informatics*, pp. , 277–308. (2020)..
34. Maniruzzaman M, Rahman MJ, Ahammed B, Abedin MM (2020) Classification and prediction of diabetes disease using machine learning paradigm. Health Inf Sci Syst 8(1). <https://doi.org/10.1007/s13755-019-0095-z>
35. Naresh E, Vijaya Kumar BP, Ayesha, Shankar SP (2020) Impact of machine learning in bioinformatics research. Statistical modelling and machine learning principles for bioinformatics techniques, tools, and applications, 41–62. https://doi.org/10.1007/978-981-15-2445-5_4
36. Cojbasic Z (2019) Machine learning for personalized medicine: clinical outcome prediction and diagnosis: plenary talk. In: 13th International symposium on applied computational intelligence and informatics (SACI). IEEE
37. Niazi S (2020) Cardiovascular care in the era of machine learning enabled personalized medicine. In: International conference on information networking (ICOIN)
38. Saeedi R (2018) Personalized human activity recognition using wearables: a manifold learning-based knowledge transfer. In: 40th Annual International conference of the IEEE engineering in medicine and biology society (EMBC)
39. Wichmann JL, Willemink MJ, De Cecco CN (2020) Artificial intelligence and machine learning in radiology. Invest Radiol 55(9):619–627

40. Gregory J (2020) Qualitative thematic analysis of reviewer critiques of machine learning/deep learning manuscripts submitted TO JMRI. *J Magn Reson Imag* 52(1)
41. Willemink MJ et al (2020) Preparing medical imaging data for machine learning. *Radiology* 295(1):4–15
42. Strom H (2018) Machine learning performance metrics and diagnostic context in radiology. In: 11th CMI International conference: prospects and challenges towards developing a digital economy within the EU
43. Kruthika KR, Rajeswari, Maheshappa HD (2019) Multistage classifier-based approach for Alzheimer's disease prediction and retrieval. *Inf Med Unlocked* 14:34–42. <https://doi.org/10.1016/j imu.2018.12.003>
44. Liu HXY (2020) A natural language processing pipeline of chinese free-text radiology reports for liver cancer diagnosis. *IEEE Access* 8:159110–159119
45. Cao Y, Geddes TA, Yang JY, Yang P (2020) Ensemble deep learning in bioinformatics. *Nat Mach Intell* 2(9):500–508. <https://doi.org/10.1038/s42256-020-0217-y>
46. Rahimian F (2015) Predicting the risk of emergency admission with machine learning: development and validation using linked electronic health records. *PLOS Med* 15(11)
47. Srivastava A (2019) Deep learning for detecting diseases in gastrointestinal biopsy images. In: 2019 Systems and information engineering design symposium (SIEDS)
48. Sevi M et al (2020) COVID-19 detection using deep learning methods. In: 2020 International conference on data analytics for business and industry: way towards a sustainable economy (ICDABI)
49. Sathiyanarayanan P (2019) Identification of breast cancer using the decision tree algorithm. In: 2019 IEEE International conference on system, computation, automation and networking (ICSCAN)
50. Mushtaq Z (2019) Performance analysis of SUPERVISED classifiers using PCA based techniques on breast cancer. In: International conference on engineering and emerging technologies (ICEET)
51. Nitica S (2020) A comparative study on using unsupervised learning based data analysis techniques for breast cancer detection. In: IEEE 14th International symposium on applied computational intelligence and informatics (SACI)
52. Wang W (2021) Few-shot learning by a Cascaded framework with shape-constrained Pseudo label assessment for whole Heart segmentation. In: *IEEE Trans Med Imag* 1(1)
53. Ji C et al. (2021) A semi-supervised learning method for mirna-disease association prediction based on variational autoencoder. *IEEE/ACM Trans Comput Biol Bioinf* 1–1
54. Darapaneni NS (2020) A machine learning approach to predicting covid-19 cases amongst suspected cases and their category of admission. In: IEEE 15th International conference on industrial and information systems (ICIIIS)
55. Rodrigues VC, Soares JC, Soares AC, Braz DC, Melendez ME, Ribas LC, Scabini LFS, Bruno OM, Carvalho AL, Reis RM, Sanfelice RC, Oliveira ON (2021) Electrochemical and optical detection and machine learning applied to images of genosensors for diagnosis of prostate cancer with the biomarker PCA3. *Talanta* 222:121444. <https://doi.org/10.1016/j.talanta.2020.121444>
56. Sarkar JP, Saha I, Sarkar A, Maulik U (2021) Machine learning integrated ensemble of feature selection methods followed by survival analysis for predicting breast cancer subtype specific miRNA biomarkers. *Comput Biol Med* 131:104244. <https://doi.org/10.1016/j.combiomed.2021.104244>
57. Radja M (2019) Performance evaluation of supervised machine learning algorithms using different data set sizes for diabetes prediction. In: 5th International conference on science in information technology (ICSITech)
58. Wang Y et al. (2020) A survey of current trends in computational predictions of protein-protein interactions. *Front Comput Sci* 14(4)
59. Mohan N, Jain V (2020) Performance analysis of support vector machine in diabetes prediction. In: 2020 4th International conference on electronics, communication and aerospace technology (ICECA)

60. Prabhu P, Selvabharathi S (2019) Deep belief neural network model for prediction of diabetes mellitus. In: 3rd International conference on imaging, signal processing and communication (ICISPC)
61. Akula R et al. (2019) Supervised machine learning based ensemble model for accurate prediction of type 2 diabetes. In: 2019 SoutheastCon
62. Tayal R, Shankar A (2020) Learning and predicting diabetes data sets using semi-supervised learning. In: 10th International conference on cloud computing, data science and engineering (Confluence)
63. Pradhan R (2020) Diabetes mellitus prediction and classifier comparative study. In: International conference on power electronics and IoT applications in renewable energy and its control (PARC)
64. Zohuri B, Zadeh S (2020) The utility of artificial intelligence for mood analysis, depression detection, and suicide risk management. *J Health Sci* 8(2). <https://doi.org/10.17265/2328-7136/2020.02.003>
65. Sartipi S et al. (2018) Diagnosing of autism spectrum disorder based on Garch variance series for RS-fMRI data. In: 9th International symposium on telecommunications (IST)
66. Rutkowski TM et al (2020) Classifying mild cognitive impairment from behavioral responses in emotional arousal and valence evaluation task—AI approach for early dementia biomarker in aging societies. In: 42nd Annual International conference of the IEEE engineering
67. Jain MP (2020) Mental health state detection using open cv and sentimental analysis. In: 3rd International conference on intelligent sustainable systems (ICISS)
68. Ahmad A (2019) Predictive and discriminative localization Of IDH genotype in high grade Gliomas using deep convolutional neural nets. In: IEEE 16th International symposium on biomedical imaging (ISBI 2019)

Dynamic Pricing-Based E-commerce Model for the Produce of Organic Farming in India: A Research Roadmap with Main Advertence to Vegetables



Sita Rani, Vivek Arya, and Aman Kataria

Abstract With the development of awareness about health and nutrition, organic products are gaining popularity among the people. Both, the demand and production of the organic products increased heavily in last few years. Although, Indian Government has taken a number of initiatives to encourage organic farming, but still there are some limiting factors due to which the different stakeholders, i.e., growers, sellers and consumers are not much attracted toward perishable organic products specially vegetables. One of the prime factors to discourage its e-trade is the pricing mechanisms. In this paper, a dynamic pricing-based E-commerce model is proposed for the e-trade of the organic vegetables. In the proposed model, four factors, i.e., supply, demand, quality and freshness are considered to dynamically fix the prices of the organic vegetables. When, compared with two other models proposed by Kavyashri et al. and Anusha et al., it has been observed that our proposed model gives more realistic prices to attract the interest of all the stakeholders in the e-trade of organic vegetables.

Keywords Agriculture · E-commerce · Organic farming · Model · Vegetables

1 Introduction

Organic farming is the process of cultivating crops without using synthetic agro-chemicals like fertilizers, pesticides or genetically modified organisms which can maintain the health of the soil, biodiversity, ecosystem and people [1]. It is an

S. Rani (✉)

Department of Computer Science and Engineering, Gulzar Group of Institutions, Khanna, Punjab 141001, India

V. Arya

Department of Electronics and Communication Engineering, FET, Gurukul Kangri (Deemed to be University), Haridwar, Uttrakhand 249404, India

A. Kataria

CSIR-CSIO, Chandigarh 160030, India

approach of farming focused on cultivating the land and growing crops to keep the soil in good health with the usage of organic wastes such as crops, animals and farm wastes, aquatic wastes along with other biological materials and bio-fertilizers to provide required nutrients to crops for better sustainable production in an eco-friendly pollution-free environment [2].

According to United States Department of Agriculture (USDA), “organic farming is a system which rely upon crop rotations, crop residues, animal manures, off-farm organic waste, mineral grade rock additives and biological system of nutrient mobilization and plant protection” [3]. Whereas the Food and Agriculture Organization (FAO) of the United Nations suggested that “organic agriculture is a unique production management system which promotes and enhances agro-ecosystem health, including biodiversity, biological cycles and soil biological activity and this is accomplished by using on-farm agronomic, biological and mechanical methods with exclusion of all synthetic off-farm inputs” [4]. Organic farming is not new method of farming in India but was used before green revolution and was in practice from ancient time [5, 6]. With breakthrough in population, our concern would not be only to maintain production of agriculture but to raise it further with continual method. The researchers and scientists have analyzed that the “Green Revolution” with huge input practice has arrived at a stage and is now sustained with declined recur of decreasing reward. So, an essential equity required to be preserved in every possible way for the survival of life and property.

By observing the increased demand of organic products and their available methods of reaching the population, it has been observed some advanced selling models are required for organic produce in India. So, in this paper we present:

- Role of E-commerce in the trading of organic products, specially vegetables.
- Current scenario of organic products in India.
- A new model based on dynamic pricing policy for the e-trade of organic vegetables.
- Comparison of the proposed model with two existing models.

2 E-commerce in Agriculture

In current era of technology, the Internet has astonishing impact on the society across the entire world [7]. Usage of internet has revolutionized the world and commuted the meaning of education, communication, marketing, health care, etc. [8, 9]. The use of Internet in the agricultural domain may lead transformation in the economy and improve the subsistence of the farmers. Agricultural E-commerce promotes the prospectus of advanced work models by yielding farmer to consumer, consumer to farmer, farmer to business and business to consumer services [10]. The application of E-commerce in the organic agricultural sector is assumed to be more beneficial, translucent and ambitious [11]. The main purpose of using E-commerce for the produce of organic farming is to eliminate brokers to provide maximum benefit to the growers and consumers, fast, convenient and cross-boundary delivery, and

genuine pricing [12]. Agricultural E-commerce helps the farmers to sell their products in a broader market regardless of distance and approach the consumers directly [13]. Successful operation of appropriate E-commerce model for organic agricultural products will facilitate to growers to earn maximum profit. It will also aid the economic growth of the nation.

Developing an E-commerce model for selling organic agricultural products is very taxing project and implementing the model to trade the products successfully is more demanding task. Indian Government has been taking many steps to improve the e-business of organic agricultural produces. Many private companies are also taking necessary measures to ease the deals of agri-products through E-commerce. A number of E-commerce portals are working and business of a range of products is done online. But still, collaborators are hesitant to exploit this media of trading specially for agri-products mainly vegetables. Very few stakeholders are utilizing this method of trading because of which rate of accomplishment of success is very slow. Price fixing methods used by current E-commerce portals are not able to contribute much either to enhance dividends or to decrease loss of the farmers. The shoppers are also not enthusiastic in the purchase of agri-products through these portals as not finding benefits in the product price. Business of agri-products through E-commerce is not beneficial for grower as well as consumer. As E-commerce is the comparatively new method of trading of agri-products, some common traits of E-commerce are yet to be merged. There is a need of change in the pricing mechanism for the efficient run of the system. Dynamic pricing policy by considering the demand, freshness and supply of the products, will play crucial role to make augmentation of E-commerce for organic agri-products a success.

3 Situation of Organic Farming in India

In the year 2011, the population of our India was 1210 million and vegetable production was 146.55 million tons. Expected vegetable in the same year was 230.40 g/person/day whereas in the year 1952 expected consumption was 87.66 g/person/day. At present, recommended level of dietary allowance (RDA) for vegetables is 300 g/person/day. But still we are not self-sufficient to meet this requirement of vegetables and experiencing a shortfall of approximately 30 million tones. It has been observed from the records that there is a 25% post-harvest depletion of vegetables. And only 5% of the total organic vegetable produce is exported. As the agricultural land is decreasing day by day because of urbanization; so, there is big burden on the agriculture field to feed regularly increasing population. There is a continuous bargain for quantity and quality of the agricultural produce. Although it is difficult, it would have been an achievement if both the requirements can be targeted in one go.

41% of the total organic producers of the world are from Asia out of which 0.83 million are from India. From the facts, it has been observed that in the year 2016 total land under organic farming in Asia was tentatively 4.9 million hectare. In India,

Table 1 Yield and growth in the production of organic vegetables for the years 2016–2019

Year	Vegetable	2016	2017	2018	2019
Area (Hectares)	292,309.50				
Yield (Kg)	Chilli	6689.5	22,866	24,553	59,455
	Brinjal	3560	21,643	39,555	24,082
	Okra	493.1	13,085	53,214	28,337
	Tomato	11,015	72,658	246,104	84,564
	Capsicum	0.8	2308	21,090	44.5
	Total	21,758.4	132,559.9	384,516	196,482.5
Yield growth rate (%)	Chilli	0.00	241.82	7.38	142.15
	Brinjal	0.00	507.95	82.76	-39.12
	Okra	0.00	2553.62	306.68	-46.75
	Tomato	0.00	559.63	238.72	-65.64
	Capsicum	0.00	288,387.50	813.82	-99.79
	Total	0.00	509.24	190.07	-48.90

in the year 2018 a total of 3.56 million hectares were reserved for organic farming under the umbrella of National Program for Organic Production. India has got 1st rank in the world in the production of organic produces being 9th in the agricultural land used for organic farming.

In the year 2016–2017, India exported tentatively 0.31 million tons of organic products worldwide. In the same year, at the world level total area under organic vegetation was 0.43 million hectare which was 0.7 person more than the previous year. If we analyze from the production aspect, there is a rising trend in the production of organic vegetables in India every year. This inference is drawn from the data available at Web site of PGS India, shown in Table 1. Data related to yield and percentage growth in yield is presented for five different vegetables, i.e., chilli, brinjal, okra, tomato and capsicum from a fixed land area.

4 Motivation for the Proposed Work

Main motive behind organic vegetable farming is to provide nutrient-rich diet to assure better health of the residents. But, it has been observed from a number of studies that organic farming gives low productivity. On the other side, awareness about organic products has given a rise to its demand in the market. Still, the domain of organic farming is facing a number of challenges in its development. Key challenges faced in the sector are related to supply chain management of organic vegetables, food origin and place of fork, smaller sized farms, sales and marketing strategies, pricing policies, lack of awareness among the people, certification policies issues and market intelligence and insurance for organic vegetation.

But the major issues which are responsible to limit the e-trading of organic vegetables are price determination, wastage of the unsold vegetables, margins of the sellers and suitable price for the buyer. Fundamental aim behind the implementation of the E-commerce portals to sell organic vegetables is to facilitate both the grower and the consumer with better prices. So, the selling price should be fixed by considering the profit of the grower and seller as well as it should attract the buyer. Along with, other important parameters need to be considered while fixing the price of the organic vegetables is freshness of the vegetables along with demand and supply. Existing pricing methodologies are not suitable for the customers which lead to the failure of many organic vegetable E-platforms.

5 Related Work

To promote organic farming, a number of initiatives have been taken by the Indian Government. Researchers have proposed a variety of models to fix the right prices for the organic perishable products to facilitate all the stakeholders, i.e., growers, sellers and consumers. Different models proposed along with the parameters considered in the pricing policy are summarized in Table 2. There are four major parameters, which are considered to fix the price of organic products, i.e., supply, demand, quality and freshness. But none of the model is designed by considering all the parameters together and specially for organic vegetables.

Table 2 Parameters considered by different researchers for various pricing policies for perishable organic products

Publication Detail	Parameters			
	Supply	Demand	Quality	Freshness
Chunlin et al. [14]	X		X	X
LI et al. [15]	X	X		
Xu et al. [16]	X	X		X
Xing et al. [17]	X	X		
Rupal et al. [18]			X	X
Takeshi et al. [19]	X		X	X
Jaekwon et al. [20]			X	X
Anusha et al. [21]			X	X
Lian et al. [22]			X	
Kavyashri et al. [23]	X			X
Banerjee et al. [24]			X	
Yevgenia et al. [25]		X		

6 Dynamic Pricing Policy: E-commerce Model for Organic Vegetables

A regularly adaptive pricing mechanism is mandatory for perishable agricultural products so for organic vegetables. This type of model can easily adapt the dynamic market conditions, take into consideration the revenues/benefits of all the stakeholders and also focus on the perishable feature of the produce. Fundamental aim behind dynamic pricing policy is to provide right product at right price to the right person at right time. In the proposed model the various parameters which have been considered for the dynamic pricing policy are:

- Supply (S)
- Demand (D)
- Quality (Q)
- Freshness (F).

Among these parameters, supply and demand are the quantitative parameters whereas quality and freshness are the qualitative and subjective parameters. Although, a number of models have been proposed previously by considering different parameters for different agricultural products. But there is no existing model for the e-trade of organic vegetables which have been proposed yet by considering all these four parameters together.

6.1 *Ranking of the Parameters*

In the proposed model, all the four parameters have been ranked and assigned weights by considering their relevance in finalizing the price of the organic vegetables. We have assigned high weight values to the quantitative parameters and low to the qualitative parameters. The equation used to depict the dynamic pricing mechanism model is:

$$P_i = 2.3S_i + 4D_i + 1.2Q_i + 0.9F_i \quad (1)$$

Both the qualitative/subjective parameters, i.e., quality and freshness of the product are quantified to apply the model for price calculations. Coefficients for the model equation are fixed by analyzing the sample prices from the secondary data.

6.2 Flow Chart for the Model

The process of the dynamic pricing mechanisms for the proposed E-commerce model for organic vegetables is depicted with a flow chart in Fig. 1. In the beginning, values of the four parameters to predict the price of the organic vegetables are entered as input. Then subjective parameters, i.e., quality and freshness are quantified. After

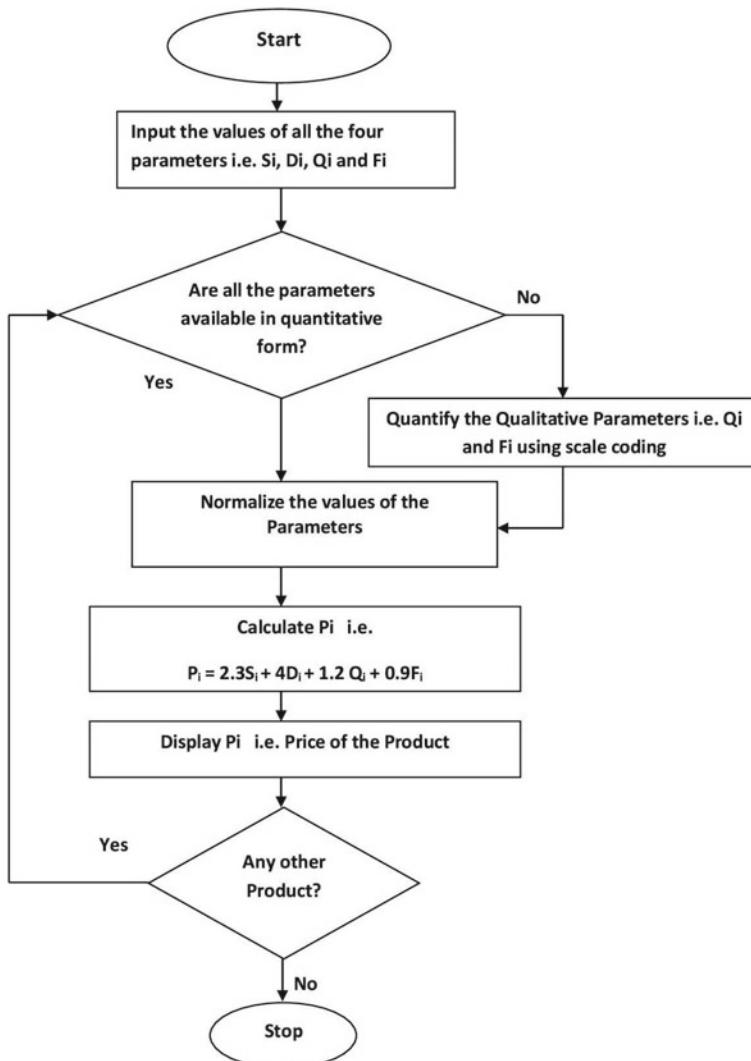


Fig. 1 Flow chart: proposed E-commerce model

that all the four parameters in the quantitative form will be applied in the model to calculate the price of the vegetable. The process is repeated for all types of products available in the cart.

7 Results and Discussion

The proposed dynamic pricing model for e-trading of organic vegetables is used to estimate the prices of five vegetables, i.e., chilli, okra, tomato, brinjal and capsicum. The results obtained from the proposed model are compared with models of Kavyashri et al. and Anusha et al. The results for per 100 kg of each vegetable obtained from all the three vegetables are presented below in Table 3.

Data presented in Table 3 is depicted with the line graph in Fig. 2. After discussion with the different stakeholders from different places, it has been observed that prices

Table 3 Comparison of the price estimated with the proposed model and by Kavyashri et al. [23] and Anusha et al. [21]

Vegetable	Our proposed model	Model by Kavyashri et al.	Model by Anusha et al.
Price in Rs. For per 100 kg			
Okra	380	344	369
Chili	480	512	457
Tomato	290	310	267
Brinjal	278	293	324
Capsicum	493	512	497

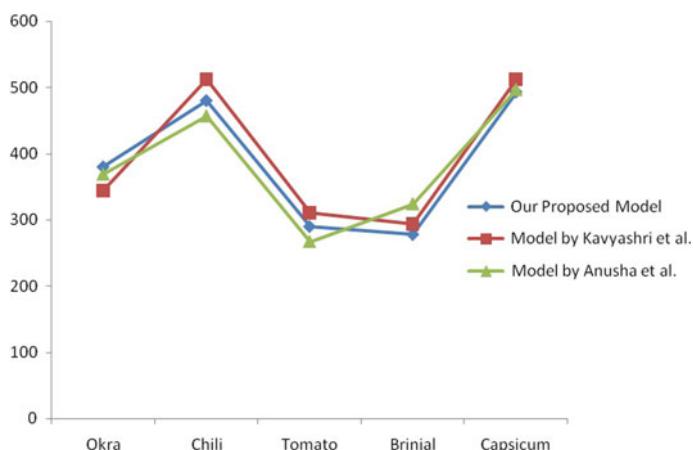


Fig. 2 Comparison of prices predicted by proposed dynamic pricing model with the existing models: line graph

fixed with our proposed model are more close to the expectations of the growers, sellers and the consumers for the organic vegetables sold using e-trade.

8 Conclusions

Amplified usage on the Internet and communication technology (ICT) over last few years has contributed to a number of domains in day-to-day life. It has also benefited the area of agriculture tremendously. Along with other aspects, it encouraged the e-trade of the agricultural products. As in the current era, people are becoming more health and nutrition conscious, so organic products are gaining popularity. In this paper, we have proposed a dynamic pricing-based E-commerce model for organic vegetables by considering the constraints of the growers, sellers and consumers. In the proposed model, supply, demand, quality and freshness parameters are considered to fix the prices. After analyzing the results, it is observed that the prices fixed with the proposed model are more satisfying to all the stakeholders. This model can be extended to consider additional parameters like environmental conditions, eating habits of the population, etc. to have more realistic prices for the organic vegetables.

References

1. Ramesh P, Panwar N, Singh A, Ramana S, Yadav SK, Shrivastava R et al (2010) Status of organic farming in India. *Curr Sci* 1190–1194
2. Leifeld J (2012) How sustainable is organic farming? *Agr Ecosyst Environ* 150:121–122
3. Barik AK Organic farming in India: present status, challenges and technological breakthrough. In: 3rd Conference on bio-resource and stress management international, pp 101–110
4. Effendi I, Shunhaji A (2020) Consumer factors buying organic products in North Sumatera. *Esenisi: Jurnal Bisnis dan Manajemen*, 10:57–68
5. Aldebron C, Jones MS, Snyder WE, Blubaugh CK Soil organic matter links organic farming to enhanced predator evenness. *Biol Control* 146:104278
6. Aghasafari H, Karbasi A, Mohammadi H, Calisti R (2020) Determination of the best strategies for development of organic farming: a SWOT–Fuzzy analytic network process approach. *J Cleaner Prod* 277:124039
7. Rani S, Kataria A, Sharma V, Ghosh S, Karar V, Lee K et al (2021) Threats and corrective measures for IoT security with observance of Cybercrime: a survey. *Wirel Commun Mobile Comput* 2021
8. Bilal M, Kumari B, Rani S (2021) An artificial intelligence supported E-commerce model to improve the export of Indian handloom and handicraft products in the World. In: 2nd Doctoral symposium on computational intelligence (An international conference) organized by Institute od Engineering and Technology, a constituent college of Dr. APJ Abdul Kalam Technical University, Lukhnow, India on March 06, 2021. Available at SSRN 3842663, 2021.
9. Gupta O (2017) Study and analysis of various bioinformatics applications using protein BLAST: an overview. *Adv Comput Sci Technol* 10:2587–2601
10. Ashokkumar K, Bairi GR, Are SB (2019) Agriculture E-commerce for increasing revenue of farmers using cloud and web technologies. *J Comput Theor Nanosci* 16:3187–3191
11. Carpio CE, Isengildina-Massa O, Lamie RD, Zapata SD (2013) Does e-commerce help agricultural markets? The case of MarketMaker. *Choices* 28

12. Gupta O, Rani S (2013) Accelerating molecular sequence analysis using distributed computing environment. *Int J Sci Eng Res IJSER*
13. Rahayu S, Fitriani L, Kurniawati R Bustomi Y (2019) E-commerce based on the Marketplace in efforts to sell agricultural products using Xtreme programming approach. *J Phys: Conf Ser* 066108
14. Luo C, Liu J (2008) Dynamic pricing for perishable products by fuzzy decision. In: 2008 IEEE International conference on service operations and logistics, and informatics. China, pp 2849–2852. <https://doi.org/10.1109/SOLI.2008.4682743>
15. Gen-dao L, Wei L (2010) Dynamic pricing of perishable products with random fuzzy demand. In: 2010 International conference on management science & engineering 17th Annual conference proceedings, pp 191–199
16. Lei X, Xiang-zhi B, Long-wei T (2010) Notice of retraction: dynamic simultaneous optimization of production and pricing under reference effect in perishable products supply chain. In: 2010 International conference on e-business and e-Government. China, pp 3354–3357
17. Wang X, Wen H, Yu B, Zhao S, Li Y (2016) Pricing for perishable goods in advance selling strategy. In: 2016 International conference on logistics, informatics and service sciences (LISS). Australia, pp 1–4
18. Rana R, Oliveira FS (2014) Real-time dynamic pricing in a non-stationary environment using model-free reinforcement learning. *Omega* 47:116–126
19. Koide T, Sandoh H (2009) Reference effect and inventory constraint on optimal pricing for daily perishable products. In: 2009 IEEE International conference on industrial engineering and engineering management. China, pp 370–374
20. Chung J, Li D (2010) A simulation on impacts of a dynamic pricing model for perishable foods on retail operations productivity and customer behaviours. In: 2010 IEEE International conference on industrial engineering and engineering management. Singapore, pp 1300–1304
21. Anusha P, Erol R (2017) A new hybrid model for dynamic pricing strategies of perishable products. In: 2017 Seventh International conference on innovative computing technology (INTECH). Luton, pp 85–89
22. Peng L, Liu H (2007) A dynamic pricing method in e-commerce based on PSO-trained neural network. In: Integration and innovation orient to e-society, vol 1. Springer, pp 323–329
23. Ghose TK, Tran TT (2009) Dynamic pricing in electronic commerce using neural network. In: International conference on E-technologies. Canada, pp 227–232
24. Banerjee T, Mishra M, Debnath NC, Choudhury P (2019) Implementing E-commerce model for agricultural produce: a research roadmap. *Periodicals Eng Nat Sci* 7:302–310
25. Kovalchuk Y, Fasli M (2008) Deploying neural-network-based models for dynamic pricing in supply chain management. In: 2008 International conference on computational intelligence for modelling control & automation. Vienna, pp 680–685

Deep Learning Techniques for Detection of Autism Spectrum Syndrome (ASS)



Anshu Sharma and Poonam Tanwar

Abstract Autistic Ailment is a development disorder distinguished by various factors such as non-verbal communication, repetitive patterns in behavior, and so on. It generally occurs at childhood but it is a type of ailment which grows till lifetime if not treated properly. In recent years, Autism is increasing in India at a massive rate, and therefore, it requires proper and timely diagnosis. Although Autism can be detected by using various tools that are used for screening purpose like Autism Detection Observation Schedule (ADOS), such tools are very time-consuming and lacks accuracy. With the advancement in data analytics, many machine learning techniques like (support vector machine and random forest techniques) and image processing have been used to diagnose the traits of Autism. The main aim of this study is to propose deep learning architecture to detect Autism Syndrome which works on unstructured data and provides faster and timely diagnosis.

Keywords Deep learning · Classifier · CNN · MRI scan

1 Introduction

Autism is a constantly recurring disorder which directly impacts on brain's functioning. It is a neurodevelopment ailment which generally occurs during childhood [1]. The children between age group 2–3 years have high chances of suffering from this disorder [2]. Both genetic and environmental factors are responsible for development of this ailment among children [3].

In India, Once Autism was included in the list of most rare occurring disease but by 2021 it is considered as the most occurring developmental disorder. Children affected from this disorder face lots of difficulties in showing gestures, explaining problems.

A. Sharma (✉) · P. Tanwar

Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India

P. Tanwar

e-mail: poonamtanwar.fet@mriu.edu.in

Autism is a condition which is related to brain's functioning due to which child is unable to do anything properly like child faces difficulty while making gestures, not able to speak properly, not able to concentrate on nearby objects, repeat same gestures again and again, and so on. They face lots of difficulty in explaining things through expressions and body gestures [4].

According to World Health Organization (WHO), 1 out of 168 children are suffering from autism ailment and this rate is rapidly increasing every year. So, it is very important for this disease to be diagnosed as early as possible.

Some of the most common causes that lead to the occurrence of Autism ailment among children includes genetic factors which means that either of parent or some other family member is suffering from this disorder, children is not vaccinated properly and on time, children is given birth at an old age, mother of child suffered from complications during pregnancy, child is born in early months of pregnancy, and so on [5].

There are various tell-tale signs shown by the children suffering from Autism. Some of them are:

1. A child faces lots of problem in explaining things by speech or using expressions and body gestures.
2. Child suffering from Autism finds lots of difficulties in concentrating on things and sometimes keeps on watching same object for longer time period.
3. Child keeps on repeating same thing or words again and again.
4. They don't react to the words or talks of other people.
5. Generally, there voice tone is very low in comparison with typically developed children.
6. They face lots of difficulties in even babbling.

Paper contains following sections: Sect. 2 shows the description of deep learning technique, Sect. 3, Literature Review, shows the different research work in the field of Autism, Sect. 4 contains the main objective behind the research, Sect. 5 describes the brief methodology used, and Sect. 6 includes the discussion and future scope of this study.

2 Deep Learning Technique in Diagnosing Autism

Nowadays, the use of artificial intelligence in the field of healthcare is increasing day by day. There are lots of techniques which are used to diagnose the traits of ASD disorder and one such technique is deep learning.

Deep learning is an advanced version of machine learning with added structures and brain functions called artificial neural network. Deep learning techniques such as convolution neural network (CNN), deep neural network (DNN), recurrent neural network (RNN), and deep belief network (DBN) are used in various fields such as speech recognition, computer vision, machine vision, audio recognition, natural language processing (NLP), and so on.

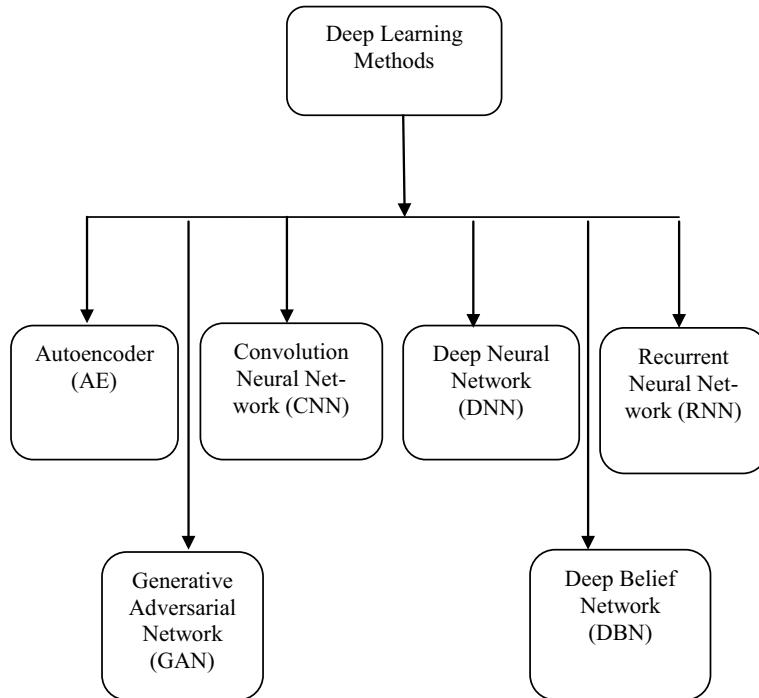


Fig. 1 Illustration of various deep learning methods

Currently, deep learning methods are often used as ML approach and it has enough potential to model linear and even non-linear data structures as well to extract high level features from data [6]. Different deep learning models are shown in Fig. 1.

3 Literature Review

There are various techniques have been used to diagnose Autism traits such as image detection technique, fuzzy logic, and multimedia technique. Among them, machine learning and deep learning techniques are most prominent in prediction of syndrome-based diseases.

Duda et al. [7] used six different machine learning methods. For this author used forward feature selection and tenfold cross-validation to train and test 6 ML models on complete 65-items on Social Responsiveness Scale (SRS) from dataset of 2925 individuals (2775 ASD and 150 ADHD [Attention Deficit Hyperactivity Disorder]). Author found that five of 65 behaviors were sufficient enough to detect ASD from ADHD with high accuracy. Author was able to achieve accuracy 96%.

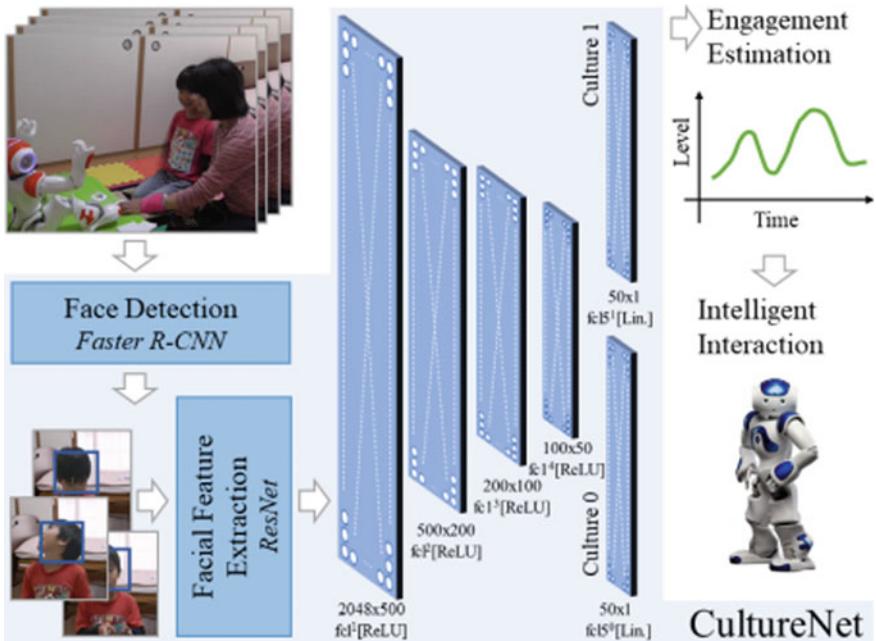


Fig. 2 Automated estimation of engagement from facial images of ASD using robot-assisted therapy. Ognjen et al. (2018) [4]

Ognjen et al. (2018) created novel deep learning model CultureNet as shown in Fig. 2 to get leverage multi-cultural data which author collected in single video session of a robot-assisted Autism therapy. Author used video data of 30 subjects from different cultural background (Asia and Europe). In this research work, author applied different deep learning techniques on facial images of subjects with ASC (Autism Spectrum Condition), collected from one-day robot-assisted therapy session.

Plitt et al. [8] used feasible resting state functional MRI [Rs-fMRI] as measuring tool to diagnose Autism. Based on age and IQ, author collected Rs-fMRI scans of 59 males having severe ASD, 59 males that are typically developing (TD) to build machine learning classifier and obtain classification features from 3 different regions of brain. For better research performance authors also added extra dataset of 178 individuals (89 each) from open-source database ABIDE (Autism Brain Imaging Data Exchange). Authors were able to get accuracy of 76.67% by this model.

Di Nuovo et al. [9] drew light on the use of novel deep learning neural network techniques to test the focus of child on their visual parameter by using robot-assisted therapy session as an indicator of their engagement. Author collected data of low-resolution videos recorded by the robot camera during child-robot interaction session from a clinical experiment as shown in Fig. 3. To analyze clinical data, author created a database and stored different video recordings received during clinical experiments

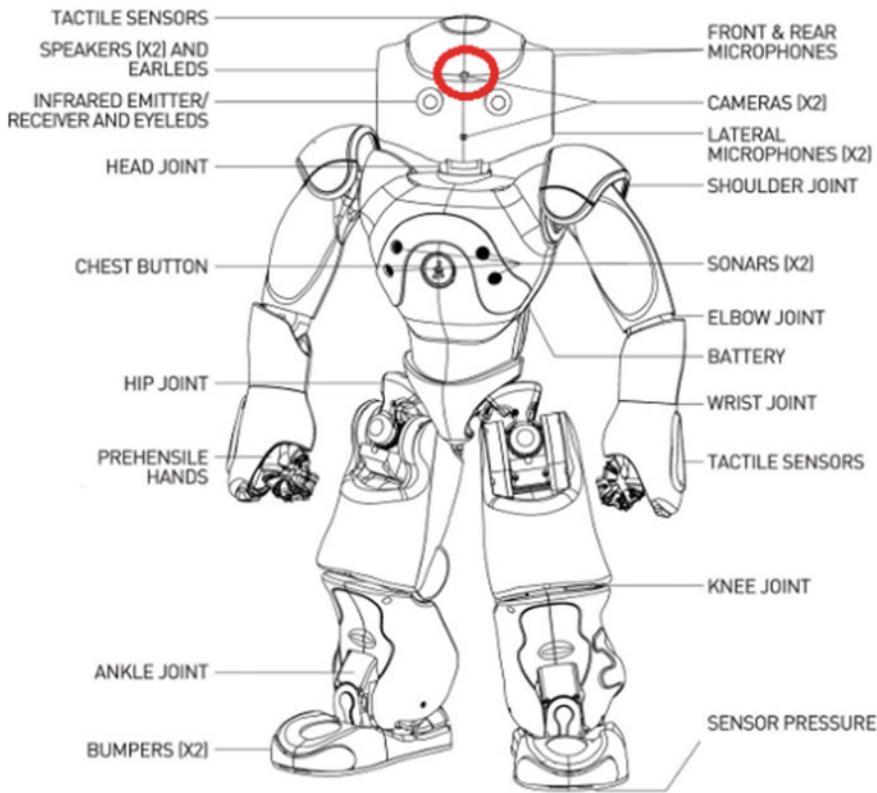


Fig. 3 Figure shows the neo robot containing camera for capturing the image of the child suffering from ASD Di Nuovo et al. [9]

and compare computer visions using centric deep learning approaches like deep neural network and KNN techniques.

Altay and Ulas [10] applied two machine learning algorithms, Linear Discriminant Analysis (LDA) and K-Nearest Neighbor (KNN), to develop a model which will work for subjects between 4 and 11 age groups. Author collected 292 sample datasets out of which 70% used to train model and rest 30% used to test same. Author achieved accuracy of 90.8% using LDA and 88.5% using KNN.

Raja and Masoodb [11] proposed different machine and deep learning techniques for different age groups, i.e., adults, adolescents, and children for these authors collected dataset with following instances (Adults = 704, Adolescents = 104, and Children = 292) and considered 21 attributes for each age group. Authors applied techniques like SVM, KNN, CNN to the dataset and achieved better accuracy through CNN approach, i.e., (Adults = 99.5, Adolescents = 98.3 and Children = 96.8).

Kosmicki et al. [12] applied machine learning techniques on different modules of ADOS (Autism Diagnostic Observation Schedule), i.e., module 2 and module 3 on 4540 subjects and applied ML technique for selection of features to detect Autism. In

this study, authors found that from 28 used behaviors only few features are capable in finding autistic traits, i.e. (Module 2 = 9 and Module 3 = 12), with module 98.27% and 97.66% accuracy, respectively.

4 Problem Formulation

In India, Autism disorder is growing at a higher momentum, and it is estimated that it will rise by 15% till 2021, and hence, it is very important for this disease to be diagnosed as early as possible. Most of the research based on machine learning and image processing provides better accuracy than earlier used tools, i.e., ADOS, but they works on small and structure dataset [13]. Therefore, deep learning techniques are used which works on large dataset.

The main aim of this study is to analyze various techniques used by various researchers and compare best among them. In this study, a new architecture is also proposed which helps to identify the traits of the Autism using MRI images data by applying deep learning features.

5 Proposed Methodology

To fulfill the objective of the research MRI datasets have been collected from open source and then a deep learning architecture is proposed to process this dataset in order to identify traits of Autism.

5.1 Neuro-Image Datasets

Data Collection—The dataset based on image of brain, i.e., MRI scan collected from world largest online repository ABIDE (Autism Brain Imaging Data Exchange) which includes different parameters like repetition time (TR) and time of echo (TE) value, flip angle, etc. that helps in identifying traits of the Autism for different age group.

Data transformation—The collected data was then cleaned to remove noisy data and then divided into two, i.e., 75% of data is used for training purpose and rest 25% is used for testing purpose. After transformation of data it is pre-processed using time spatial smoothing and Temporal filtering which removes components that are unwanted from the voxels of brain.

Deep Learning Model—The deep learning model is designed to fit MRI Scan data into a matrix. The model that has been proposed here will work on Convolution neural network (CNN). The stochastic gradient descent will be used as it provides the best result. There is also a requirement of activation function. So, in our research

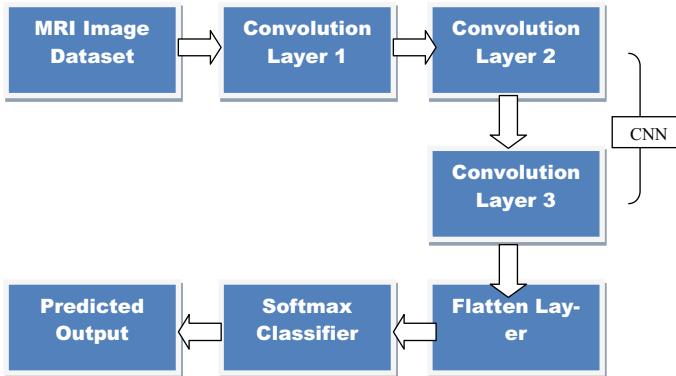


Fig. 4 Block diagram of proposed architecture

we propose the use of ReLU activation function and Softmax Classifier. The model will pass through 4 layers which include 3 CNN layer and one flatten layer.

In the study, the Adam Algorithm will also be used which will help in implementing Gradient descent.

5.2 Proposed Deep Learning Architecture

See Fig. 4.

6 Discussion

In this research, a deep learning model is proposed using which the accuracy of model to detect Autism will increase.

The above research work leads to following discussions:

1. Among different deep learning architecture, mostly researchers have worked on CNN Architecture to detect Autism among subjects.
2. Figure 5 shows the accuracy achieved by using different ML and DL techniques.
3. Figure 6 depicts the graph displaying the accuracy corresponding to the architecture.

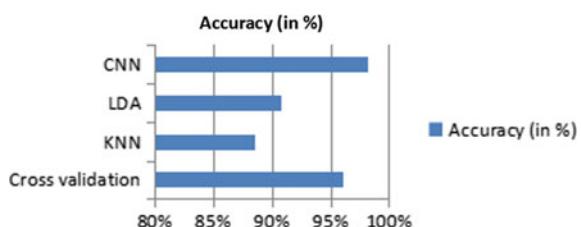
7 Limitation and Future Scope

Different study done by various researchers has a limitation that mostly all the studies are based on the data collected from the publicly available database like ABIDE.

Fig. 5 Accuracy achieved by using different types of datasets

Technique Used	Accuracy (in %)
Cross validation	96%
KNN	88.5%
LDA	90.8%
CNN	98.1%

Fig. 6 Graph depicting accuracy achieved by various datasets



In future, a multi-model system can be developed where we can predict traits of Autism using facial expression, body postures, etc.

References

1. Hauck F, Kliewer N (2017) Machine learning for autism diagnostics: applying support vector classification. In: International conference on health informatics and medical systems, pp 120–123. ISBN: 1-60132-459-6.21
2. Nayar A <http://www.csrmmandate.org/autism-centre-for-excellence-unlocking-unlimited-potentials-for-autistic-children>. 04 Dec 2017
3. Thabtah F (2018) Machine learning in autistic spectrum disorder behavioral research: a review and ways forward. Inf Health Soc Care 44:278–297. ISSN: 1753-8157. <https://doi.org/10.1080/17538157.2017.1399132>
4. Fergus P, Abdulaimma B, Carter C, Round S (2015) Interactive mobile technology for children with autism spectrum condition (ASC). In: IEEE 11th Consumer Communications and Networking Conference (CCNC). IEEE Xplore. <https://doi.org/10.1109/CCNC.2014.7111685>
5. Frith U, Happé F (2005) Autism spectrum disorder. Curr Biol 15(19):R786–R790
6. Bipin Nair BJ, Shobha Rani N, Saikrishna S, Adith C (2019) Experiment to classify autism through brain MRI analysis. Int J Recent Technol Eng (IJRTE), 8(1S4):383–386. ISSN: 2277-3878
7. Duda M, Ma R, Haber N, Wall DP (2016) Use of machine learning for behavioural distinction of autism and ADHD. Transl Psychiatry 6:e732. <https://doi.org/10.1038/tp.2015.221>

8. Plitt M, Barnes KA, Martin (2014) Functional connectivity classification of autism identifies highly predictive brain features but falls short of biomarker standards. *NeuroImage: Clin* 7:359–366. <https://doi.org/10.1016/j.nicl.2014.12.013>
9. Di Nuovo A, Conti D, Trubia G, Buono S, Di Nuovo S (2018) Deep learning systems for estimating visual attention in robot-assisted therapy of children with autism and intellectual disability. *Robotics* 7(2):25
10. Altay O, Ulas M (2018) Prediction of the autism spectrum disorder diagnosis with linear discriminant analysis classifier and K-nearest neighbor in children. In: 2018 6th International symposium on digital forensic and security (ISDFS). IEEE Xplore. <https://doi.org/10.1109/ISDFS.2018.8355354>
11. Raja S, Masoodb S (2020) Analysis and detection of autism spectrum disorder using machine learning techniques. In: International conference on computational intelligence and data science (ICCIDIS 2019). The North Cap University, Gurugram, India, Elsevier, pp 994–1004. <https://doi.org/10.1016/j.procs.2020.03.399>
12. Kosmicki JA, Sochat V, Duda M, Wall DP (2015) Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning. *Transl Psychiatry* 5:e732. ISSN: 2158-3188. <https://doi.org/10.1038/tp.2015.7>
13. Liu W, Li M, Yi L (2016) Identifying children with autism spectrum disorder based on their face processing abnormality: a machine learning framework. In: International society for autism research, Wiley periodicals, vol 8, pp 888–898. ISSN: 1939-3729. <https://doi.org/10.1002/aur.1615>

Test Suite Minimization Based upon CMIMX and ABC



Neeru Ahuja and Pradeep Kumar Bhatia

Abstract Regression testing is an essential part of testing. Regression testing focuses on finding the new errors when some modification is done in software due to circumstances. During the testing, redundant test cases may produce which incur additional cost and time. To save the computational cost, test case minimization is applied. Test case minimization removes redundant test cases. Some criteria should be specified to minimize test cases, and we have selected the fault coverage criterion. For minimization, different evolutionary, nature-inspired algorithms are used nowadays, and ABC is such an influential metaheuristic algorithm motivated by the intelligent activities of honey bees. This paper has proposed a combination of CMIMX and ABC-Min algorithm that minimizes the test suite. We have selected the APFD metric to reveals the effectiveness of the proposed approach. A comparison has been made with and without using minimization, and it becomes clear that results are better using with minimization technique.

Keywords Regression testing · Test case · CMIMX · ABC algorithm

1 Introduction

Software engineering is a comprehensive study of engineering to design, development, and maintenance of software. It is an application of systematic and discipline processes to produce reliable and economical software. The software product is about writing code for the software application, but it comprises many different activities. For the validated software product, all user requirements should be precise in a well-defined manner. The exact requirement leads to software which is according to user requirement. The next phase of software development is designing and coding. As we all know, the software does not wear out, so each step should be carefully taken out. A minor change in software product leads to many errors or bugs that increase cost, time, and faults. Whenever any modification or addition is done in code, the

N. Ahuja (✉) · P. K. Bhatia

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

tester has to retest all code. It is called regression testing, and it is an expensive and time-consuming activity. It is possible that test suites also contain duplicate test cases, and to solve this problem, the test suite minimization technique is used.

1.1 Test Suite Minimization

Test suite minimization removes duplicate test cases. It can be specified as Given [1]: A test case t_1 in test suite T is called duplicate if other test case t_2 in test suite T attains same defined coverage from test requirements set $R \{r_1, \dots, r_n\}$. Test suite minimization can be achieved using different criteria like statement coverage, path coverage, fault coverage, etc. Test suite minimization is implemented with many algorithms like evolutionary algorithm and nature-inspired algorithms. Genetic algorithm is one of the evolutionary algorithms that are widely used in the minimization process.

1.2 The Basic Principle of ABC

Dervis Karaboga introduced ABC algorithm in 2005, inspired by foraging activities of honey bees. It consists of three phases in which movement of bees is recorded employee bee, onlooker bee and scout bee. Bees size in employee and onlooker bee is equal. The pseudocode of ABC algorithm is as follows:

Step 1 Initial food source

Step 2 Iteration = 1

Step 3 Employee phase

Step 3.1 For each food source update variable and partner using Eq. 1

$$X_{\text{new}} = X + \emptyset(X - X_p) \quad (1)$$

Step 3.2 Update variable

Step 3.3 Apply greedy algorithm if no updation increases trail counter by 1.

Step 4 Calculate probabilities (p) using Eq. 2.

$$P_i = f_i / \sum f_i \quad (2)$$

Step 5 Onlooker phase

Step 5.1 Select random variable r

Step 5.2 Check if $r < p$

Step 5.3 Update food source based on random variable and probability.

- Step 6 Memorize the best solution
- Step 7 Scout phase
 - Step 7.1 If trail counter > limit apply scout phase
 - Step 7.2 Calculate fitness and set trail.
- Step 8 Iteration = iteration + 1
- Step 9 Stop when reach to max iteration.

1.3 The Basic of CMIMX

CMIMX algorithm minimizes the test cases based on coverage matrix. Let $F[i][j]$ is a matrix of 8×10 . If T_i covers entity j , mark one else 0. After applying the algorithm, it will represent the matrix called minCov, which contains minimum rows covering all the entities.

Our major contribution of paper is as following:

- Basic introduction of test suite minimization, ABC algorithm, and CMIMX.
- Study of existing work.
- Proposal of a new technique using CMIMX and ABC algorithm.
- Implementation of proposed algorithm.
- Comparative results are defined w.r.t to before minimization and after minimization.

In the next section, a detailed analysis of the proposed algorithm and its results are discussed.

The rest of this paper is structured as follows. Section 2 explains the literature review related to our problem. Section 3 defines the proposed algorithm using CMIMX and ABC algorithm for a minimization problem. Section 4 describes the analysis. Section 5 explores the results by comparison of before and after minimization. Section 6 discusses the conclusion.

2 State-Of-The-Art

In the field of engineering, many optimization problems exist and challenging to solve (Asthana et al. 2020). Test suite minimization is also an NP-hard problem in which search space grows exponentially and complexity increases. Many algorithms are used for solving problems, such as evolutionary, nature-inspired algorithms. Khari et al. [2] proposed a method to minimize the test case with maximum path coverage using ABC and cuckoo search algorithm. Asthana et al. [3] implemented a lion algorithm to select and prioritize test cases with APFD metrics. Kumar et al. [4] proposed a neuro-fuzzy technique to solve the reduction problem, and they also explained the execution time to show the algorithm's effectiveness. Sheoran et al.

[5] implemented the ABC algorithm for data flow testing and used global and local search techniques using the ABC algorithm for path extraction. Shweta et al. [6] compared the ABC algorithm with cuscuta search, GA, and found cuscuta search results are low as compared to ABC and GA. Anwar et al. [7] present an approach, viz. hybrid-adaptive neuro-fuzzy inference system with amalgamation of genetic algorithm and particle swarm optimization algorithm. Chetouane et al. [8] introduce “a machine learning-based algorithm for test suite reduction that combines k-means clustering with binary search. The idea behind the algorithm is to cluster test cases that are close together and to select a representative test case from each of the clusters to be used in the new reduced test suite”. QLSCA was implemented by Zamil et al. [9] and proposed algorithm used Q learning sine–cosine algorithm. Multiobjective optimization was implemented by Kiran et al. [10] using hybrid variant of ANFIS and compared with existing techniques]. GA using weighted fitness function was used by Bhatia [11].

3 Proposed Approach

The proposed algorithm has pros of both techniques, i.e., CMIMX and ABC algorithm. CMIMX minimizes test cases, and based on minimized test case, ABC improves efficiency of test suite. The detailed steps of proposed algorithm are described as follows (Fig. 1):

1. Input fault coverage matrix $F[i][j]$.
2. Minimized number of test cases based on fault coverage matrix using CMIMX procedure.

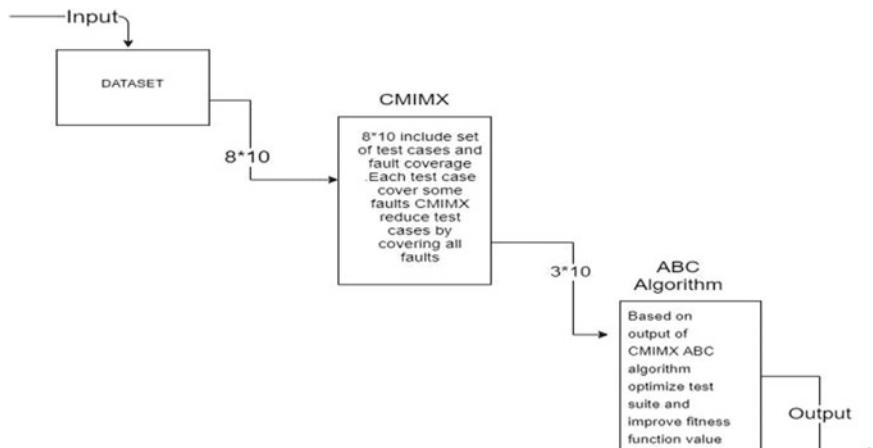


Fig. 1 Architecture of proposed algorithm

- 2.1. Set $\text{MinCov} = \Phi$, $\text{YetToCover} = j$. Unmark each of the i tests and j entities. An unmarked test case is still under observation, whereas; a marked test is already added to minCov .
- 2.2. Repeat the steps until $\text{YetToCover} > 0$. Among unmarked entities, select those columns from $F[i][j]$, containing fewer 1 s. And FC is the set of indices having all such columns. From all the unmarked tests that also cover entities in FC , select those tests that have the maximum number of non-zero entities. Let T be one of these rows. Mark Test T and add it to minCov . Mark all entities covered by test T and reduce YetToCover by the number of entities covered by T .
3. Test suites are initialized randomly using a real encoding that is called food source for the ABC algorithm. Like $TS1 = [8, 5, 3, 8, 1, 2, 6, 1]$, and length of the test suite will be equal to the number of the test case in the fault matrix. Set trail counter, limit.
4. Calculate the happiness value using performance metric and time. Performance metric is calculated using Eq. 3.

$$\text{APFD} = 1 - \{(tf_1 + tf_2 + tf_3 + \dots + tf_i)/i * j\} + 1/2j \quad (3)$$

i = number of test cases.

j = number of faults.

tf_i = Position of first test case that shows the fault.

5. Apply the employee bee phase.
6. Apply the onlooker bee phase.
7. Apply scout phase if needed.

4 Analysis of ABC-MIN Algorithm

4.1 Case Study

The proposed algorithm is implemented on a dataset taken from Singh et al. [12], containing 8 test cases and ten faults covered. Each test case is represented in the fault coverage matrix shown in Table 1. Here, Y illustrates fault occurs, and N means no-fault.

4.2 Performance Metric

Minimization algorithm use fault coverage for a minimized subset. Metric used for fault coverage is APFD and calculated using Eq. 3.

Table 1 Fault matrix [12]

Test case/faults	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	Execution time
t1	N	Y	N	Y	N	N	Y	N	Y	N	7
t2	Y	Y	N	N	N	N	N	N	N	N	4
t3	Y	N	N	N	Y	N	Y	Y	N	N	5
t4	N	Y	N	Y	N	N	N	N	Y	N	4
t5	N	N	Y	N	N	Y	N	N	N	Y	4
t6	Y	N	N	N	N	N	Y	N	N	N	5
t7	N	N	Y	N	N	Y	N	Y	N	N	4
t8	N	Y	N	N	N	N	N	N	N	Y	2

Table2 Minimized test cases

Test case/faults	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	Execution time
t3	Y	N	N	N	Y	N	Y	Y	N	N	5
t4	N	Y	N	Y	N	N	N	N	Y	N	4
t5	N	N	Y	N	N	Y	N	N	N	Y	4

4.3 Implementation of Proposed Algorithm

Fault matrix is taken from Singh et al. [12] as stated above. First of all set $\text{minCov} = \Phi$ and $\text{YetToCover} = 10$. We have eight tests, and ten entities are unmarked and $\text{YetToCover} > 0$. Among unmarked entities, only 5 contain a single 1. Therefore $\text{FC} = \{5\}$ and among unmarked tests, T3 covers 1,5,7,8. Thus $T = 3$ and $\text{minCov} = \{3\}$. Test T3 is marked, and entities 1, 5, 7, 8 are covered by test T3 are also marked. $\text{YetToCover} = 10 - 4 = 6$. Like this loop will continue till $\text{YetToCover} > 0$. In the end, the fault matrix is minimized using the CMIMX technique and shown in Table 2.

The next step is to initialize the food source, i.e., test suite, by using real encoding and set trail equal to zero and limit is 20% of food source. Calculate fitness function using Eq. 3 which is shown in Fig. 2.

For the employee bee phase, we will produce a new test suite on the basis of random numbers. If the fitness value of the new test suite is greater than the old test suite value of the new test suite will be replaced old test suite else trial counter increased by one. Further probabilities are calculated by using Eq. 3, and its values lie between [0, 1].

After the first iteration, probability and trail values are shown in Fig. 3.

Now we move on to the onlooker phase, which applies greedy selection for choosing test suites between new and old test suites. It also selects those test suite whose probability values [0,1] is greater, and if the probability of old test suite is greater, it will remain intact and increased the trail value by one else replaced with the new test suite. If trail value is greater than the limit, apply scout phase like employee

	Testsuite	Fitness	time		
1	[8,5,3,8,1,2,6,1]	0.5625	0.0036		
2	[6,2,4,3,2,3,6,1]	0.7125	0.0025		
3	[5,2,4,4,7,2,4,8]	0.6375	0.0039		
4	[1,1,6,3,8,2,5,5]	0.6375	0.0024		
5	[4,4,2,3,6,8,5,6]	0.5500	0.0023		
6	[6,8,3,4,2,1,5,6]	0.4125	0.0027		
7	[7,5,1,6,2,1,1,1]	0.6125	0.0021		
8	[2,4,3,2,6,1,6,8]	0.7000	0.0028		
9	[4,4,4,8,5,2,2,2]	0.6625	0.0022		
10	[3,3,5,2,6,3,6,4]	0.6000	0.0023		
11					

Fig. 2 Initial test suite with fitness value and time

	1	2	3	4	5	6	7
1	0.1094	2					
2	0.0902	0					
3	0.0993	1					
4	0.1138	2					
5	0.0674	0					
6	0.1160	1					
7	0.0818	0					
8	0.1116	2					
9	0.1303	1					
10	0.0803	1					

Fig. 3 Probability and trail value

bee phase and set trail value equal to 0 and repeat process till max iteration reaches. After max iteration best and worst values for the fitness function are shown in Fig. 4.

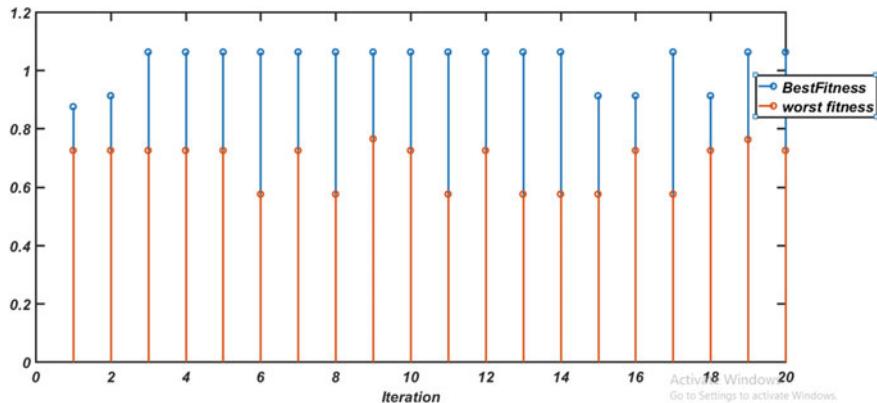


Fig. 4 Best and worst values of fitness function with minimization

5 Result and Discussion

In this section, we have discussed the results of the case study before minimization and after minimization. Figure 4 represents the best value of APFD using with minimization and without minimization. In Fig. 5, 1 specifies the best values for the proposed algorithm, and 2 specify the best values without minimization, and the same applies to Fig. 6 for the worst value.

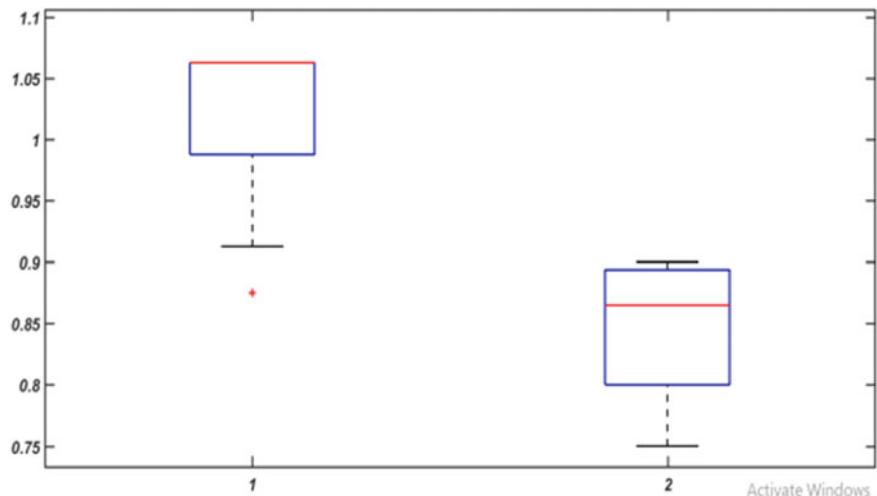


Fig. 5 Best values

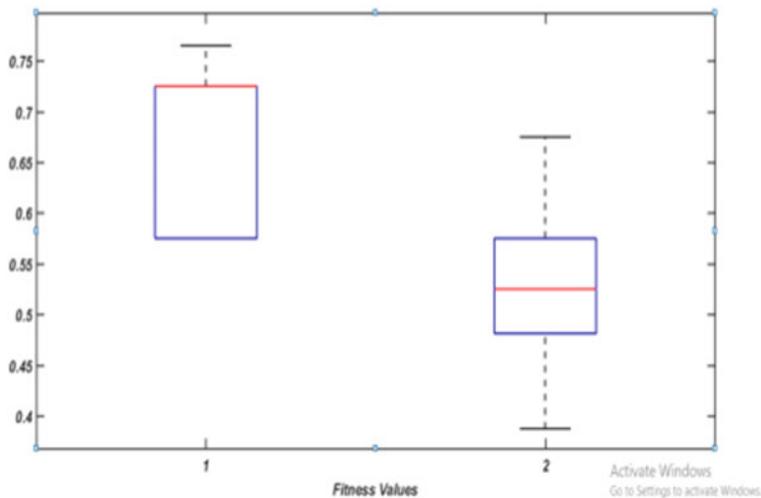


Fig. 6 Worst value

Figure 6 represents the worst value of APFD using with minimization and without minimization. It is clear from the analysis of Figs. 5, 6 that when we use the minimization technique, the size of test cases is reduced, and the fault rate is increased. It also leads to improve computational cost.

6 Conclusion

Regression testing is executed to confirm that changes are correct and do not affect the efficiency of the software. Many nature-inspired techniques have been applied for optimization purposes. In the present analysis, we have used CMIMX with the Artificial Bee Colony algorithm. CMIMX procedure is applied to the fault coverage matrix to reduce test cases and coverage of maximum faults. Further, the ABC algorithm is used for measuring the performance of APFD. The study considered fault coverage criteria to measure the efficiency of the proposed algorithm. Prima facie, it can be concluded from experiments that test suite minimization using the proposed approach show better results and also improves the value of the APFD metric. The computational time and cost were also reduced using the proposed approach.

References

1. Mathur A (2013) Foundations of software testing. Pearson, India
2. Khari M, Kumar P, Burgos D, Crespo RG (2018) Optimized test suites for automated testing using different optimization techniques. *Soft Comput* 22(24):8341–8352
3. Asthana M, Gupta KD, Kumar A (2020) Test suite optimization using Lion Search algorithm. In: Ambient communications and computer systems. Springer, Singapore, pp 77–90
4. Kumar G, Chahar V, Bhatia PK Software test case reduction and prioritization
5. Sheoran S, Mittal N, Gelbukh A (2019). Artificial bee colony algorithm in data flow testing for optimal test suite generation. *Int J Syst Assur Eng Manage* 1–10
6. Mittal S, Sangwan OP (2018) Prioritizing test cases for regression techniques using meta-heuristic techniques. *J Inf Optim Sci* 39(1):39–51
7. Anwar Z, Afzal H, Bibi N, Abbas H, Mohsin A, Arif O (2019) A hybrid-adaptive neuro-fuzzy inference system for multi-objective regression test suites optimization. *Neural Comput Appl* 31(11):7287–7301
8. Chetouane N, Wotawa F, Felbinger H, Nica M (2020). On using k-means clustering for test suite reduction. In: 2020 IEEE International conference on software testing, verification and validation workshops (ICSTW). IEEE, pp 380–385
9. Zamli KZ, Din F, Ahmed BS, Bures M (2018) A hybrid Q-learning sine-cosine- based strategy for addressing the combinatorial test suite minimization problem. *PloS ONE*, 13(5), e0195675
10. Kiran A, Butt WH, Shaukat A, Farooq MU, Fatima U, Azam F et al (2020) Multi-objective regression test suite optimization using three variants of adaptive neuro fuzzy inference system. *PLoS ONE* 15(12):e0242708. <https://doi.org/10.1371/journal.pone.0242708>
11. Bhatia PK (2020) Test case minimization in COTS methodology using genetic algorithm: a modified approach. In: Proceedings of ICETIT 2019. Springer, Cham, pp. 219–228
12. Singh L, Singh SN, Dawra S, Tuli R (2019). A new technique for test suite minimization in regression testing. In: Proceedings of 2nd International conference on advanced computing and software engineering (ICACSE)

Feature Selection for Bi-objective Stress Classification Using Emerging Swarm Intelligence Metaheuristic Techniques



Prableen Kaur, Ritu Gautam, and Manik Sharma

Abstract Stress is a major psychological disorder that conspicuously affects the psychological and physiological behavior of humans. Here, a dataset of MBA/MCA students is collected and analyzed to determine the overall rate of educational stress among these students. Seven different Swarm Intelligence (SI) based metaheuristic techniques, viz. Ant Lion Optimizer (ALO), Gray Wolf Optimization (GWO), Dragonfly Algorithm (DA), Satin Bowerbird Optimization (SBO), Harris Hawks Optimization (HHO), Butterfly Optimization Algorithm (BOA), Whale Optimization Algorithm (WOA) and one hybrid SI-based approach (WOA and Simulated Annealing (SA)) have been employed to find an optimal set of features for bi-objective stress diagnosis problem. As far as the stress classification rate is concerned, the hybrid swarm intelligence metaheuristic (WOA-SA) outperforms individual SI techniques as the use of simulated annealing in the amalgamation of WOA and SA improves the exploiting phase of the WOA. The results are also validated using the convergence rate and the Wilcoxon signed-rank test.

Keywords Stress · Classification · Feature selection · Swarm intelligence

1 Introduction

Stress is one of the major psychological disorders that generally affect the physical and mental state of the victim [1]. Nowadays, a significant number of folks are distressing from stress and the victims of this deadly disease are continually growing in an expedited manner [2]. In general, there are three different categories of stress called acute, episodic and chronic stress [3]. The studies witnessed that stress has a strong connection with other chronic and life-threatening human disorders [4]. The long term or continuous stress may lead to serious health problems. Moreover, it is one of the major reasons behind the growth of cardiac, stomach, psychological, lung and reproductive disorders in human beings [5].

P. Kaur · R. Gautam · M. Sharma (✉)

Department of Computer Science and Automation, DAV University, Jalandhar, India

SI-based computing techniques have been pre-eminently and successfully used to solve a thick variety of real-life applications. Query optimization [6], disease diagnosis [7], stock prediction [8], supply chain management [9], inventory control [10] are some of the major application areas for these techniques. The leftover part of this section will briefly present the literature related to stress, feature selection and SI metaheuristic techniques [11, 12].

In the last three decades, several computing techniques have been designed and used for early and precise diagnosis of different chronic and life-threatening human disorders [12, 13]. Swarm Intelligence techniques play a striking role in the diagnosis of assorted human disorders such as heart problems, cancer, and diabetes [14]. SI metaheuristics are emerging nature-inspired computing techniques that present the social behavior of different creatures (nature) to solve different real-life problems. These techniques are becoming more prevalent in solving a variety of healthcare applications also. The effectiveness of these approaches lies in easy to implement approach; no gradient information is required and can be used in multidisciplinary areas.

In this work, eight different SI metaheuristic methods have been engaged in a dataset (MBA/MCA students) to find the optimal feature set for bi-objective stress diagnostic problem. There are three principal benefactions of this work. First of all, the performance of eight different SI-based feature selection techniques, viz. Ant Lion Optimizer (ALO), Gray Wolf Optimization (GWO), Whale Optimization Algorithm (WOA), Dragonfly Algorithm with levy Flight (DA), Butterfly Optimization Algorithm (BOA), and the hybrid approach of WOA and simulated annealing in solving bi-objective stress diagnostic problem has been evaluated. The different performance metrics, viz. accuracy, fitness, dimensions and execution time has been computed and analyzed. Finally, the results of different SI-based feature selection metaheuristics techniques have been validated using the convergence rate and Wilcoxon signed-rank test. To the best of our knowledge, no one has explored the performance of these SI-based feature selection techniques for the stress-related dataset.

2 Material and Methods

2.1 Data

To determine the performance of the SI-based feature selection metaheuristic techniques in solving bi-objective stress diagnosis problem a dataset comprises 315 instances and 29 attributes, seven attributes for personal details (such as gender, marital status, annual father's income, postgraduate course, year, of course, drug/alcohol intake, socially active) and other 22 attributes with stress-related details (such as level of stress due to studies, academic marks, level of satisfaction with grades, learning capabilities, attendance, stress about career, extra classes to cover the syllabus, the pressure of studies you can handle, performance appreciated by teacher

and parents, **lack** of interest, interest in sports, confidence while taking admission, confusion level regarding duties, size of social circle, financial stress on studies, level of drug/alcohol intake, rate of sound sleep, sleeping disorder, rate of disturbance in sleeping after an early wake-up). The responses for these factors were recorded in the form of categorical variables (six-point psychometric scale) such as 1-very high, 2-high, 3-medium, 4-low, 5-very low and 6-no stress). Out of 315 instances, 212 are MBA students and the rest of 103 are MCA students. The dataset comprises different physical, independent and psychological characteristics of the MBA/MCA students.

2.2 Mathematical Model of Feature Selection

Feature selection is a paramount stage of data pre-processing that extracts a subset of a feature from the original set of data. Feature selection is an extraction process that selects a meaningful subset of the feature in context to the concerned optimization problem (Canedo, 2015) (Arora, 2020). Mathematically,

Given

$$\text{Original Dataset } (F_0): (F_1, F_2, F_3, F_4, \dots, F_n) \quad (1)$$

Objective

$$\text{To Extract Subset } (F_{SE}): (F_1, F_2, F_3, \dots, F_m) \quad (2)$$

where F_1, F_2, \dots, F_n are attributes or features of the dataset. ‘m’ and ‘n’ are integers such that $m < n$.

A brief picture of the feature selection process used in this manuscript is depicted in Fig. 1.

It is observed that the complete scenario of the feature selection process revolved around five entities, viz. original dataset, a subset of selected features, evaluation mechanism, selection criterion and finally the validations. Technically, feature selection is a bi-objective optimization dilemma where a balance has to be made between the number of features selected and the rate of classification. The number of features should be minimum, whereas the rate of classification is a maximization problem. The bi-objective fitness function is used in this research work is given below [15–18].

$$F_{\text{objective}} = \alpha \gamma_R(D) + (1 - \alpha) \frac{|R|}{|N|} \quad (3)$$

where $\gamma_R(D)$ represents the rate of classification error of the K-nearest neighbor classifier. $|R|$ and $|N|$ represent a number of selected and total features respectively and α belong to $[0, 1]$.

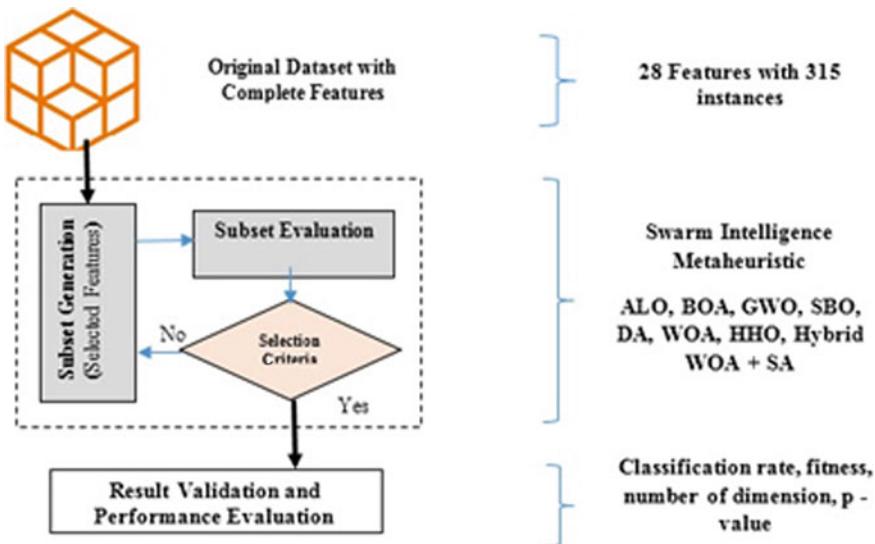


Fig. 1 Working of feature selection process

2.3 Methods

In the last three decades, several swarm-intelligent techniques (based on natural phenomenal) have been designed and employed to solve different real-life optimization problems. Based upon the principles used in the design of swarm-intelligent computing techniques, these can be broadly classified as physics-based, chemistry-based and biology-based techniques [19, 20]. In this manuscript, eight different swarm intelligence algorithms (ALO [21], BOA [22], GWO [23], DA [24], HHO [25], SBO [26], and WOA [27]) have been implemented for selecting optimal features for the diagnosis of stress among MBA/MCA students.

3 Results and Discussions

3.1 Data Analysis

A 5-point Likert scale has been used to point out the rate of stress among students. Figure 2 shows the number of students that falls under different categories of stress. It is observed that most of the students are victims of high and medium stress. The percentage of the victims of medium and high stress is 39.36% and 59.36% respectively. Fortunately, the number of victims of serious stress is too low. There are only four students who are suffering from this kind of stress. Unfortunately, no one

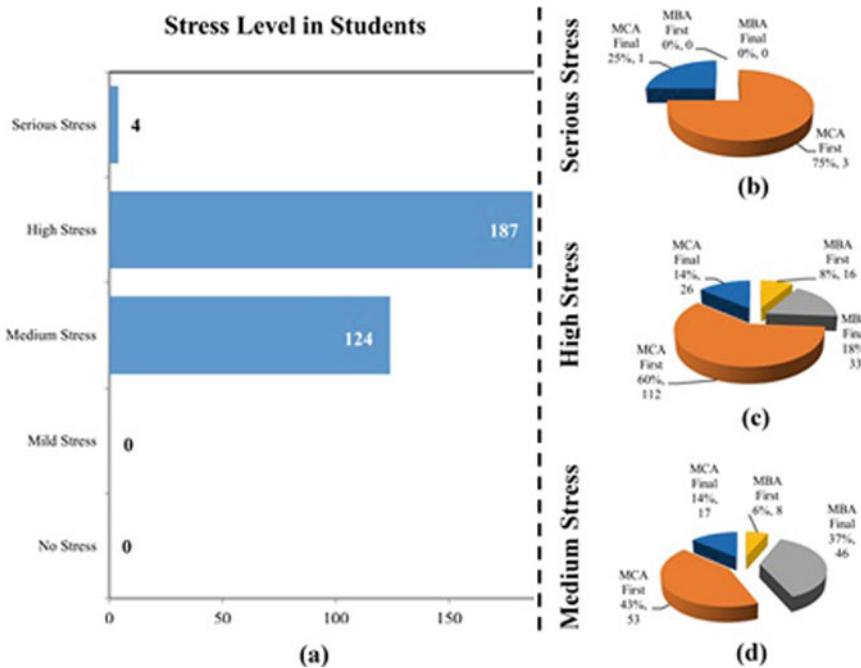


Fig. 2 **a** Number of students in different stress categories, **b** Serious stress level in MBA and MCA first and final year students, **c** High-stress level in MBA and MCA first and final year students, **d** Medium stress level in MBA and MCA first and final year students

is free from stress. The bifurcation of the students that fall into different categories is presented in Fig. 2. Figure 2a shows the number of students suffering from different stress levels. However, Fig. 2b, c and d present the bifurcation of students in the respective categories based on course and course year.

3.2 Performance Analysis

Here, several experiments were carried out over the above-mentioned dataset using seven individual (BOA, ALO, HHO, GWO, DA, SBO and WOA) and one hybrid (WOA-SA) SI metaheuristic techniques. The results are mentioned in Tables 1, 2, 3, 4 and 5. The optimal values have been marked in bold. The performance of WOA-SA has been compared with the outcomes of the BOA, ALO, HHO, GWO, DA, SBO and WOA itself. It is found that minimum, average and maximum rate of stress classification obtain using an amalgamation of WOA and SA is better than other SI techniques like BOA, ALO, HHO, GWO, DA SBO and WOA.

Table 1 Minimum rate of classification

BOA	ALO	HHO	GWO	DA	SBO	WOA	WOA-SA
0.778	0.835	0.841	0.803	0.810	0.841	0.816	0.867
0.791	0.829	0.841	0.822	0.797	0.835	0.816	0.879
0.784	0.816	0.835	0.829	0.816	0.854	0.822	0.879
0.791	0.822	0.835	0.822	0.829	0.841	0.810	0.873
0.810	0.810	0.841	0.822	0.835	0.835	0.803	0.848
0.797	0.829	0.848	0.835	0.841	0.841	0.803	0.854
0.810	0.835	0.848	0.835	0.841	0.854	0.822	0.879
0.778	0.810	0.835	0.803	0.797	0.835	0.803	0.848
0.792	0.823	0.840	0.822	0.821	0.841	0.812	0.867

Table 2 Maximum rate of classification

BOA	ALO	HHO	GWO	DA	SBO	WOA	WOA-SA
0.860	0.905	0.892	0.886	0.886	0.917	0.873	0.943
0.860	0.905	0.911	0.898	0.879	0.911	0.898	0.930
0.892	0.879	0.924	0.905	0.879	0.911	0.898	0.924
0.873	0.898	0.911	0.911	0.905	0.917	0.879	0.924
0.860	0.905	0.898	0.905	0.917	0.911	0.898	0.917
0.867	0.905	0.898	0.930	0.886	0.917	0.892	0.936
0.892	0.905	0.924	0.930	0.917	0.917	0.898	0.943
0.860	0.879	0.892	0.886	0.879	0.911	0.873	0.917
0.869	0.899	0.907	0.906	0.892	0.913	0.889	0.929

Table 3 Average rate of classification

BOA	ALO	HHO	GWO	DA	SBO	WOA	WOA-SA
0.817	0.867	0.868	0.857	0.851	0.881	0.849	0.899
0.826	0.860	0.869	0.864	0.851	0.872	0.854	0.902
0.835	0.854	0.874	0.869	0.851	0.880	0.864	0.903
0.832	0.861	0.879	0.869	0.858	0.881	0.847	0.904
0.837	0.854	0.873	0.869	0.862	0.872	0.857	0.892
0.834	0.860	0.872	0.874	0.865	0.881	0.855	0.897
0.837	0.867	0.879	0.874	0.865	0.881	0.864	0.904
0.817	0.854	0.868	0.857	0.851	0.872	0.847	0.892
0.830	0.859	0.873	0.867	0.856	0.878	0.854	0.900

Table 4 Minimum fitness values

BOA	ALO	HHO	GWO	DA	SBO	WOA	WOA-SA
0.142	0.102	0.112	0.119	0.117	0.088	0.129	0.063
0.142	0.101	0.094	0.106	0.124	0.093	0.108	0.076
0.110	0.128	0.084	0.101	0.125	0.092	0.107	0.080
0.131	0.107	0.094	0.093	0.102	0.088	0.126	0.080
0.143	0.099	0.103	0.101	0.088	0.093	0.108	0.087
0.135	0.101	0.104	0.074	0.118	0.088	0.115	0.067
0.143	0.128	0.112	0.119	0.125	0.093	0.129	0.087
0.110	0.099	0.084	0.074	0.088	0.088	0.107	0.063
0.134	0.106	0.098	0.099	0.112	0.090	0.116	0.076

Table 5 Average number of dimension

BOA	ALO	HHO	GWO	DA	SBO	WOA	WOA-SA
12.1	21.3	16.35	19.8	17.3	16.05	21.45	16.4
12.8	19.55	15.65	20.45	17.95	14.95	18.65	15.8
13.25	22.15	17.85	20.75	17.3	15.35	19.7	16.1
14.5	22	17.3	20.2	16.9	16.05	19.15	16.25
13.9	20.7	16.5	20.15	17.3	14.95	19.9	15
12.75	18.1	15.8	20.2	17.95	16.05	19.05	15.25
14.5	22.15	17.85	20.75	17.95	16.05	21.45	16.4
12.1	18.1	15.65	19.8	16.9	14.95	18.65	15
13.21667	20.63333	16.575	20.25833	17.45	15.56667	19.65	15.8

3.2.1 Fitness Value

Here, the solution of the bi-objective stress diagnosis problem is delineated as a vector (one-dimensional), and each element of the vector contains either zero or one. The length of the vector is restricted to the number of features/attributes in the dataset. One characterizes the presence of the attribute and the absence of a feature is presented with zero. KNN classifier is employed to assess the value of the fitness function.

However, when there is a concern of selecting a minimum number of features only, then the performance of BOA is found to be excellent.

3.2.2 Dimensions

The dimension corresponds to the optimal solution of bi-objective stress diagnosis problem obtained using (BOA, ALO, HHO, GWO, DA, SBO and WOA) and hybrid (WOA-SA) are outlined in Table 5. For dimensions, the findings of BOA are better than other SI metaheuristic techniques.

4 Conclusion

The research revealed that fortunately, the percentage of students suffering from serious stress is too low (0.012%). However, most of the students suffer from a high and medium level of stress. The percentage of the victims of medium and high stress is 39.36% and 59.36% respectively. Different performance metrics like fitness, classification rate, minimum, maximum, mean to solve the stress diagnostic problem are computed and examined. For this bi-objective problem, the hybrid approach of WOA and SA was found to be more precise and effective as compared to other SI-based feature selection metaheuristic techniques, i.e., ALO, BOA, GWO, DA, SBO, HHO, WOA itself. The use of SA in WOA-SA improved the exploiting phase by locating the most competent region pointed out by WOA. However, when there is a concern of selecting a minimum number of features only, then the performance of BOA is found to be excellent. Finally, the results are validated using one of the major non-parametric tests called the Wilcoxon signed-rank test. In future, the effectiveness of these algorithms can be evaluated for other human psychological disorders like depression, anxiety, schizophrenia, insomnia, etc.

References

1. Kaur P, Sharma M (2019) Diagnosis of human-psychological disorders using supervised learning and nature inspired computing techniques: a meta-analysis. *J Med Syst* 43:204
2. Nieuwenhuijsen K, Bruinvelds D, Frings-Dresen M (2010) Psychosocial work environment and stress-related disorders, a systematic review. *Occup Med* 60(4):277–286
3. Reda A (1994) Sources and levels of stress in relation to locus of control and self esteem in university students. *Educ Psychol* 14(3):323–330
4. Sharifi-Rad M et al (2020) Lifestyle, oxidative stress, and antioxidants: back and forth in the pathophysiology of chronic diseases. *Front Physiol* 11:694
5. Salari N, Hosseiniyan-Far A, Jalali R et al (2020) Prevalence of stress, anxiety, depression among the general population during the COVID-19 pandemic: a systematic review and meta-analysis. *Glob Health* 16:57
6. Sharma M, Singh G, Singh R (2018) CDSS query optimizer using hybrid Firefly and controlled Genetic algorithm. *J King Saud Univ-Comput Inf Sci*
7. Poo MM, Du JL, Ip NY, Xiong ZQ, Xu B, Tan T (2016) China brain project: basic neuroscience, brain diseases, and brain-inspired computing. *Neuron* 92(3):591–596

8. Yusof Y, Mustaffa Z. (2015). Time series forecasting of energy commodity using grey wolf optimizer. In: Proceedings of the international multi conference of engineers and computer scientists (IMECS'15), vol 1, p 1
9. Auhar SK, Pant M (2015) Genetic algorithms, a nature-inspired tool: review of applications in supply chain management. In: Das K, Deep K, Pant M, Bansal J, Nagar A (eds) Proceedings of fourth international conference on soft computing for problem solving. Advances in intelligent systems and computing, vol 335. Springer, New Delhi, pp 71–86
10. Kumar SK et al (2013) Logistics planning and inventory optimization using swarm intelligence: a third party perspective. *Int J Adv Manuf Technol* 65(9–12):1535–1551
11. Kaur K, Kumar Y (2020) Swarm intelligence and its applications towards various computing: a systematic review. In: 2020 International conference on intelligent engineering and management (ICIEM), pp 57–62
12. Gautam R, Kaur P, Sharma M (2019) A comprehensive review on nature-inspired computing algorithms for the diagnosis of chronic disorders in human beings. *Prog Artif Intell* 1–24
13. Kaur P, Sharma M (2018) Analysis of data mining and soft computing techniques in prospecting diabetes disorder in human beings: a review. *Int J Pharm Sci Res* 9(7):2700–2719
14. Sharma M, Singh G, Singh R (2017) Stark assessment of lifestyle based human disorders using data mining based learning techniques. *IRBM* 36(6):305–324
15. Schiezaro M, Helio P (2013) Data feature selection based on Artificial Bee Colony algorithm. *EURASIP J Image Video Process* 2013(1):47
16. Bolón-Canedo V, Sánchez-Maróño N, Alonso-Betanzos A (2015) A critical review of feature selection methods. In: Feature selection for high-dimensional data. Artificial intelligence: foundations, theory, and algorithms. Springer, Cham
17. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF (2020) A review of unsupervised feature selection methods. *Artif Intell Rev* 53:907–948
18. Hancer E, Xue B, Zhang M (2020) A survey on feature selection approaches for clustering. *Artif Intell Rev* 53:4519–4545
19. Kaur P, Sharma M (2017) A survey on using nature-inspired-computing for fatal-disease diagnosis. *Int J Inf Syst Model Des* 8(2)
20. Himabindu K, Jyothi S (2017) Nature-inspired computation techniques and its applications in soft computing: survey. *Int J Res Appl Sci Eng Technol* 5(VII):1906–1915
21. Mirjalili S (2015) The Ant Lion optimizer. *Adv Eng Softw* 83:80–98
22. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization, *Soft Comput* 23(3)
23. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf optimizer. *Adv Eng Softw* 69:46–61
24. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27:1053–1073
25. Heidaria AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
26. Moosavi SHS, Bardsiri VK (2017) Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation. *Eng Appl Artif Intell* 60:1–15
27. Mirjalili S, Lewis A (2016) The Whale optimization algorithm. *Adv Eng Softw* 95:51–67

Regulated Energy Harvesting Scheme for Self-Sustaining WSN in Precision Agriculture



Kunal Goel and Amit Kumar Bindal

Abstract In case of precision agriculture-based WSN, energy consumption may vary due to different parameters (i.e., dynamic computational overload/sensor density variations) and traditional energy harvesting scheme does not consider these conditions during harvesting and thus may reduce the overall lifespan of the network. In order to meet the current energy requirements of WSN, an energy harvesting scheme can regulate itself as per the current energy requirements of the WSN. In this paper, a regulated energy harvester will be introduced to overcome the above-discussed constraint and its performance will be analyzed using various performance parameters (throughput/residual energy/harvested energy, etc.).

Keywords PA · WSN · Energy harvesting · LEACH · TEH · NEH and REH

1 Introduction

Precision agriculture (PA) offers the automation of various agricultural activities using sensors to improve the quality crops as well as it also optimizes the overall operating cost. A field monitoring and parameter collection lead to long-lasting data processing, due to limited energy backup, it is not feasible to require the external power support, and it can be achieved using the concept of energy harvesting [1–4].

Following are the causes that may reduce the lifespan of sensors:

- Sensors may be deployed in large agricultural land having heterogeneous environment; in such case, enough battery capacity is required to transmit the signals at distant level.
- Enormous and complex computations may decline the overall lifespan of sensors.

K. Goel · A. K. Bindal (✉)

Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar (Deemed To Be University), Mullana-Ambala, Haryana 133-207, India

e-mail: amitbindal@mmumullana.org

- Battery can be depleted due to vibrant environment as sensors may engage in excessive data collection followed by transmission.
- Energy resources may be unnecessarily exhausted due to packet retransmission/route discovery/packet collision, etc.

In order to fulfill the energy retirement of PA, following are the available options:

- Periodic battery replacement.
- Using the ambient energy resources.

Constraints for the implementation of energy harvester w.r.t PA are discussed below:

- Sensors have low battery backup but their energy may be executed rapidly due to random transmission, so there is need to synchronize both energy harvesting cycle and transmission cycle.
- Adaption of compatible energy harvesting source is another major issue in PA. Harvesting using natural resource experiences the uncertain weather/environment conditions, usage of mechanical sources suffer from heat/energy loss during conversion process, performance of thermal harvesters faces the ambiguous variations in temperature and RF harvesting has also its own limitations due to dependency over external sources.

All above-discussed factors raise the need of the optimization of energy harvesting and power consumption [5, 6].

2 Literature Survey

In this section, various energy harvestings over WSN have been discussed. Goel et al. [7] describe the various sensors to measure the various parameters and discussed the concept of microbial fuel cell for the energy harvesting for the nodes used in precision agriculture. Goel et al. [8] describe the scheme for optimal consumption of energy in precision agriculture with respect to various parameters like throughput, residual energy, energy consumption, no. of alive and dead nodes. Ali et al. [9] developed a solution for smart irrigation using photovoltaic energy that is utilized to operate sensors in a specific agricultural land. Experiments show that it is the cheapest solution that can optimize the usage of natural resources as well as it can also reduce the overall farming cost. López et al. [10] developed an energy harvesting prototype model for WSN-based PA applications. It consists of two different modules, i.e., lens module to capture sunlight and a thermoelectric to handle electric signals produced through temperature difference. Experimental results indicate that its maximum energy generation capacity is limited up to 6.93 mW only and it can be implemented as a low-cost energy harvesting solution. Sadowski et al. [11] investigated the performance of different types of wireless systems (low powered WAN/Zigbee/WiFi) with energy harvesting capabilities for smart farming. Analysis

shows that each energy consumption of each wireless system differs as per data processing, it is highest for WiFi system followed by Zigbee, and it is optimal for low powered WAN. Saxena et al. [12] developed a solar energy-based harvesting solution that forwards the collected field data using Internet of Things (IoT). Experiments show that sensors may engage in different network operations, i.e., crop/water management/pesticide level/climate monitoring, etc. thus may lead to energy depletion and shorten network lifetime. Outcomes indicate that proposed solution can provide real-time data with the support of energy harvesting as well as productivity of agricultural land can be further improved. Shatar et al. [13] integrated the solar power with thermoelectric module to handle the load variations due to the change in the direction of sunlight/weather conditions, etc. Lab-based experimental results show that it is able to fulfill the power requirements of agricultural sector under the constraints of energy consumption. However, battery charging state under the constraints of real weather conditions is still an open issue. Escolar et al. [14] developed a solar energy harvesting prototype using low powered wide area network for smart agriculture. Experiments found the limitations of such type of network, i.e., compatibility issues between routing protocol and end-user terminals, variations in packet transmission/sampling frequencies, etc. Its performance analysis under the constraints of transmission delay/collisions/different data rates in real-time environment is still an open issue. Ikeda et al. [15] developed a prediction model for energy harvesting over WSN. It uses the temperature difference between air and the soil surface to produce electric signals. Prediction model forecasts the energy variations in energy requirement over an interval. Real-time experimental results indicate that it outperforms in terms of higher prediction accuracy/forecast errors as compared to traditional harvesting schemes. Sadowski et al. [16] incorporated the solar panels with sensors nodes to fulfill the energy harvesting requirements of smart farming. It continuously monitors the battery level and variations in voltage and under these constraints data is collected and finally, transmitted to base station. Experimental results show that it is suitable for different agricultural applications. Dhillon et al. [17] investigated the role of different energy sources (Wind/Solar/Vibration) that can be utilized for energy harvesting for smart farming using WSN. Study found that energy produced using all these sources varies as per field/weather conditions and if a prediction model is used to predict the current energy level, harvested energy.

3 Regulated Energy Harvesting for WSN

Wireless Sensor Network: WSN,
Energy Level Threshold: elTh,
Energy: e,
Current Energy Level: cElvl,
Current Load: Cld,
Load Type: Ld,
Constant: C,

Variable: V ,
MAC Filter: M_f ,
Transmission: T_s ,
Harvesting Status: H_{vs} ,
Harvesting Update Interval: HvU ,
Total energy required: $Etr =$,
Battery Voltage: Bv ,
Battery Capacity: Bc ,
Discharge/Charge rate: Cr .

Step 1: Initialize WSN
Step 2: Define Type (Ld)
Step 3: If Ld-Type!=C else go to step 5:
Step 4: If (Check(cElvl)> elTh
 Set (T_s , True)
 Else
 Set (T_s , False)
 Check(Cld, Etr)
 Set (H_{vs} , True, HvU)
 Repeat step: 4
 End if
End if
Step 5: If Ld-Type==V
 Check(cElvl)
 Predict(HvU,Cld,Dcr)
 Set (H_{vs} , True, HvU)
 If (Check(Etr)> elTh
 Set (T_s , True)
 Else
 Set (T_s , False)
 Repeat step: 5
 End if
End if
Proc Predict(HvU,Cld,Cr)
{
 1 cr=60 minutes for charge or discharge of a battery,
 time t: 60 minutes/Cr
 // if cr =20c then 60/20c=3 minutes are required for charge/discharge
 // and if ex is energy unit consumed for n packets then exi=ex/n is the en-
 ergy unit for packet x
 update (HvU, t)
}

In case of agriculture, sensors may produce data in a constant manner or data may be generated randomly, in case of constant data transmission, energy consumption remains approx. constant, but energy requirements for dynamic data transmission vary which rises the need of a harvesting scheduler to fulfill current energy requirements. Following are the steps of the proposed scheduler:

First of all, initialize the WSN and define the current load type that may be constant or variable in nature and battery charge/discharge/lifetime is affected by this load type. In case of constant load, the current energy level is checked, it should be enough to manage the transmission; otherwise, it is paused.

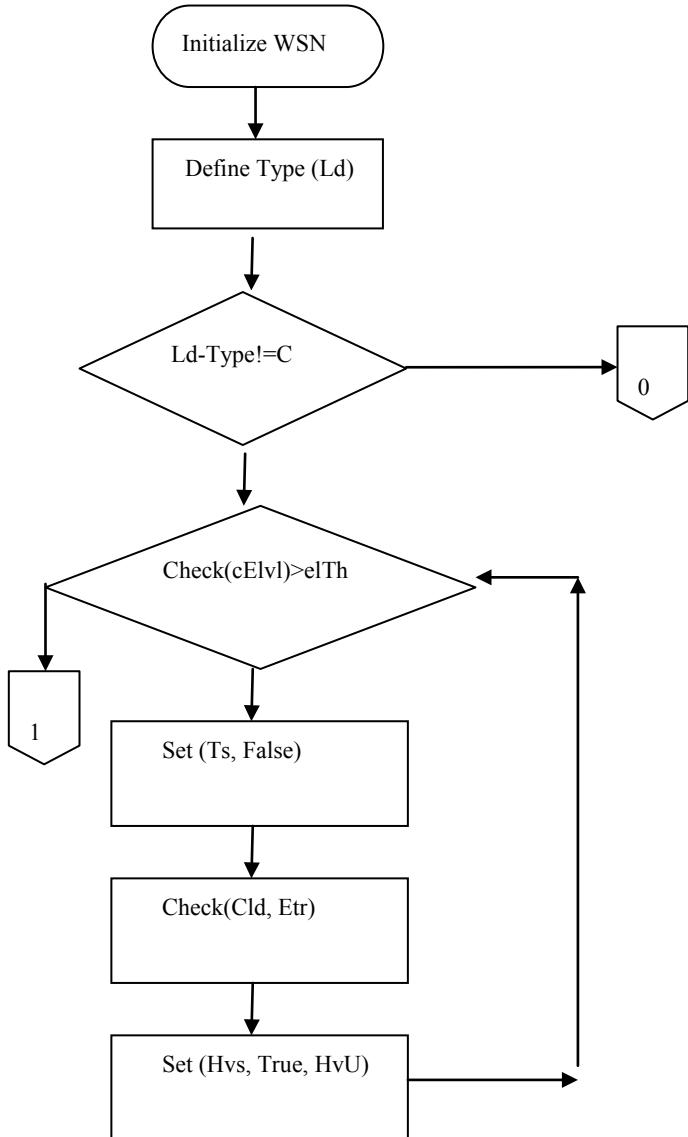
Total energy required to carry out the current load is calculated and an energy harvesting interval is set to meet the current energy requirements.

The above step may be repeated if required.

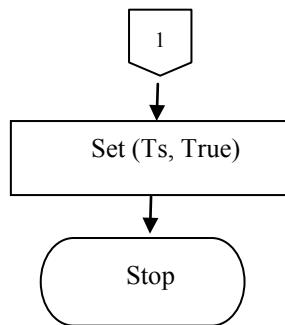
On other hand, in case of variable load, current energy level is checked as well as a predicate is also defined using different parameters, i.e., harvesting update interval, current load and discharge/charge rate, etc. to determine the variations in energy requirement w.r.t. current load.

Finally, energy harvesting is turned on for a specific interval w.r.t. total energy required, and current energy level is also calculated for transmission as shown in Flow Charts given below.

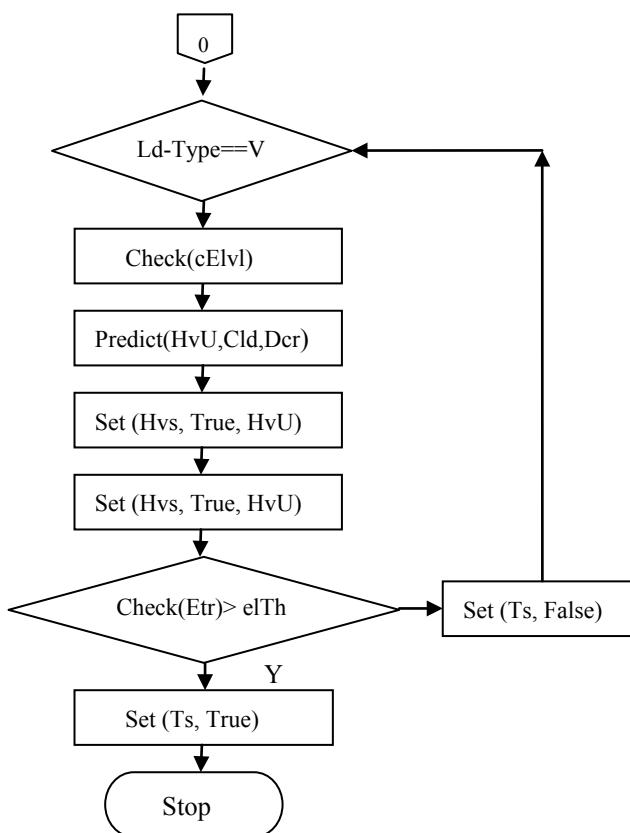
Flow Chart: Initial Steps of REH



Flow Chart 1: Start Transmission



Flow Chart 0: Steps of Energy Harvesting Using REH



4 Simulation Configuration and Performance Analysis of NEH/TEH/REH Using Leach Protocol

See Table 1.

Figure 1 shows the variations in throughput using LEACH protocol w.r.t. different harvesting scenarios under the constraint of sensor density that varies from 100–300 only. It can be observed that using NEH scenario, with 100–300 sensor density, it remains lowest as compared to other scenarios (TEH/REH). In case of TEH scenario, with 100–300 sensor density, it is slightly higher as compared to NEH scenario but remains low than REH scenario. It can be analyzed that using REH scenario, with sensor density 100, it is highest and slightly degraded as sensor density varies from 200–300.

However, it is also degraded using other scenarios also with sensor density 100/200/300, using NEH, it is $34,807.88/25,163.79/27,521.96$ Kbps. In case of TEH, it is $34,830.44/25,501.95/25,439.32$ Kbps. In case of REH, it is $34,830.44/25,501.95/25,439.32$ Kbps.

Figure 2 shows the variations in remaining energy level using LEACH protocol w.r.t. different harvesting scenarios under the constraint of sensor density that varies from 100–300 only. It can be observed that it is minimum using NEH scenario w.r.t. sensor density and it is marginally improved with TEH scenario and is successfully retained using REH which is highest as compared to others.

Table 1 Simulation configuration

Simulation parameters	Parameter values
Routing protocols	LEACH
Terrain size	1500*300
MAC protocol	802.11
Node density	100/200/300
Propagation model	Friss
Data type	Constant bit rate (CBR)
Sampling interval	1 ms
Simulation time	10 s
Network simulator	NS-3.31
Initial energy	6 J
txPower/rxPower	7.5
Energy harvester type	Basic energy harvester
Simulation scenarios	A. No Energy Harvesting (NEH) B. Traditional Energy Harvesting Scheme (TEH) C. Regulated Energy Harvesting (REH)

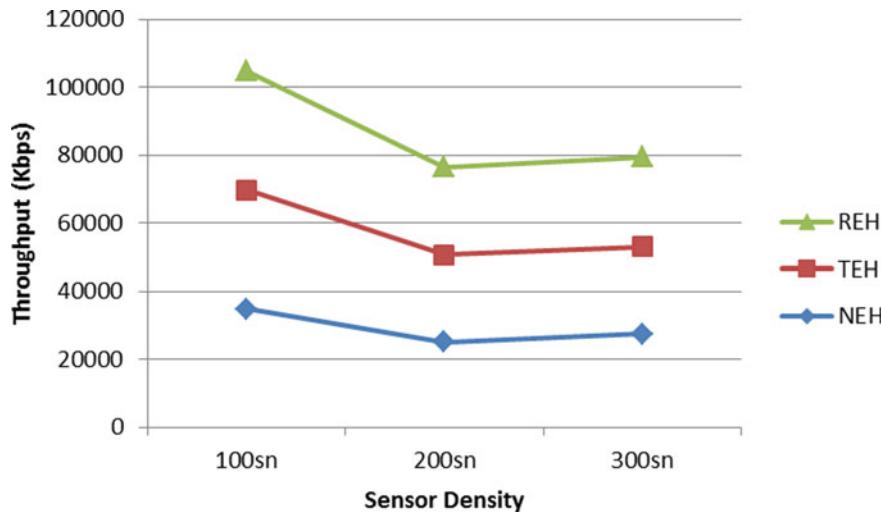


Fig. 1 Throughput w.r.t. harvesting scenarios (NEH/TEH/REH)

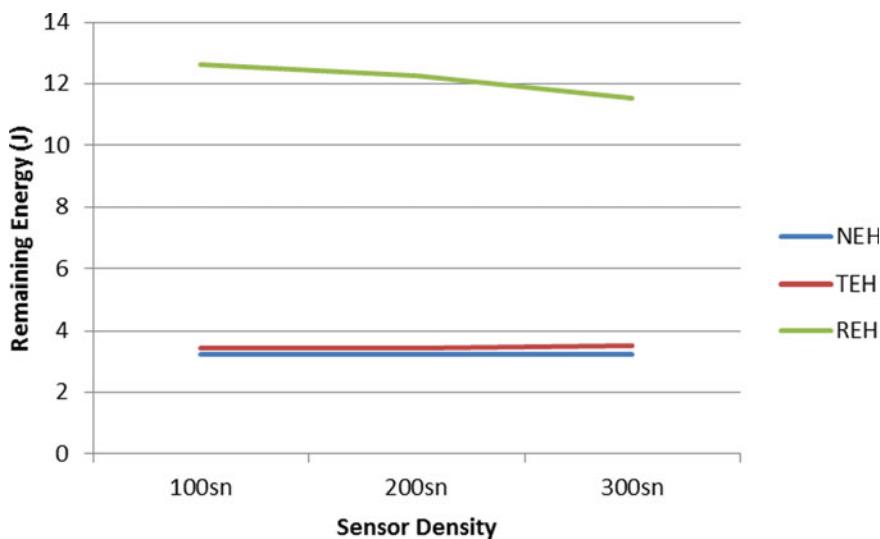


Fig. 2 Remaining energy w.r.t. harvesting schemes (NEH/TEH/REH)

With sensor density 100/200/300, using NEH, it is 3.22235 J where as in case of TEH, it is 3.45557/3.45495/3.51836 J and in case of REH, it is 12.6482/12.2447/11.5333 J.

Figure 3 shows the number of alive sensor w.r.t. sensor density with different energy harvesting scenarios using LEACH protocol. It can be observed that REH/TEH both offer higher number of alive sensors as compared to NEH scenario.

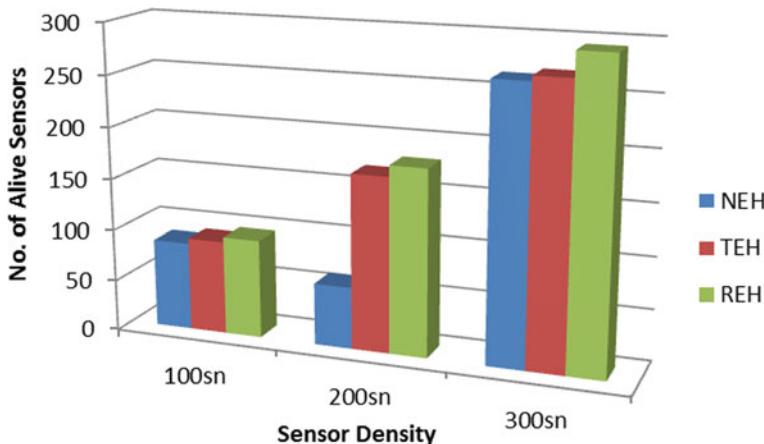


Fig. 3 No. of alive sensors w.r.t. harvesting schemes (NEH/TEH/REH)

With sensor density 100/200/300, using NEH, number of alive sensors is 86/60/265. In case of TEH, these are 91/169/270 and in case of REH, these are 96/180/294.

Figure 4 shows the number of dead sensor w.r.t. sensor density with different energy harvesting scenarios using LEAC H protocol. It can be observed that using REH/TEH, there is less number of dead sensors as compared to NEH scenario.

With sensor density 100/200/300, using NEH, number of dead sensors is 14/40/35. In case of TEH these are 9/31/30 and in case of REH, 4/20/6.

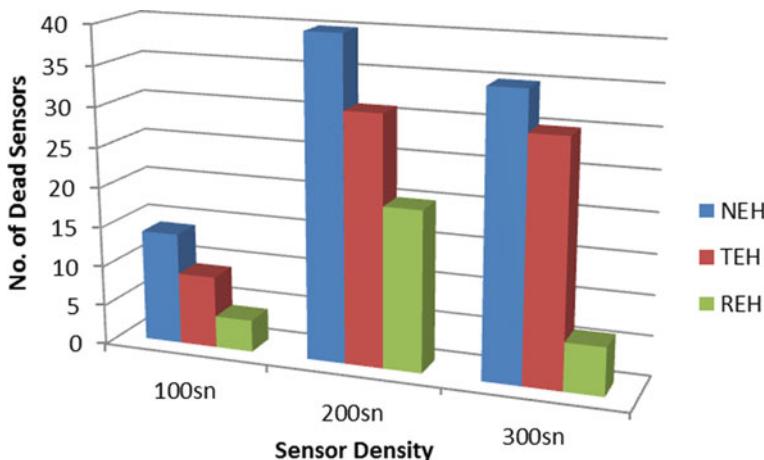


Fig. 4 No. of dead sensors w.r.t. harvesting schemes (NEH/TEH/REH)

4.1 Energy Harvesting Variation Analysis of TEH

Figure 5 shows the variation in energy harvesting level (Watt) over the simulation interval using TEH w.r.t. 100 sensor density. It can be analyzed that it varies till the end of simulation.

Figure 6 shows the variation in energy harvesting level (Watt) over the simulation interval using TEH w.r.t. 200 sensor density. It can be analyzed that it varies during

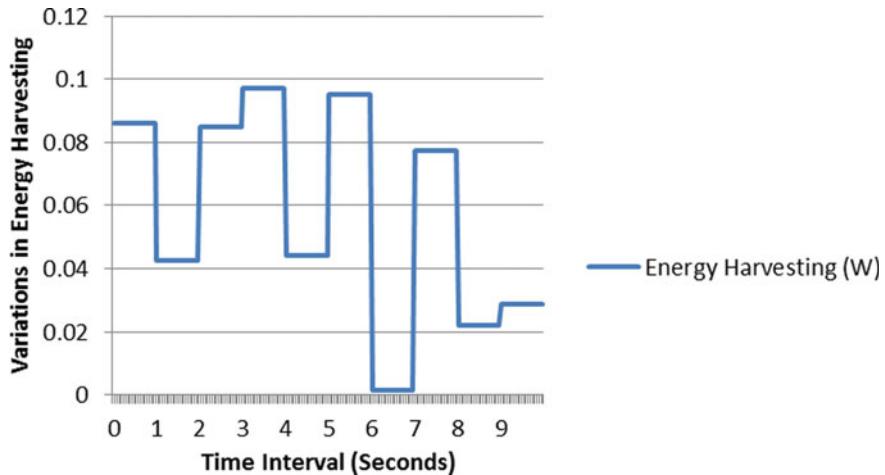


Fig. 5 Variations in current energy harvesting using TEH w.r.t. sensor density (100)

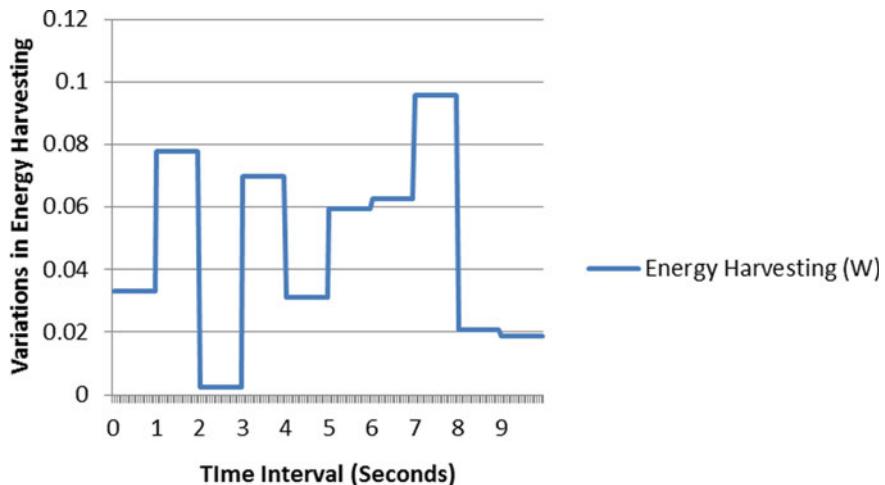


Fig. 6 Variations in current energy harvesting using TEH w.r.t. sensor density (200)

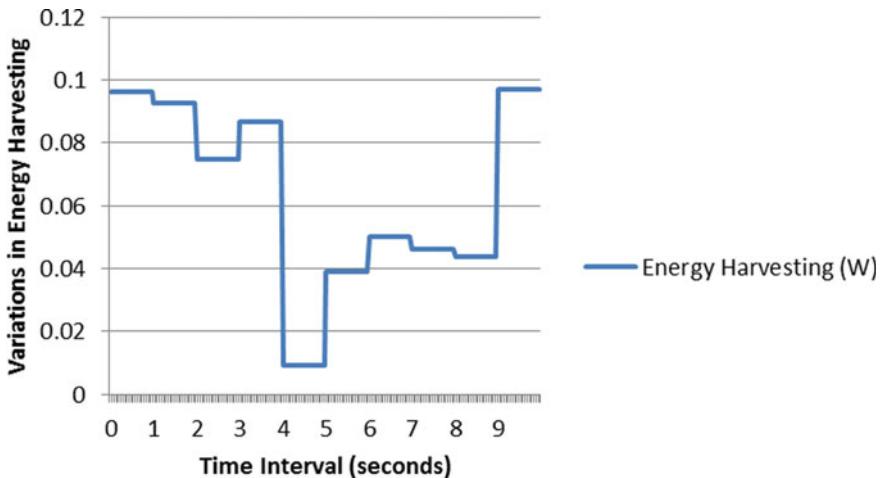


Fig. 7 Variations in current energy harvesting using TEH w.r.t. sensor density (300)

simulation and reaches up to its peak level, and after that, there is sharp declined in its level.

Figure 7 shows the variation in energy harvesting level (Watt) over the simulation interval using TEH w.r.t. 300 sensor density. It can be analyzed that it varies during simulation and reaches up to its deepest level, and after that, there is increasing in a constantly till the end of simulation.

4.2 Energy Harvesting Variation Analysis of REH

Figure 8 shows the variation in energy harvesting level (Watt) over the simulation interval using REH w.r.t. 100 sensor density. It can be analyzed that it varies till the end of simulation.

Figure 9 shows the variation in energy harvesting level (Watt) over the simulation interval using REH w.r.t. 200 sensor density. It can be analyzed that it sharply declined and reaches up to its lowest level, and after that, it is gradually increasing and retains its highest level.

Figure 10 shows the variation in energy harvesting level (Watt) over the simulation interval using TEH w.r.t. 300 sensor density. It can be analyzed that there are lot of variations in its level. Sometimes it decreases in a constant manner and reaches up to its lowest level and after few intervals, it reaches up to its peak value till the end of simulation.

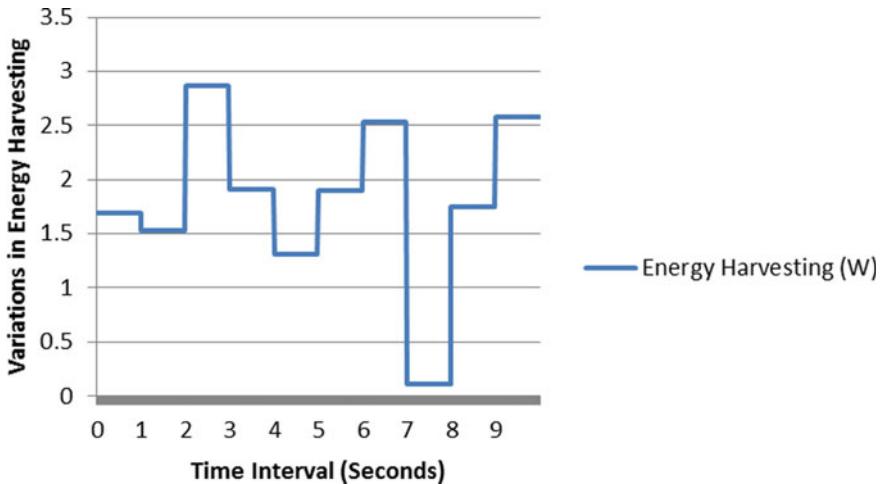


Fig. 8 Variations in current energy harvesting using REH w.r.t. sensor density (100)

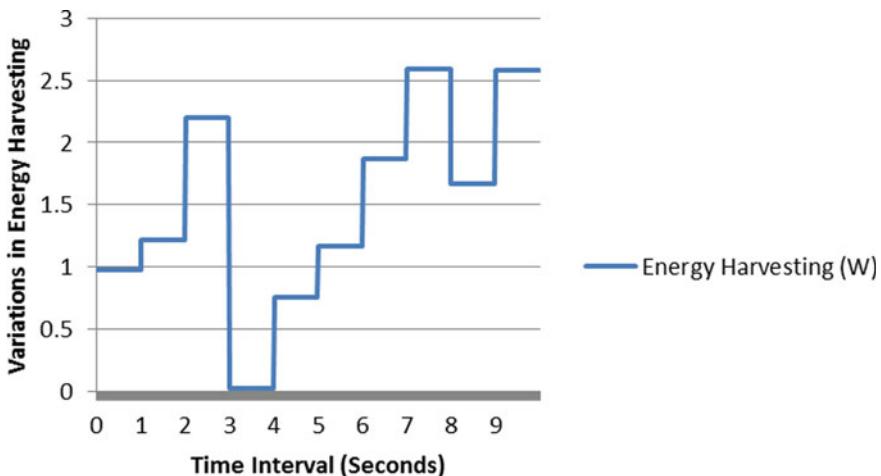


Fig. 9 Variation in current energy harvesting using REH w.r.t sensor density (200)

4.3 Analysis of Total Energy Harvested Using TEH

Figure 11 shows the total harvested energy using TEH scenario w.r.t. 100 sensor density. It can be studied that TEH did the harvesting till the end of simulation.

Figure 12 shows the total harvested energy using TEH scenario w.r.t. 200 sensor density. It can be investigated that TEH did the harvesting till the end of simulation.

Figure 13 shows the total harvested energy using TEH scenario w.r.t. 300 sensor density. It can be examined that TEH did the harvesting till the end of simulation.

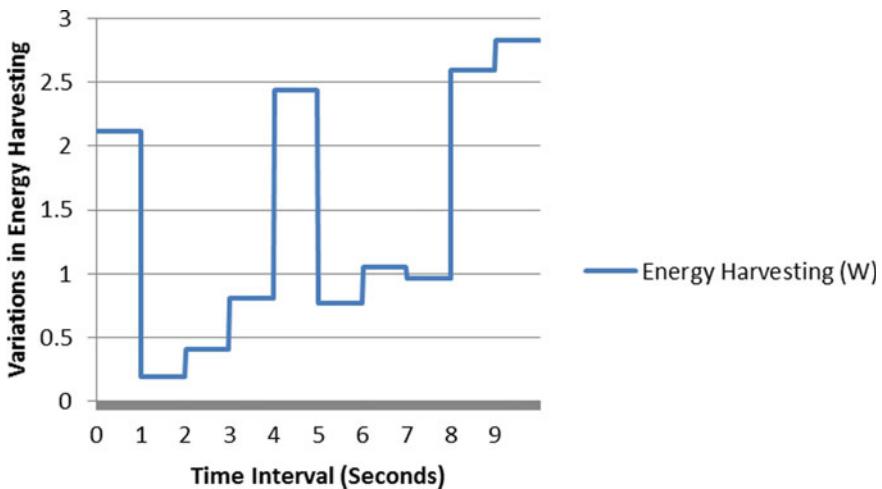


Fig. 10 Variation in current energy harvesting using REH w.r.t sensor density (300)

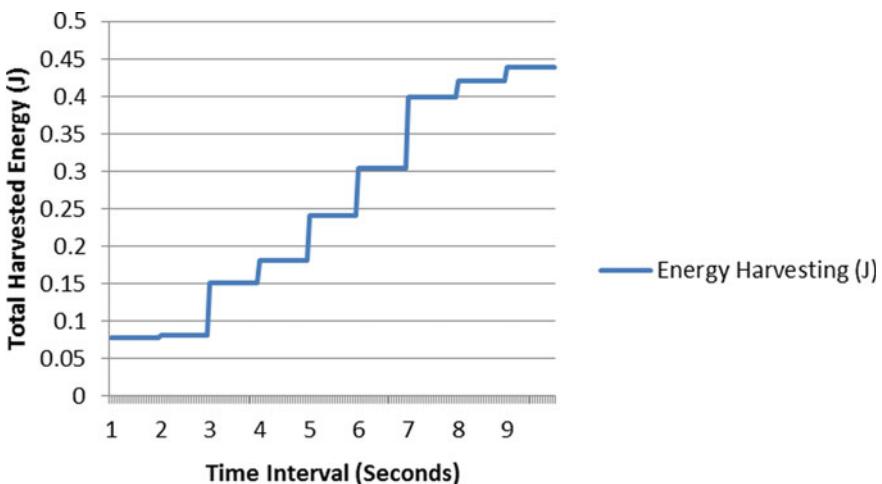


Fig. 11 Total harvested energy (Joules) using TEH w.r.t. sensor density (100)

4.4 Analysis of Total Energy Harvested Using REH

Figure 14 shows the total harvested energy using REH scenario w.r.t. 100 sensor density. It can be studied that REH did the harvesting till the end of simulation but its value varies as per each interval.

Figure 15 shows the total harvested energy using REH scenario w.r.t. 200 sensor density. It can be analyzed that REH did the harvesting till the end of simulation but its value varies as per each interval.

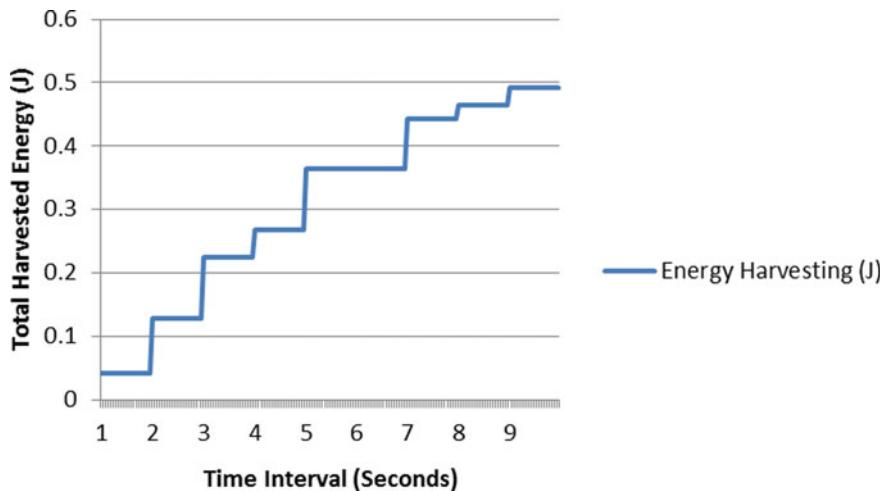


Fig. 12 Total harvested energy (Joules) using TEH w.r.t. sensor density (200)

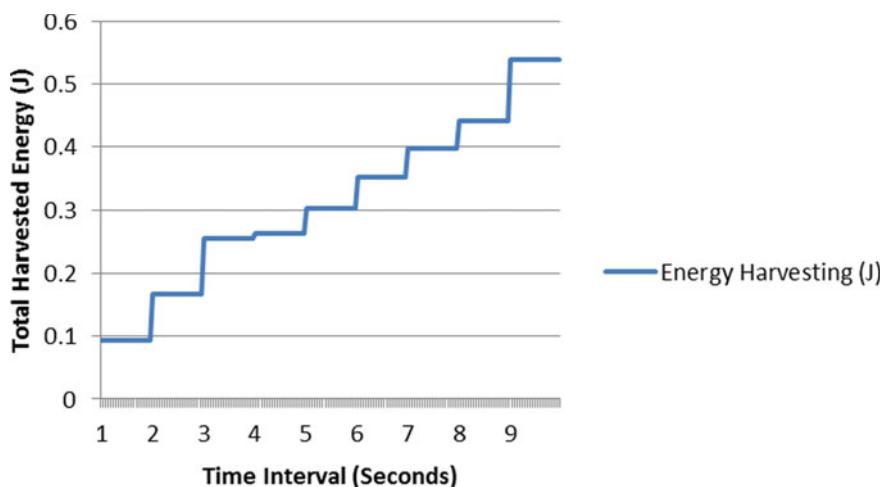


Fig. 13 Total harvested energy (Joules) using TEH w.r.t. sensor density (300)

Figure 16 shows the total harvested energy using REH scenario w.r.t. 300 sensor density. It can be analyzed that REH did the harvesting till the end of simulation but its value varies as per each interval.

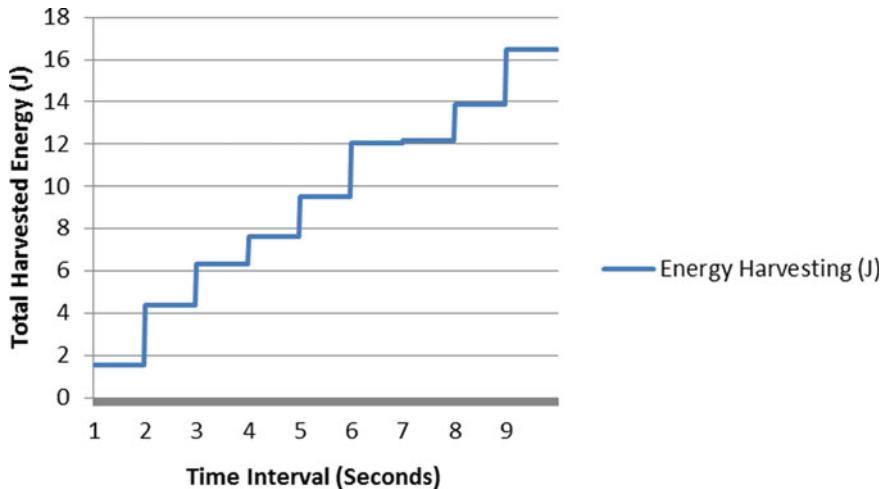


Fig. 14 Total harvested energy (Joules) using REH w.r.t. sensor density (100)

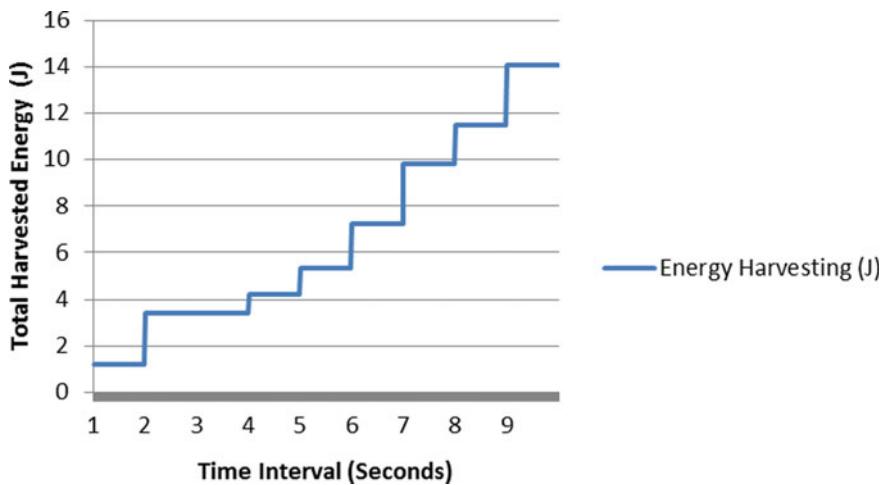


Fig. 15 Total harvested energy (Joules) using REH w.r.t. sensor density (200)

4.5 Comparison of Maximum Harvested Energy Using Both Scenarios (TEH and REH)

Figure 17 shows the comparison of maximum harvested energy using different scenarios (TEH/REH). It can be observed that REH offers highest energy harvesting level as compared to TEH w.r.t. sensor density (100–300). However, its level is

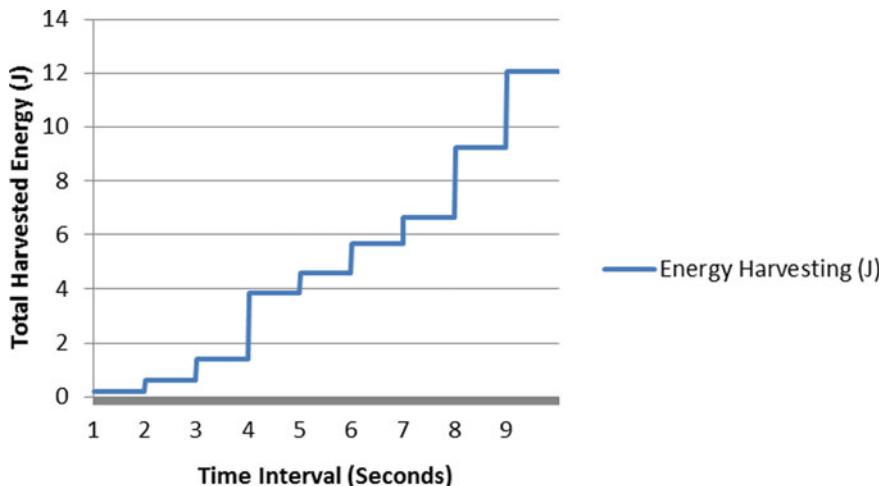


Fig. 16 Total harvested energy (Joules) using REH w.r.t. sensor density (300)

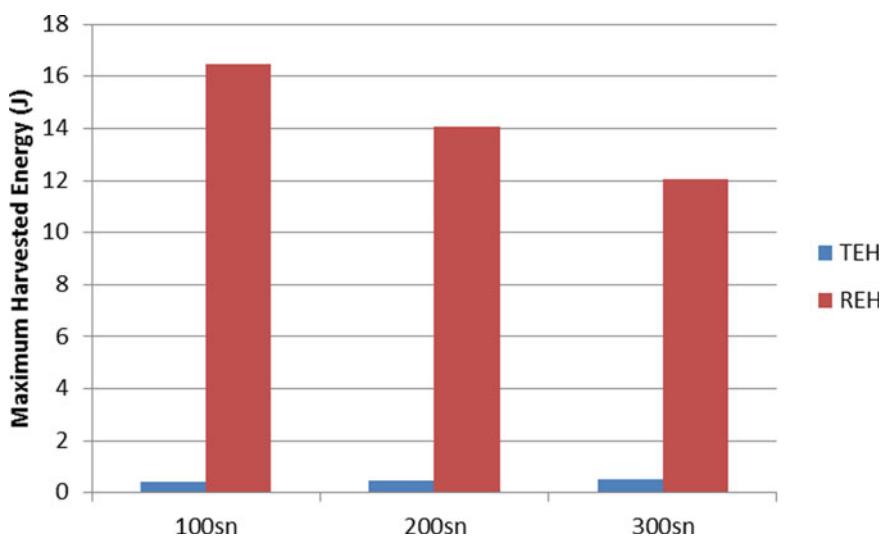


Fig. 17 Comparison of maximum harvested energy using both schemes

slightly decreasing w.r.t. sensor density and comes to its lowest level with 300 sensor density.

In case of sensor density 100, it is 0.438394 J using TEH and it is 16.4804 J using REH. In case of sensor density 200, it is 0.492984 J using TEH and it is 14.0743 J using REH. In case of sensor density 300, it is 0.539009 J using TEH and it is 12.0496 J using REH.

5 Conclusion

In this paper, issues and solutions-related energy requirements for agriculture-based WSN were explored and a regulated energy harvester was introduced. Its performance was analyzed using LEACH protocol under the constraints of sensor density variations (100–300) with various performance parameters, i.e., throughput, remaining energy, variations in energy harvesting, total harvested energy and maximum harvested energy level. w.r.t. sensor density (100–300), in case of NEH scenario, without energy harvesting LEACH protocol delivered lowest which is further boosted using TEH scenario and finally, REH delivered highest Throughput that also varies under sensor density constraints. Residual energy remained at its constant level using NEH whereas it slightly varies using TEH and using REH, it is highest as compared to others. Using LEACH with all scenarios, number of alive/dead sensors varies w.r.t. sensor density and REH offered higher lifespan for sensors as compared to TEH and NEH. Variation analysis of energy harvesting using TEH and REH shows that TEH tried to harvest the WSN as usual and there were little bit variations in the energy harvesting levels whereas REH provided the energy harvesting support as per current energy requirements of the WSN. Analysis of total harvested energy using TEH/REH shows that REH used frequent harvesting cycles w.r.t. interval and network energy demand and it also successfully managed the maximum harvesting level as compared to TEH.

References

1. Gulec O, Haytaoglu E, Tokat S (2020) A novel distributed CDS algorithm for extending lifetime of WSNs with solar energy harvester nodes for smart agriculture applications. *IEEE Access* 8:58859–58873
2. Rajasekaran T, Anandamurugan S (2018) Challenges and applications of wireless sensor networks in smart farming—a survey. In: *Advances in big data and cloud computing*, Part of the *Advances in intelligent systems and computing* book series, vol 750. Springer, pp 353–361
3. Titri S, Izeboudjen N (2020) WSN based smart farm powered by solar energy harvesting technique. In: *International conference in artificial intelligence in renewable energetic systems, artificial intelligence and renewables towards an energy transition*. Springer, pp 798–807
4. Honc D, Merta J (2020) Smart, precision or digital agriculture and farming—current state of technology. In: *International workshop on soft computing models in industrial and environmental applications*, 15th International conference on soft computing models in industrial and environmental applications. Springer, pp 245–254
5. Tang X, Wang X, Cattley R, Gu F, Ball AD (2018) Energy harvesting technologies for achieving self-powered wireless sensor networks in machine condition monitoring: a review. *Sensors*, 1–39
6. Dhall R, Agrawal H (2018) An improved energy efficient duty cycling algorithm for IoT based precision agriculture. *Procedia Comput Sci* 141:135–142
7. Goel K, Bindal AK (2018) Wireless sensor network in precision agriculture: a survey report. In: *2018 Fifth International conference on parallel, distributed and grid computing (PDGC)*. IEEE, pp 176–181
8. Goel K, Bindal AK (2020) Optimal energy scheme in precision agriculture to prolong the lifespan of nodes in WSNs. *J Green Eng* 10(10)

9. Ali S, Saif H, Rashed H, AlSharqi H, Natsheh A (2018) Photovoltaic energy conversion smart irrigation system-Dubai case study (Goodbye overwatering & waste energy, Hello water & energy saving). In: IEEE 7th World conference on photovoltaic energy conversion (WCPEC) (A Joint conference of 45th IEEE PVSC, 28th PVSEC & 34th EU PVSEC). IEEE, pp 2395–2398
10. López JJE, Atoche AAC, Sinencio ES (2019) Design and fabrication of a 3-D printed concentrating solar thermoelectric generator for energy harvesting based wireless sensor nodes. *IEEE Sens Lett* 3(11):1–4
11. Sadowski S, Spachos P (2020) Wireless technologies for smart agricultural monitoring using internet of things devices with energy harvesting capabilities. *Comput Electron Agric* 172:1–8
12. Saxena M, Dutta S (2020) Improved the efficiency of IoT in agriculture by introduction optimum energy harvesting in WSN. In: International conference on innovative trends in information technology. IEEE, pp 1–5
13. Shatar NM, Abdul Rahman MAA, Shaikh Salim SAZ, Ariff MHM, Muhtazaruddin MN, Badilisah AKA (2018) Design of photovoltaic-thermoelectric generator (PV-TEG) hybrid system for precision agriculture. In: 2018 IEEE 7th International conference on power and energy (PECon). IEEE, pp 50–55
14. Escolar S, Rincón F, del Toro X, Barba J, Villanueva FJ, Santofimia MJ, Villa D, López JC The PLATINO experience: a LoRa-based network of energy-harvesting devices for smart farming. In: XXXIV conference on design of circuits and integrated systems (DCIS). IEEE, pp 1–6
15. Ikeda N, Shigeta R, Kawahara Y (2019) Energy prediction for energy management method of sensor node powered by temperature difference between air and shallow underground soil. *IEEE Sens* 1–4
16. Sadowski S, Spachos P (2018) Solar-powered smart agricultural monitoring system using internet of things devices. In: 9th Annual information technology, electronics and mobile communication conference (IEMCON). IEEE, pp 18–23
17. Dhillon SK, Madhu C, Kaur D, Singh S (2020) A review on precision agriculture using wireless sensor networks incorporating energy forecast techniques. *Wirel Press Commun* 113:2569–2585

Empirical Analysis of Facial Expressions Based on Convolutional Neural Network Methods



Rohit Pratap Singh and Laiphakpam Dolendro Singh

Abstract Facial expression is actually the main visual indication for the analysis of the underlying human emotions. A machine with stronger intelligence in emotional recognition can understand human beings better and communicate more naturally. It is therefore not surprising that recognition of facial expression by computers or smart devices has become a topic of recent research. The variation in expression, background, location, and label noise makes automatic recognition of facial expression images difficult for computers. In this paper, different models of CNN have been incorporated for empirical analysis of Facial Expression Recognition (FER) problems. To this end, Sequential, VGG16, Resnet50, were incorporated. These are applied to the Kaggle FER Challenge dataset. The best method yields a test set accuracy of 68.95%.

Keywords Facial expression · Computer vision · Deep learning · Emotion recognition

1 Introduction

Facial expressions are very essential in our day to day social communications between humans. Face shows emotions and hence it is an important medium of non-verbal communication. The face has the capability of showing obvious emotion as facial expressions. Happiness can be understood by a smiling face, a frown face expresses disapproval or sadness, a face with wide-open eyes indicates surprise and a curled-lip face reveals disgust. The recognition of these signals by machines will strengthen and comport human-machine interactions.

R. P. Singh (✉) · L. D. Singh
CSED, NIT Silchar, Silchar, Assam, India
e-mail: ldsingh@cse.nits.ac.in

FER is a key non-verbal medium by which Human-Machine Interaction (MMI) systems are capable of recognizing human intention and emotion. There are a number of places where FER is used, such as human machine interactions for surveillance, video games, and social robots. FER is also used in behavioral science to gain information about society (origin, sex, and age) and to track pain, depression, anxiety, and treatment of mental retardation in the medical sciences.

While humans can easily interpret the majority of facial expressions, accurate expression recognition by means of machines is still difficult to get. FER's main problems include optimal pre-processing, feature extraction, and classification, particularly under varying circumstances of input data, head position, environmental disorders, and illumination, etc. Automatic perception and recognition of human emotions were one of the main issues in the interaction between humans and computers. Despite ongoing research, precise recognition of facial expression under unregulated conditions remains a major challenge.

It is inherently a multidisciplinary enterprise that involves a broad range of related areas, including computer vision, linguistics, speech analysis, robotics, cognitive psychology, etc. [33]. In this paper different CNN architectures are incorporated to check the efficiency of the different models on this particular problem by performing experiments on FER-2013 dataset. The main task is to categorize given image into seven main emotions using our CNN model and then automatically achieve efficient classification.

The rest of this paper is written as follows. Section 2 reviews the related work in FER. In Sect. 3, the system architecture is discussed. Section 4 explains the experimental results and discussions. Section 5 gives the concludes the paper.

2 Related Work

In the last two decades, automatic recognition of facial emotions has gained growing interest. Most of the research exploring emotion recognition was dominated by conventional methods with handcrafted features some of which are the Facial Action Coding System with action units (AUs) [29, 31], sparse learning [36] and local binary patterns (LBPs) [5, 24, 35].

Tian et al. [29] introduced a system for automatic analysis of face in order to realize emotions which describes very light changes in a face into action units of Facial Action Coding System, containing permanent and transitory facial features. A boosted-LBP feature has been implemented by Shan et al. [24] which represents the discriminative LBP features which afterward incorporated Support Vector Machine (SVM) [4] classifiers to recognize emotions employing the extracted Boosted-LBP features. The approach is good for facial expressions with low resolution. Complex semantic relations among facial action units (SAUs) have been utilized by Wang et al. [31] employed restricted Boltzmann machine (RBM) to recognize the facial emotions.

The authors of [7] demonstrated that the use high-dimensional features of the images generated by dense and census transformed vectors [32] yielded high performance results. Zhong et al. [36] reported that that only a few facial components are advantageous in facial expression recognition. They proposed a two-stage multitask sparse learning system.

A novel classification tree was introduced in [3], based on the sparse coding [18]. A deep architecture by which face expressions were modeled by incorporating a set of local AU features is presented in [16]. Although the aforementioned traditional methods have demonstrated excellent recognition of emotion recognition extracting the face features from the datasets created in laboratory regulated settings, the robustness of these approaches is not adequate when facial images include numerous head position adjustments, which can prevent the extraction of useful AUs or LBP features by traditional methods CNNs have been unveiled to research fraternity for more than three decades [10]. However, they have only recently become a prevalent approach in tasks of image classification. The handling large number of parameters by a system while training a model is not a limiting factor nowadays; thanks to the emergence of very large amount of classification datasets, increases in computation power, and improvement in the algorithms to training the models.

Deep CNNs, recently, have been used to solve a variety of image classification problems, like object recognition [8, 20], face verification [23, 26, 27], scene recognition [9, 37], age and gender classification [12], and many more. End-to-end CNNs have been recently proven to be extremely effective in dealing with image classification problems [1, 19, 21, 22].

Enhancing the image can help in improving the model's accuracy and robustness in general. The rotating and deviating face images are not used as an input in these CNN models. As a result, these models are not robust to highly variable posture in images of facial expressions. Tang [28] employed a CNN in connection with a linear SVM. Liu et al. [17] presented the Boosted Deep Belief Network (BDBN), a unified framework for integrating three training stages through joint training. This methodology focus on joint training boosts the abilities of facial feature selection and expression classification. Liu et al. [15] utilized a common method to ensemble several different networks, with the end result being the average of their model outputs. Kahou et al. [6] created a system for combining four modality models.

Zhang et al. [34] suggested a method for recognizing pose-variant facial expression and body pose, as mentioned above. Furthermore, the generative adversarial network was implemented in this network to enlarge and enrich the training data-set by producing facial expression images in various poses. Such approaches, however, analyze emotion features from a facial expression data-set without taking into consideration contextual details from the environment.

Bazrafkan [2] demonstrated that when various databases were used for training and testing, the accuracy in emotion prediction decreased. Lee et al. [11] investigated the function of contextual information in recognition of emotions. They suggested deep CNN-based approach for detecting emotions by combining contextual and facial features. As compared to approaches that focuses on facial expression features, this approach generates better emotion recognition accuracy.

3 Dataset Description

In this work, we have used Facial Expression Recognition dataset [13, 30]. The dataset was first released in the ICML Representation Learning Challenge held in 2013 [13]. The data-set mainly consists of csv file with 2 columns namely, Emotions and Pixels. For the Emotions column, we have 7 different categories. These are 0 for ‘Angry’, 1 for ‘Disgust’, 2 for ‘Fear’, 3 for ‘Happy’, 4 for ‘Sad’, 5 for ‘Surprise’ and 6 for ‘Neutral’. The Pixels column in the dataset contains a string containing the image’s pixel values, separated by spaces. The dataset comprises grayscale images and the dimensions of all of the images are 48*48 pixels. The number of the images present in the training set is 28,709. There are 3589 images present in validation set and also equal number of images, i.e., 3589 images are there in the test set (Fig. 1).

4 System Architecture

In our proposed system we have used sequential model, VGG16 and resnet50. The job is to classify each face into one of the seven labels mentioned above based on the emotion expressed. Facial Expression Recognition has three stages namely, Pre-processing, Extraction of the features, and Classification as shown in Fig. 4. The pre-processing stage consists of preparing the dataset into a form that can be fed to the model at time of training. The image is resized to eliminate any unwanted elements. This decreases the amount of memory available and speeds up computing. In feature extraction, CNN inputs an image and returns a set of probabilities linked to seven distinct emotion types as output. In general, the size of the input image fetched by a CNN trained on ImageNet is 224×224 , 227×227 , 256×256 , and 299×299 ; other dimensions, though, can also be thought about. In classification of the facial expression, The model classifies the image as Neutral, Disgust, Surprise, Anger, Happiness, Fear, and Sadness, as labeled in the FER2013 dataset (Fig. 2).



Fig. 1 Some samples from FER dataset

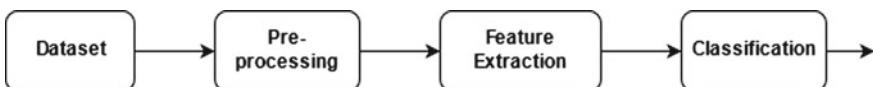


Fig. 2 Research methodology

We have designed three different CNN models for our experiments. The first one is Sequential Model, the Second one is VGG-16 and the third one is Resnet50 model. The sequential model that we constructed is made up of three layers: 3 convolutional layers, 3 pooling layers, and one dense layer. Convolution of the features with the weights of the filter is performed by the Convolutional Layer. Following the convolutional layers is a Batch Normalization Layer, that increases the network's learning rate. The Pooling Layer samples the data after the Activation Layer adds non-linearity into the network. Following that, certain random neurons are removed to avoid over-fitting. The training of this model has been done using the training dataset of the data of FER2013 dataset. The model is trained for 40 epochs. For this experiment VGG-16 [25] model from Visual Geometry Group [23] is used which has been pre-trained on ImageNet dataset. VGG-16 has won the 2014 ImageNet competition. ResNet50 [14] is a ResNet model version with 48 Convolution layers, 1 MaxPool layer, and has 1 Average Pool layer. Moreover, it is a famous ResNet model which is most widely used.

5 Experimental Results and Discussions

In this paper, we concentrated on recognition of the facial expression and used CNN (Sequential, VGG-16, ResNet50) to separate facial images into distinct emotion classes. The models are pre-trained using the ImageNet dataset. The graphs below show the loss and accuracy for each epoch, and it can be shown that as the loss reduces, the accuracy improves.

Figure 5 depicts the confusion matrix created over the test results. The blocks along the diagonal indicate that the test data are well categorized. The numbers on each side of the diagonal reflect the number of incorrectly labeled images. Since these numbers are smaller than the numbers on the diagonal, we should assume that the algorithm worked correctly and produced advanced-level results.

Table 1 lists the accuracy obtained by various approaches on the FER-2013 dataset to analyze the performance of the models. The accuracy of the Sequential model is 60.10%, VGG-16 model achieved accuracy of 65.12 %, and ResNet50 model tops the table with 68.95 % accuracy (Figs. 3, 4 and 5).

Table 1 Accuracy of different models on FER-13 dataset

S. No.	CNN models	Accuracy (%)
1	ResNet50	68.95
2	VGG16	65.12
3	Sequential	60.10

Fig. 3 The training and validation accuracy curves on FER dataset

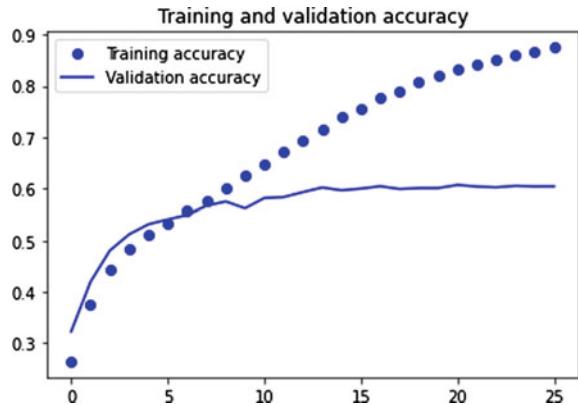
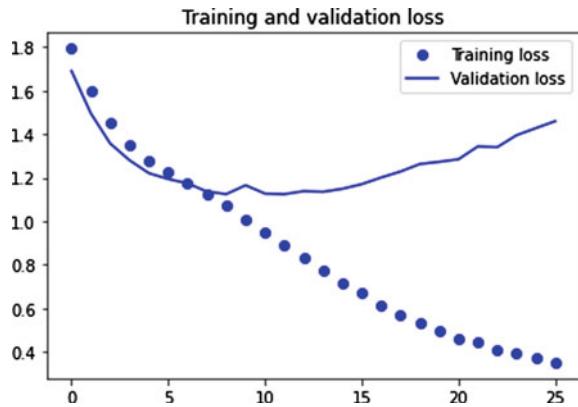


Fig. 4 The training and validation loss curves on FER dataset



6 Conclusion

This paper demonstrates the FER which has been trained using VGG-16, Resnet50, and Sequential models. The proposed model takes face images as input and generates the expression or emotion as an output. For well trained model, different pre-processing operations have been performed. Afterward, the pre-processed image data has been put into the different pretrained CNN models. Accuracy scores metric has been used to analyze the model performance. The accuracy of ResNet50 models was found to be the best out of the three models under study with an accuracy of 68.96%. VGG-16 and Sequential models get second and third place with accuracy of 65.12% and 60.10% respectively. In future work, ensembles of state-of-the-art deep neural networks will be deployed to improve system performance. Attention mechanisms will also be incorporated to enhance the accuracy of the model.

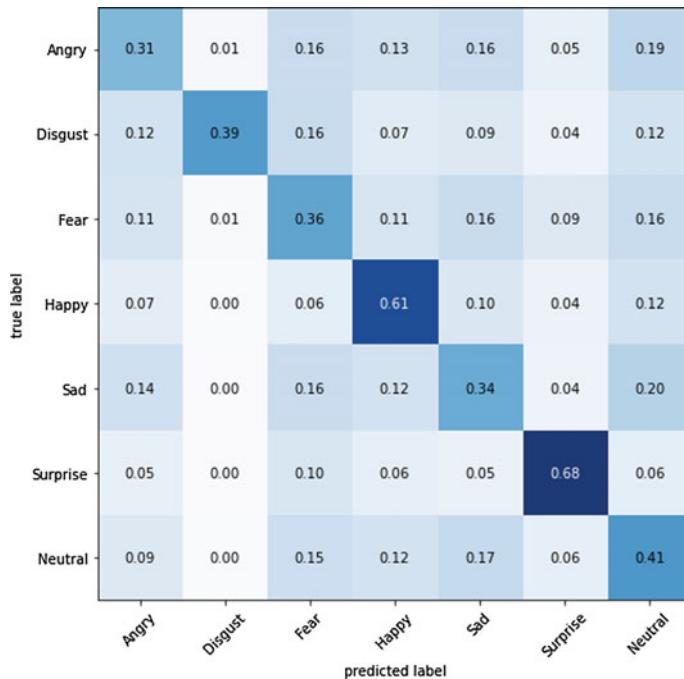


Fig. 5 Confusion matrix of sequential model on FER

References

- Agarwal M, et al (2021) A study on image analysis and recognition using learning methods: CNN as the best image learner. In: Data analytics and management, pp 23–30. Springer (2021)
- Bazrafkan S, Nedelcu T, Filipczuk P, Corcoran P (2017) Deep learning for facial expression recognition: a step closer to a smartphone that knows your moods. In: 2017 IEEE international conference on consumer electronics (ICCE). IEEE, pp 217–220
- Chen K, Comiter MZ, Kung H, McDanel B (2015) Sparse coding trees with application to emotion classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 77–86
- Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
- Huang D, Shan C, Ardabili M, Wang Y, Chen L (2011) Local binary patterns and its application to facial image analysis: a survey. IEEE Trans Syst Man Cybern Part C (Appl Rev) 41(6):765–781
- Kahou SE, Bouthillier X, Lamblin P, Gulcehre C, Michalski V, Konda K, Jean S, Froumenty P, Dauphin Y, Boulanger-Lewandowski N et al (2016) Emonets: multimodal deep learning approaches for emotion recognition in video. J Multimodal User Interfaces 10(2):99–111
- Kahou SE, Froumenty P, Pal C (2014) Facial expression analysis based on high dimensional binary features. In: European conference on computer vision. Springer, pp 135–147
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
- Kumar A, Verma S (2021) CapGen: A neural image caption generator with speech synthesis. In: Data analytics and management. Springer pp 605–616

10. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551
11. Lee J, Kim S, Kim S, Park J, Sohn K (2019) Context-aware emotion recognition networks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10143–10152
12. Levi G, Hassner T (2015) Age and gender classification using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 34–42
13. Levi G, Hassner T (2015) Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In: Proceedings of the 2015 ACM on international conference on multimodal interaction, pp 503–510
14. Li B, Lima D (2021) Facial expression recognition via ResNet-50. *Inte J Cogn Comput Eng* 2:57–64
15. Liu K, Zhang M, Pan Z (2016) Facial expression recognition with CNN ensemble. In: 2016 international conference on cyberworlds (CW). IEEE, pp 163–166
16. Liu M, Li S, Shan S, Chen X (2013) Au-aware deep networks for facial expression recognition. In: 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). IEEE, pp 1–6
17. Liu P, Han S, Meng Z, Tong Y (2014) Facial expression recognition via a boosted deep belief network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1805–1812
18. Mairal J, Bach F, Ponce J, Sapiro G (2009) Online dictionary learning for sparse coding. In: Proceedings of the 26th annual international conference on machine learning. pp 689–696
19. Matsugu M, Mori K, Mitari Y, Kaneda Y (2003) Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Netw* 16(5–6):555–559
20. Naqvi K, Hazela B, Mishra S, Asthana P (2021) Employing real-time object detection for visually impaired people. In: Data analytics and management. Springer, pp 285–299
21. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition
22. Saito S, Wei L, Hu L, Nagano K, Li H (2017) Photorealistic facial texture inference using deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5144–5153
23. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
24. Shan C, Gong S, McOwan PW (2009) Facial expression recognition based on local binary patterns: a comprehensive study. *Image Vision Comput* 27(6):803–816
25. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
26. Sun Y, Liang D, Wang X, Tang X (2015) DeepID3: face recognition with very deep neural networks. arXiv preprint [arXiv:1502.00873](https://arxiv.org/abs/1502.00873) (2015)
27. Sun Y, Wang X, Tang X (2015) Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2892–2900
28. Tang Y (2013) Deep learning using linear support vector machines. arXiv preprint [arXiv:1306.0239](https://arxiv.org/abs/1306.0239)
29. Tian YI, Kanade T, Cohn JF (2001) Recognizing action units for facial expression analysis. *IEEE Trans Pattern Anal Mach Intell* 23(2):97–115
30. Tümen V, Söylemez ÖF, Ergen B (2017) Facial emotion recognition on a dataset using convolutional neural network. In: 2017 International artificial intelligence and data processing symposium (IDAP). IEEE, pp 1–5
31. Wang Z, Li Y, Wang S, Ji Q (2013) Capturing global semantic relationships for facial action unit recognition. In: Proceedings of the IEEE international conference on computer vision, pp 3304–3311

32. Zabih R, Woodfill J (1994) Non-parametric local transforms for computing visual correspondence. In: European conference on computer vision. Springer, pp. 151–158
33. Zeng Z, Pantic M, Roisman GI, Huang TS (2008) A survey of affect recognition methods: audio, visual, and spontaneous expressions. *IEEE Trans Pattern Anal Mach Intell* 31(1):39–58
34. Zhang F, Zhang T, Mao Q, Xu C (2018) Joint pose and expression modeling for facial expression recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3359–3368
35. Zhao G, Pietikainen M (2007) Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans Pattern Anal Mach Intell* 29(6):915–928
36. Zhong L, Liu Q, Yang P, Huang J, Metaxas DN (2014) Learning multiscale active facial patches for expression analysis. *IEEE Trans Cybern* 45(8):1499–1510
37. Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database

An Advanced Hybrid Algorithm for Real-World Optimization Problem



Raghav Prasad Parouha

Abstract Between various metaheuristic algorithms, DE (differential evolution) and PSO (particle swarm optimization) is established to be efficient and powerful optimization techniques. Likewise, it has been perceived that their hybrid algorithms afford a consistent estimate to global optimum. Thus, based on multi-swarm tactic an advanced hybrid algorithm (*haDEPSO*) is introduced to solve engineering optimization problems, instead of naïve way. Proposed aDE and aPSO (an advanced DE and PSO) are incorporated in *haDEPSO*. Also, the population of one is combined with the other in a predefined way in *haDEPSO*, to create balance between global and local search capability. Introduced hybrid *haDEPSO* and its participating component aDE and aPSO has been applied over two engineering optimization problems. Results show supremacy of introduced algorithms comparing to so some modern algorithms. Finally, on basis of performance the introduced *haDEPSO* is endorsed in solving engineering optimization problems.

Keywords Hybrid algorithm · Differential evolution · Particle swarm optimization · Engineering optimization

1 Introduction

These days, almost all design optimization problems in engineering are becoming difficult and complicated as a result of involvement of mixed (i.e., continuous and discrete) variables in complex constraints. In general, such situations are multifaceted constrained problems, so they can't be settled using traditional methods effectively. Currently, to control the shortcomings in conventional optimization methods, an important group of optimization methods called meta-heuristics (MAs) have been established. As per the mechanical changes the MAs are classified into 4 types—
(i) SIAs (swarm intelligence algorithms): these are motivated by activities of social insects or animals like PSO [1], ABC (i.e., Artificial Bee Colony Algorithm) [2],

R. P. Parouha (✉)

Indira Gandhi National Tribal University, Amarkantak, M.P. 484886, India

CS (Cuckoo Search) [3], KH (Krill Herd) [4], GWO (Gray Wolf Optimizer) [5], DA (Dragonfly Algorithm) [6], etc. (ii) EAs (evolutionary algorithms): these are motivated after biology like DE [7] and GA (Genetic Algorithm) [8], etc. (iii) PBAs (physics-based algorithms): these are motivated from the rules which are governing a natural phenomenon such as HS (Harmony Search) [9], GSA (Gravitational Search Algorithm) [10], and WCA (namely Water Cycle Algorithm) [11], etc. (iv) HBAs (human behavior-based algorithms): these are inspired from the human being like TLBO (Teaching-learning-based optimization) [12], SAR (Search and rescue optimization) [13], etc.

In various MAs, DE and PSO have been broadly worked in difficult optimization problems. This DE has notable performance so that it is turned out to be an effective optimizer in the research area dealing with real-world problems. Though, some concerns are there with it, like local searchability and convergence rate probability. Nowadays, removing its flaws, many efficient DE has been introduced in the collected works [14–20]. As well, PSO is now famous in solving numerous complex optimization problems because it has effective searchability along with simplicity. Still, it can simply get stuck in a local optimum result area. To get over such concerns various alterations of PSO introduced in the literature [21–27]. Also, the idea of hybrid is very important research directions in increasing the effectiveness of single algorithm. So, in improving the efficiency of PSO and DE, several hybrid techniques have been introduced in the collected works [28–33]. Yet, to overcome their individual flaws, the techniques of hybrid are currently preferred over their separate strength.

After extensive vigorous review of literature on various alternates of PSO and DE with their hybrids, succeeding results are observed and inspired from them. (i) PSO and DE have harmonizing properties so their hybrids very popular currently. As per our knowledge, discovering methods to combine PSO and DE is an open research area today. (ii) mutation and crossover tactic with associate control factors of DE used in order to get the global best solution and that is favorable to improve convergence performance. Hence, best suitable strategies and the parameter values associated in DE, are considered as an important research study. (iii) performance by PSO significantly relies on associated parameters of it. For example, acceleration coefficients and inertia weight which direct particles to balancing diversity and get optimum results. Therefore, scholars have made an effort to adjust control factor of PSO in order to gain optimum results.

Major contribution: Stimulated by above observations and literature inspection, an advanced hybrid algorithm *haDEPSO* with the subsequent suggested component for solving engineering optimization problems. (i) aDE: it consists of novel operators with its linked parameter and (ii.). aPSO: it implicates unique gradually changing (increasing and/or decreasing) factors.

This paper is arranged as: Sect. 2 briefly explains basics of PSO and DE. Projected algorithms are described in Sect. 3. In Sect. 4, application of proposed algorithms is presented. Section 5 concludes this study.

2 Brief on DE and PSO

2.1 Differential Evolution (DE)

After doing initialization, DE is conducted 3 vital operations given below.

Mutation: at the iteration t , for every $x_{i,j}^t$ (target vector) a $v_{i,j}^t$ (mutant vector) is produced as below.

$$v_{i,j}^t = x_{r_1}^t + F(x_{r_2}^t - x_{r_3}^t) \quad (1)$$

where $r_1, r_2, r_3 \in \{1, 2, \dots, np\}$ are arbitrarily chosen integers with $r_1 \neq r_2 \neq r_3 \neq i$ and F signifies the scaling vector.

Crossover: Here, a $u_{i,j}^t$ (trial vector) produced by merging $x_{i,j}^t$ and $v_{i,j}^t$ as below.

$$u_{i,j}^t = \begin{cases} v_{i,j}^t; & \text{if rand} \leq C_r \\ x_{i,j}^t; & \text{otherwise} \end{cases} \quad (2)$$

where rand and C_r (crossover rate) \in random digit between 0 and 1, $i \in [1, np]$, and $j \in [1, D]$.

Selection: it stated as

$$x_{i,j}^{t+1} = \begin{cases} u_{i,j}^t; & \text{if } f(u_{i,j}^t) \leq f(x_{i,j}^t) \\ x_{i,j}^t; & \text{Otherwise} \end{cases} \quad (3)$$

All above operators permitted to offspring repetitively till predetermined stopping condition.

2.2 Particle Swarm Optimization (PSO)

In traditional PSO, a swarm hovers in a D -dimensional search area to find inclusive solution. Every i th swarm particle has its individual position ($x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$) and velocity $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$. While happening of evolution, every particle tracks its individual best pbest and global best gbest, at each iteration, velocity and position of the i th particle are updated as.

$$v_{i,j}^{t+1} = w v_{i,j}^t + c_1 r_1 (\text{pbest}_{i,j} - x_{i,j}^t) + c_2 r_2 (\text{gbest}_j - x_{i,j}^t) \quad (4)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (5)$$

where t : iteration index, $v_{i,j}^t$: velocity of i th particle in D -dimension at the iteration t , here c_1 denotes cognitive acceleration coefficient, c_2 denotes social acceleration coefficient, $r_1, r_2 \in [0, 1]$ and w is the inertia weight.

3 Proposed Methodology

Instead of naïve way an *haDEPSO* is introduced for further improvement in solution quality. In this hybrid *haDEPSO*, whole population arranged according as the value of fitness function and which is divided in 2 sub-populations that is pop_1 (best short) and pop_2 (rest part). Because pop_1 and pop_2 contains top and rest part of the inhabitants which indicates virtuous global and local search competency, respectively. On the separate sub-population (pop_1 and pop_2) introduced aDE: advanced DE (because of its better local searchability) and also aPSO: advanced PSO (due to its good global search capability) correspondingly. Calculating mutually sub-population then better solution is found in pop_1 (called best) and pop_2 (termed gbest) stored individually. If $\text{best} < \text{gbest}$ then pop_2 is fused with pop_1 afterward this fused population evaluated by aDE (as it alleviate the possible stagnation). Else, pop_1 is fused with pop_2 later fused population estimated by aPSO (as it recognized to improved movements). Lastly, recording the optimal solution, if stoppings condition encountered then it stops else returns to sorting progression of population. Carry on this entire process until get need optimal solution. In Fig. 1, flowchart of the *haDEPSO* is revealed.

Mainly, *haDEPSO* works on connecting superior capability of the suggested aDE and aPSO (described as follows).

3.1 ADE: Advanced DE

It consists of modified mutation approach and crossover rate also altered selection scheme are presented which is defined as below.

$$\textbf{Mutation : } v_{i,j}^t = x_{i,j}^t + \tau \times \text{rand}(0, 1) \times (\text{best}_j - x_{i,j}^t) \quad (6)$$

where $v_{i,j}^t$: mutant vector, $x_{i,j}^t$: it is target vector, $\text{rand}(0, 1)$: random number lying in interval $(0,1)$, best_j : best vector and at last τ : convergence factor (it elects the probing balance for every vector). Moreover, dynamic modifications of τ are specified as: (i) if $\tau = 1$: a vector generated in the range $[x_{i,j}^t, \text{best}_j]$. That may increase the convergence rate of Differential Evolution (DE), however note that it may bring the risk of growing likelihood of facing local optima and (ii) if $\tau = \mu(1 - t/t_{\max}) + 1$, where t and t_{\max} are current and total iteration, and μ : positive constant: 1st iteration— $\tau \approx \mu + 1$ (term t/t_{\max} can be ignored as $t = 1$ is much smaller than t_{\max}), in max iteration— $\tau =$

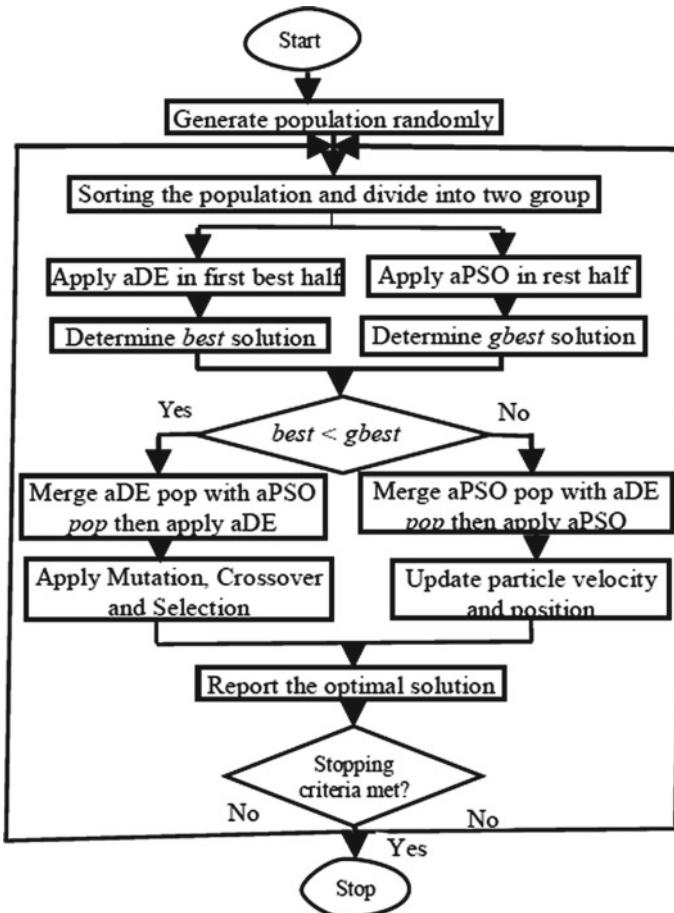


Fig. 1 Flowchart of haDEPSO

1 as $(1 - t/t_{\max}) = 0$). Thus τ linearly starts with $\mu + 1$ to 1 all over the optimization process. That can renovate the convergence and also avert local optima.

So, τ is collected of a chain of large values that enhance the exploring capability for every vector earlier. Later, it makes surety for exploration and exploitation due to collection of a series of small values.

$$\text{Crossover : } u_{i,j}^t = \begin{cases} v_{i,j}^t; & \text{if } \text{rand}(0, 1) \leq C_r (\text{crossover rate}) \\ x_{i,j}^t; & \text{otherwise} \end{cases} \quad (7)$$

where $C_r = e^{\frac{(t-t_{\max})}{t_{\max}}}$ (it guarantees of separate diversity in initial stage and enhances global searchability in later stage).

$$\text{Selection : } x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t; & \text{if } f(u_{i,j}^t) > f(x_{i,j}^t) \text{ and } \text{rand}(0, 1) < p \\ u_{i,j}^t; & \text{otherwise} \end{cases} \quad (8)$$

where $p \in \text{rand}(0, 1]$. In this scheme collectively innovator vector will get chance to continue and also share its observed information with the others. These encourage searching competences and stabilize essential exploration and exploitation of aDE to produce better quality solutions.

3.2 APSO: Advanced PSO

Ideally, PSO requires tough exploration facility (particles can rove complete search space rather than clustering around the current best solution) and boost exploitation capability (particles can explore in a local region) at primary and later period of the evolution correspondingly. Taking every issue such as benefit, drawbacks and parameter impacts of the PSO, an aPSO is presented in this work. It depends over new gradually changing (increasing and/or decreasing) parameters (w , c_1 and c_2) given below.

$$w = w_f + (w_i - w_f) \left(\frac{t}{t_{\max}} \right)^2; c_1 = c_{1f} \left(\frac{c_{1i}}{c_{1f}} \right)^{\left(\frac{t}{t_{\max}} \right)^2} \& c_2 = c_{2i} \left(\frac{c_{2f}}{c_{2i}} \right)^{\left(\frac{t}{t_{\max}} \right)^2}$$

where, w_i and w_f are initial & final values for w ; c_{1i} & c_{1f} : initial & final values of c_1 ; c_{2i} & c_{2f} : initial & final values of c_2 ; t & t_{\max} : iteration index & highest amount of iteration. Therefore the velocity & position for the i th particle are updated as

$$\begin{aligned} v_{i,j}^{t+1} = & \left(w_f + (w_i - w_f) \left(\frac{t}{t_{\max}} \right)^2 \right) v_{i,j}^t + \left(c_{1f} \left(\frac{c_{1i}}{c_{1f}} \right)^{\left(\frac{t}{t_{\max}} \right)^2} \right) r_1 (p_{\text{best } i,j}^t - x_{i,j}^t) \\ & + \left(c_{2i} \left(\frac{c_{2f}}{c_{2i}} \right)^{\left(\frac{t}{t_{\max}} \right)^2} \right) r_2 (g_{\text{best } j}^t - x_{i,j}^t) \end{aligned} \quad (9)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (10)$$

4 Application

To examine performance, proposed aDE, aPSO and *haDEPSO* applied to solve 2 complex engineering optimization problems (EOPs) viz.

- i. Welded beam design (WBD) problem
- ii. Three-bar truss design (TRD) problem.

More details of these problems can be found in [34]. Simulations were done over Intel (R) Core (TM) i7 @ 2.30 GHz, RAM: 6.00 GB, with Windows 10 Operating System, also C-language. By vast investigation, parameters of the proposed methods are recommended as $w_i = 0.4$, $c_{1i} = 0.5$ & $c_{2i} = 2.5$ and $w_f = 0.9$, $c_{1f} = 2.5$ & $c_{2f} = 0.5$ and bracket operator penalty [35] ($R = 1e^{03}$) to handle constraint is chosen for this study because it has higher competence. Whole best values in every table are noted with bold letters of the matching methods. To have reasonable judgment the np (population size) = 100, independent run (30), stopping criteria (240,000 function evaluations maximum) are considered same as comparative algorithms in all cases. The outcomes of proposed methods on 2 EOPs are equated with PSO [1], ABC [2], CS [3], KH [4], GWO [5], DA [6], GA [8], EO [36], CSDE [37], SCA [38], EPO [39], SHO [40] and GSA-GA [41].

The experiential outcomes of proposed with other methods on particular EOPs are presented in Table 1 (for WBD) and Table 2 (for TRD). From these tables, it is very clear that the proposed methods (aDE, aPSO and *haDEPSO*) give better and/or equally results on all EOPs. Finally, these aDE, aPSO and *haDEPSO* produces less standard for all cases which describes their stability.

The schematic diagram and convergence graphs of proposed and best non-proposed methods are presented in Figs. 2 and 3 on EOPs. These figures clearly depict that proposed methods converges faster than the others. So, projected methods are computationally effective.

Generally, it can be said that performance of the proposed methods (aDE, aPSO and *haDEPSO*) are better and/or equally in comparison with others. Though, in proposed 3 methods, the *haDEPSO* has superior capability.

5 Conclusion

This work proposes *haDEPSO* for engineering optimization problems, instead of naïve way, where integration of an aDE (advanced DE) and aPSO (advanced PSO) is done in proposed hybrid. As a result—(i) *haDEPSO* is developed on uniting aDE and aPSO. This is motivated from multi-swarm tactic in which one population is fused with the other according to some predefined way and it leads to surety of convergence and diversifying solutions, (ii) in improving performance and alter the control factors of DE, aDE proposed. The novel mutation approach, crossover possibility and changed selection systems of aDE will assure exploration and exploitation at starting

Table 1 Simulation results for WBD problem

Methods	Best values for variables				Best	Worst	Mean	Std.
	$x_1(\text{H})$	$x_2(\text{L})$	$x_3(\text{T})$	$x_4(\text{b})$				
PSO	0.197411	3.315061	10.00000	0.201395	1.820395	3.048231	2.230310	0.324525
ABC	0.205730	3.470489	9.036624	0.205730	1.724852	1.734852	1.741913	0.031
CS	0.2015	3.562	9.0414	0.2057	1.7312065	1.8786560	2.3455793	0.2677989
GWO	0.205678	3.475403	9.036964	0.206229	1.726995	1.727128	1.727564	0.0011157
DA	0.194288	3.46681	9.04543	0.205695	1.70808	1.94076	2.52106	0.250234
GA	0.164171	4.032541	10.00000	0.223647	1.873971	2.320125	2.119240	0.034820
EO	0.2057	3.4705	9.036664	0.2057	1.724853	1.736725	1.726482	0.003257
SCA	0.2444	6.2380	8.2886	0.2446	2.3854	6.3996	3.2551	0.9590
EPO	0.205411	3.472341	9.035215	0.201153	1.723589	1.727211	1.725124	0.004325
SHO	0.205563	3.474846	9.035799	0.205811	1.725661	1.726064	1.725828	0.000287
aDE	0.184288	3.26641	8.24133	0.204585	1.72355	1.72874	1.72625	0.000284
aPSO	0.184288	3.26641	8.24133	0.204585	1.72486	1.72698	1.73542	0.000354
haDEPSO	0.184288	3.26641	8.24133	0.204585	1.69782	1.72321	1.72421	0.000132

Table 2 Simulation results for TRD problem

Methods	Best values for variables		Worst	Mean	Std.
	$x_1(A_1)$	$x_2(A_2)$			
GSA-GA	0.788676171219	0.408245358456	263.8958433	263.8958459	263.8958437
CSDE	0.7886	0.4082	263.148352124271	65.535	263.148352318831
PSO	0.7803	0.4330	264.543754826635	2.70864524844245e+05	264.775374387455
DE	0.7887	0.4080	263.148352624688	65.535	263.41127728312
KH	0.7885	0.4088	263.9	265.0	263.9
CS	0.7357	0.5.945	263.602007628033	5.23525611223402e+12	263.671445662651
aDE	0.788526	0.408452	262.6263	263.7845	263.7845
aPSO	0.788526	0.408452	263.4263	263.8497	263.8497
<i>haDEPSO</i>	0.788526	0.408452	261.1438	262.9796	2.5142e-09

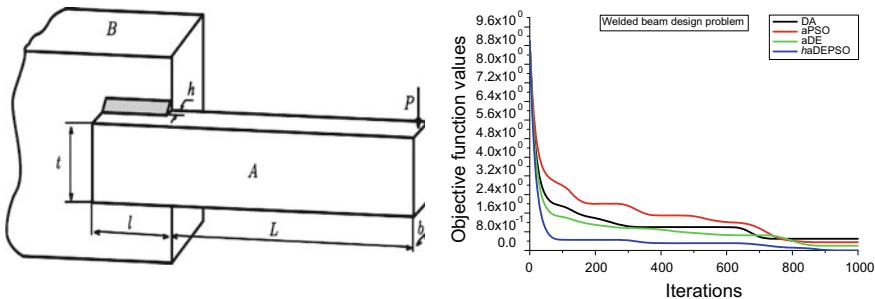


Fig. 2 Schematic diagram and convergence of WBD problem

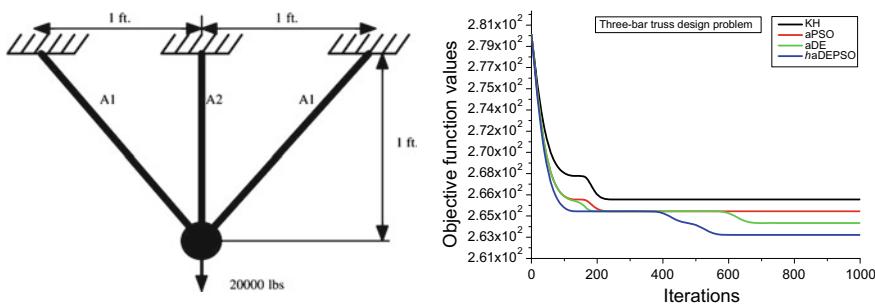


Fig. 3 Schematic diagram and convergence of TRD problem

and ending respectively and (iii) to escape particles stagnant, an aPSO is proposed which consisting of new progressively changing (increasing and/or decreasing) parameters. These new parameters, can be encouraged particles in searching better quality solution of aPSO algorithm.

The efficiency proposed methods (*haDEPSO*, *aDE* and *aPSO*) evaluated on 2 complex engineering optimization problems. It is analyzed numerically, statistically and graphically the proposed algorithms in comparing with modern algorithms. Results reveal that introduced procedures are more robust with higher effectiveness. Also, feasibilities, superiorities and solution optimality among proposed (*haDEPSO*, *aDE* and *aPSO*) and comparative methods *haDEPSO* has outperformed.

References

1. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceeding of IEEE international conference on neural networks, pp 1942–1948
2. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm. *J Glob Optim* 39(3):459–471

3. Yang XS, Deb S (2009) Cuckoo Search via Lévy flights. In: Proceedings of world congress on nature & biologically inspired computing, Coimbatore, India, pp 210–214
4. Gandomi H, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17(12):4831–4845
5. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
6. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems. *Neural Comput Appl* 27(4):1053–1073
7. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
8. Davis L (1991) Handbook of genetic algorithms
9. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *SIMULATION* 76(2):60–68
10. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) A gravitational search algorithm. *Inf Sci* 179(13):2232–2248
11. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110–111:151–166
12. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43(3):303–315
13. Shabani, Asgarian B, Gharebaghi SA, Salido MA, Giret A (2019) A new optimization algorithm based on search and rescue operations. *Math Prob Eng* 2019:1–23
14. Yang X, Li J, Peng X (2019) An improved differential evolution algorithm for learning high-fidelity quantum controls. *Sci Bull* 64(19):1402–1408
15. Prabha S, Yadav R (2019) Differential evolution with biological-based mutation operator. *Eng Sci Technol Int J*, pp 1–11
16. Liu Z-G, Ji X-H, Yang Y (2019) Hierarchical differential evolution algorithm combined with multi-cross operation. *Expert Syst Appl* 130:276–292
17. Gui L, Xia X, Yu F, Wu H, Wu R, Wei B, He G (2019) A multi-role based differential evolution. *Swarm Evol Comput* 50:1–15
18. Li S, Gu Q, Gong W, Ning B (2020) An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models. *Energy Convers Manage* 205:1–16
19. Hu L, Hua W, Lei W, Xiantian Z (2020) A modified Boltzmann Annealing differential evolution algorithm for inversion of directional resistivity logging-while-drilling measurements. *J Petrol Sci Eng* 180:1–10
20. Ben GN (2020) An accelerated differential evolution algorithm with new operators for multi-damage detection in plate-like structures. *Appl Math Model* 80:366–383
21. Parouha RP (2019) Nonconvex/nonsmooth economic load dispatch using modified time-varying particle swarm optimization. *Comput Intell*, pp 1–28. <https://doi.org/10.1111/coin.12210>
22. Hosseini SA, Hajipour A, Tavakoli H (2019) Design and optimization of a CMOS power amplifier using innovative fractional-order particle swarm optimization. *Appl Soft Comput* 85:1–10
23. Kohler M, Vellasco MMBR, Tanscheit R (2019) PSO+: a new particle swarm optimization algorithm for constrained problems. *Appl Soft Comput* 85:1–26
24. Khajeh MR, Ghasemi HG (2019) Arab, Modified particle swarm optimization with novel population initialization. *J Inf Optim Sci* 40(6):1167–1179
25. Ang KM, Lim WH, Isa NAM, Tiang SS, Wong CH (2020) A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems. *Expert Syst Appl* 140:1–23
26. Lanlan K, Ruey SC, Wenliang C, Yeh C (2020) Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications. *Appl Soft Comput* 88:1–10

27. Xiong H, Qiu B, Liu J (2020) An improved multi-swarm particle swarm optimizer for optimizing the electric field distribution of multichannel transcranial magnetic stimulation. *Artif Intell Med* 104:1–14
28. Parouha RP, Das KN (2016) DPD: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints. *Expert Syst Appl* 63:295–309
29. Mao B, Xie Z, Wang Y, Handroos H, Wu H (2018) A hybrid strategy of differential evolution and modified particle swarm optimization for numerical solution of a parallel manipulator. *Math Prob Eng*, pp 1–9
30. Tang B, Xiang K, Pang M (2018) An integrated particle swarm optimization approach hybridizing a new self-adaptive particle swarm optimization with a modified differential evolution, *Neural Comput Appl*, pp 1–35
31. Too J, Abdullah AR, Saad NM (2019) Hybrid binary particle swarm optimization differential evolution-based feature selection for EMG signals classification. *Axioms* 8(3):1–17
32. Dash J, Dam B, Swain R (2020) Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization, *AEU - Int J Electron Commun* 114:153019
33. Zhao X, Zhang Z, Xie Y, Meng J (2020) Economic-environmental dispatch of microgrid based on improved quantum particle swarm optimization. *Energy* 195:117014
34. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, *Appl Soft Comput* 10:629–640
35. Deb K (1995) Optimization for engineering design: algorithms and examples, Prentice-Hall of India, New Delhi
36. Faramarzi A, Heidarnejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: A novel optimization algorithm. *Knowl Based Syst* 191:105190
37. Zhang Z, Ding S, Jia W (2019) A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. *Eng Appl Artif Intell* 85:254–268
38. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
39. Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl Based Syst* 159:20–50
40. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70
41. Garg H (2019) A hybrid GSA-GA algorithm for constrained optimization problems. *Inf Sci* 478:499–523

An Online Document Emoji-Based Classification Using Twitter Dataset



Shelley Gupta, Archana Singh, and Jayanthi Ranjan

Abstract An enormous amount of unstructured data is generated by online media like online connecting sites, microblogs, ecommerce sites, etc. Nowadays, online data plays a very decisive part in analyzing and predicting the opinions and sentiments of people toward an organization, brand, famous personality, etc. which in turn tremendously helps company to improve product quality, model market strategy, measure return of investment, determine people's attitude toward famous personalities, artists, etc. The major component of these online documents is text and emoji which has motivated us to provide an approach to classify the online document into “Online Document Emoji Class, ODEC”, i.e., classifying the various online document on the basis of number of emojis in the sentiment. In this paper, we have also examined the tweets of numerous world-famous personalities as online document based on classification of popular regions. The results show that the number of emoji used for most of the cases is approx. 60% of the total number of tweets data. Thus, this quantification of online data will help us to identify the relevance of emoji study in sentiment analysis and better efficient way of classification of data.

Keywords Sentiment analysis · Online document classification · Online Document Emoji Class · Twitter data analysis · Tweets · Emoji · Emoji count

S. Gupta (✉)

Department of Computer Science and Engineering, Amity School of Engineering and Technology, Noida, India

e-mail: shelley.gupta@abes.ac.in

A. Singh

Department of Information Technology, Amity School of Engineering and Technology, Noida, India

e-mail: asingh27@amity.edu

J. Ranjan

GITAM University, Hyderabad, Telangana, India

1 Introduction

Sentiment analysis has gained huge interest of researchers as the online buyers express their likings, reviews, comments, etc. on social media platform. The analysis of such expressions to determine its polarity is known as sentiment analysis. This analysis helps an organization to improve their product quality, define market policies, improve customer services, etc. Nowadays, an online user's post is accomplished with text and emojis both to express their opinions [1–3].

The text in posts includes words, special characters, numerals, etc. The emojis are an encoded character sequence that presents an emotion of user utilized as an inline in post sentences. Using emojis, users' online expressions have received a completely modern mode of communication in the form of pictures depicting a mode, sport, accessories, festival, event, etc. [4]. This has replaced the use of long sentences with pictorial expressions in online expressions [5].

In our research work, we have analyzed tweets of various personalities as an individual online document to determine the number of emojis mostly used by online users. To reinforce our work, we have used 36,600 tweets of most followed personalities across the world from prominent regions like America, Europe and Asia [3, 6].

Thus, the prime aim of our research work is to: (1) To categorize the online document into Online Document Emoji Class, ODEC, which classifies an online document sentiment into a class based on the number of emojis in the document, i.e., 1–2 emojis, 3–5 emojis, 6–9 emojis, 10–10+ emojis. (2) To evaluate the total count and percentage of post in an online document belonging to different classes of ODEC. Thus, the main aim is to reflect the great relevance and need of sentiment analysis approaches accounting online documents with text and emoji both for factual sentiment analysis.

2 Literature Review

Sentiment analysis aims at identifying the polarity of a given text, i.e., determining whether the sentiment expressed in the text is positive, negative or neutral [1, 2]. The sentiment analysis can be done at sentence level, document level or aspect level based on machine learning, rule-based, deep learning and hybrid approaches [2]. The online sentiments consist of emojis and text both [2, 3]. The approaches which have done sentiment analysis using text are numerous in number [2, 7]. The rule-based approaches that has performed sentiment analysis using text are General Inquire [8], WordNet [9], ANEW [10], LIWC [11], SentiwordNet [1, 12], SenticNet [13], Senti-n-gram [2], etc. The research work [14] involves sentiment analysis of text using deep learning approaches. Similarly, the research work [15] involves doing sentiment analysis using hybrid approach, which is a combination of machine learning and rule-based approaches.

In the beginning, the emoticons came. An emoticon is a typographic representation of facial expression using punctuation marks, number, letters, etc. Emoticon literal meaning is “emotion icon”. The Scott Falman proposed the first emoticons as:-) and:-(. In the late 1990s, emoticons started playing important role in messages and emails [5].

Emojis are actual pictures, representing various class and style of facial expression, art, music, places, weather and objects. The literal meaning of emoji is “image” and “character” [5, 16]. Emoji originated on Japanese mobile phones in 1997 and incorporated too many operating system of mobile in late 2010. Now, Unicode Standard released hundreds of emoji characters with faces representing different moods. The latest Unicode version 13.0 was released on 2020 providing sequences for gender and skin tones [4].

The approaches that perform sentiment analysis involving text and emoji both are less in number [3]. The emoji sentiment ranking counted the existence of each emoji, the tweet labels accommodating it determine its sentiment, and its discrete probability distribution was evaluated. They allocated scores of +2 and -2 to strong positive and strong negative emojis, respectively, and +1 and -1 to weak positive and weak negative emojis, respectively [17].

The research work [18] collected 134,194 Arabic tweets and investigates which emojis were most usually used in them. The most frequently used ones were classified into four categories: Anger, disgust, joy, and sadness with a weight between -5 and +5 were assigned to each emoji, where the signs – and + represent negativity or positivity of the sentiment, depending on the category.

The prime objective of the research paper is to analyze the Twitter dataset to determine the number of emojis used by an online user in their online document and to classify this document [19]. The research work has evaluated the number of emojis used with respect to the aggregate number of tweets considered for different popular personalities of prominent regions like America, Europe and Asia.

3 Proposed Approach

Figure 1 depicts our proposed approach is shown in consisting of: (1) Data collection approach (2) ODEC Classification approach.

3.1 Data Collection Approach

Step 1. Acquire API Keys and Access Keys from Twitter

For accessing Twitter streaming API, we require 4 keys: Twitter API Key, API secret, Access Token and Access Token Secret. These four keys are obtained by the below steps:

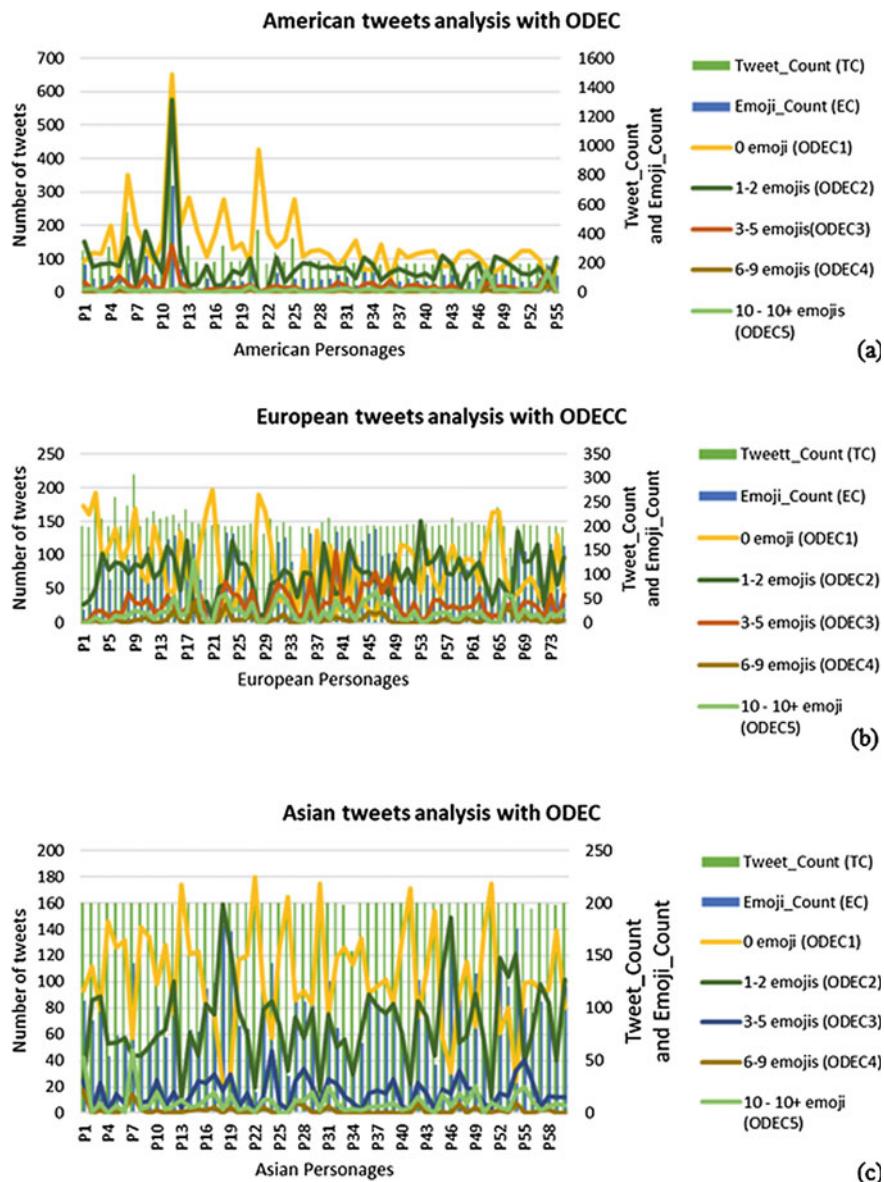


Fig. 1 Tweet's analysis of America, Europe and Asia regions using Online Document Emoji Class (ODEC)

Creating or using an existing Twitter account.

Move to <https://apps.twitter.com/> along with login the account, make new app and create your Twitter Application.

Get Twitter API secret by API keys tab.

Get Access Token and Access Token Secret by clicking on “Create my access token” button.

Step 2. Twitter handle classification

- Identifies most followed popular personalities on Twitter [3, 6].
- Collection of Twitter handle of selected personalities.
- Classification of Twitter handle on the basis of three regions like America, Europe and Asia.

Step 3. Fetching Twitter data using Twitter streaming API

The tweets of the personalities for above Twitter handles are fetched using Twitter Streaming API keys and Python library called *Tweepy*. The credentials like API Key, API secret, Access Token, and Access Token Secret are embedded into the python code using commands like *tweepy.OAuthHandler (API Key, API secret)*, *set_access_token (accessToken, accessTokenSecret)*. Approximately 36,600 tweets have been downloaded.

Step 4. Preprocessing of dataset

The dataset is preprocessed to remove the pictures, stop words, numeric and special characters.

3.2 ODEC Classification Approach

Step 1. Online document classification

Online Document Emoji Class is a class set which will classify the online document as: ODEC = {0 emoji, 1–2 emojis', 3–5 emojis', 6–9 emojis', 10–10+ emojis'}, shown in Table 1.

- Tweet_Count (TC): The function to calculate aggregate number of tweets in the online document, D considered.
- Emoji_Count (EC): The function to aggregate number of emoji in the online document, D considered.

Step 2. Graphs are drawn

The combo graphs are drawn with plotting of ODEC classes on vertical primary axis using clustered graph the personages and number of tweets are taken on vertical and horizontal primary axis respectively. The values of aggregate number of tweets

Table 1 Online Document Emoji Class, ODEC

Online Document Emoji Class, ODEC		
ODEC class no	ODEC Class	Remarks
ODEC1	0 emoji	The online sentiment containing 0 emoji will be classed in '0 emoji'
ODEC2	1–2 emojis	The online sentiment containing 1–2 emoji in totality will be classed in '1–2 emojis'
ODEC3	3–5 emojis	The online sentiment containing 3–5 emoji in totality will be classed in '3–5 emojis'
ODEC4	6–9 emojis	The online sentiment containing 6–9 emoji in totality will be classed in '6–9 emojis'
ODEC5	10–10+ emojis	The online sentiment containing 10–10+ emoji in totality will be classed in 10–10+ emojis

in each ODEC class are plotted on vertical primary axis. The tweet_count and emoji_count are plotted on the secondary vertical axis using line graphs.

Step 3. Conclusions drawn

The inference is drawn based on analysis of around 183 popular personalities across the world.

4 Experiments and Results

The Twitter API (Application Programming Interface) has been used to download tweets, using 4 API keys: Twitter API Key, API secret, Access Token, and Access Token Secret. The detailed process of downloading the dataset using Twitter API is given in Sect. 3.1. This user generated dataset consists of approximately 36,600 tweets posted by around 183 top most followed personalities on Twitter [3, 6]. For each personality, we have taken near about 200 tweets.

Figure 1c presents the ODEC analysis for the three popular regions. Figure 1a–c represents the ODEC classification of popular personalities of America, Europe and Asia respectively.

Figure 1 shows that the aggregate number of emojis used as compared to total number of tweets is significant in number for each of the three regions. Table 2 provides the comparative data insight for total count of emoji usage. The results include:

- The European personages have used maximum of 1–2 emoji in their online sentiments as compared to other regions.
- The three regions have used mostly 1–2 emojis' in their tweets as compared to 3–5 emojis', 6–9 emojis', etc.
- Next to 1–2 emoji, PR classes have used 3–5 emojis' most in their tweets. Further 6–9 emojis' and 10–10+ emojis' most.

Table 2 Tweet analysis with ODEC classification

Popular region (PR)	ODEC, tweet and emoji count						Emojii_count
	TC	ODEC1 (ODEC1/TC) * 100	ODEC2 (ODEC2/TC) * 100	ODEC3 (ODEC3/TC) * 100	ODEC4 (ODEC4/TC) * 100	ODEC5 (ODEC5/TC) * 100	
America	60,640	38,807	64.00	15,159	25.00	3,979	6.56
Europe	60,478	27,456	45.40	19,717	32.60	7,401	12.24
Asia	48,430	26,431	54.38	14,105	29.12	3,720	7.68
						1,272	2.63
						2,756	5.69068759
						21,853	45.12

5 Discussion

ODEC classification for regions like America, Europe and Asia clearly shows that:

- Usage of 1–2 emoji is used maximum by personages across the world.
- Also 10–10+ emoji usage among the personages is more as compared to the usage of 6–9 emoji within the tweets considered.
- The number of emoji used for most of the cases is approx. 60% of the total number of tweets data.

Thus, the overall usage of emojis' within the user generated online documents is much significant in number. The sentiment analysis is complete and accurate when both the content of online documents, i.e., text and emoji are considered for sentiment determination. Most of the work in sentiment analysis done till now as discussed in Sect. 2 involves considering text or emoji as content not both.

In this paper, it raised the concern of significant contribution of emojis' and text together to perform sentiment analysis.

6 Conclusion and Future Scope

This paper attempts to classify user generated tweets followed by popular personalities across the worlds on Twitter social media platform. The online document, i.e., tweets, has been classified on the basis of number of emojis' used, *ODEC classification*. These classifications have been done using prominent regions of America, Europe and Asia. The paper performs unsupervised classification technique. This work signifies that the accuracy of score and polarity of sentiment expression on social media is incomplete without considering the emojis.

The present work gives the future direction to perform perceptive and cognitive models using deep learning techniques and the online document class explored in this paper. Attempts can be made to similar analysis using other online document content like picture, GIF, videos, etc.

References

1. Asghar MZ, Khan A, Bibi A, Kundi FM, Ahmad H (2017) Sentence-level emotion detection framework using rule-based classification. *Cogn Comput* 9(6):868–894
2. Dey A, Jenamani M, Thakkar JJ (2018) Senti-N-gram: an n-gram lexicon for sentiment analysis. *Expert Syst Appl* 103:92–105
3. Gupta S, Singh A, Ranjan J (2020) sentiment analysis: usage of text and emoji for expressing sentiments. In: Advances in data and information sciences. Springer, Singapore, pp 477–486
4. Full Emoji List, v13.1. (2021) Retrieved 1 May 2021, from <https://unicode.org/emoji/charts/full-emoji-list.html>

5. Fernández-Gavilanes M, Juncal-Martínez J, García-Méndez S, Costa-Montenegro E, González-Castaño FJ (2018) Creating emoji lexica from unsupervised sentiment analysis of their descriptions. *Expert Syst Appl* 103:74–91
6. Find out who's not following you back on Twitter, Tumblr, & Pinterest, <https://friendorfollow.com/twitter/most-followers/>
7. Dashtipour K, Poria S, Hussain A, Cambria E, Hawalah AY, Gelbukh A, Zhou Q (2016) Multilingual sentiment analysis: state of the art and independent comparison of techniques. *Cogn Comput* 8(4):757–771
8. Stone PJ, Dunphy DC, Smith MS (1966) The general inquirer: a computer approach to content analysis
9. Fellbaum C (1998) WordNet. Wiley Online Library
10. Bradley MM, Lang PJ (1999) Affective norms for English words (ANEW): instruction manual and affective ratings. Technical Report C-1. The Center for Research in Psychophysiology, University of Florida
11. Pennebaker JW, Francis ME, Booth RJ (2001) Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates, 71, 2001
12. Baccianella S, Esuli A, Sebastiani F (2010) Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the conference on language resources and evaluation, LREC: 10, pp 2200–2204
13. Cambria E, Havasi C & Hussain A (2012). Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In: Proceedings of the Florida artificial intelligence research society conference, Flairs (pp. 202–207).
14. Sun X, Zhang C, Ding S, Quan C (2018) Detecting anomalous emotion through big data from social networks based on a deep learning method. *Multim Tools Appl*, pp 1–22
15. Zobeidi S, Naderan M, Alavi SE (2019) Opinion mining in Persian language using a hybrid feature extraction approach based on convolutional neural network. *Multim Tools Appl* 78(22):32357–32378
16. Gupta S, Singh A, Ranjan J (2021) Emoji score and polarity evaluation using CLDR short name and expression sentiment. In: Abraham A et al (eds) Proceedings of the 12th international conference on soft computing and pattern recognition (SoCPaR 2020). SoCPaR 2020. Advances in intelligent systems and computing, vol 1383. Springer, Cham. https://doi.org/10.1007/978-3-030-73689-7_95
17. Novak PK, Smailović J, Sluban B, Mozetič I (2015) Sentiment of emojis'. *PloS One* 10(12):e0144296
18. Hussien WA, Tashtoush YM, Al-Ayyoub M, Al-Kabi MN (2016) Are emoticons good enough to train emotion classifiers of Arabic tweets? In: 2016 7th international conference on computer science and information technology (csit). IEEE, pp 1–6
19. Alzubi OA, Alzubi JA, Tedmori S, Rashaideh H, Almomani O (2018) Consensus-based combining method for classifier ensembles. *Int Arab J Inf Technol* 15(1):76–86

Approaches to Optimize Memory Footprint for Elephant Flows



Vivek Kumar , Dilip K. Sharma , and Vinay K. Mishra

Abstract The technology revisions are too quick to resist or stay apart. Most of the work, on massive and streaming data, tries to scale up the computation. The scale up and scale out comes with its own set of advantages and disadvantages. The computational hardware has already been used to its limits with best of the algorithms. Our work proposes and implements two approaches: (a) memory bound and (b) ingestion bound that optimizes memory footprint and speeds up the computation. The work compares the stated techniques on two datasets: (a) moma and (b) yelp and found up to 80% conditional optimization. The work shows results of optimization for big data stream and proves that the approach is worth implementing as compared to other state of the art techniques for big data stream aka Elephant Flows.

Keywords Big data · Optimization · Stream processing · In-memory · Elephant flows · Yelp dataset

1 Introduction

A continuous stream of data is known as data stream. The sources of data streams are network traffic, sensor network, search logs, scientific data, utility monitoring, financial applications, etc. In the streaming model, there is a sequence of elements presented to the algorithm. The algorithm is allowed a single pass over the stream. The algorithm computes a function or relation of the data stream. The algorithm is approximate in order to be efficient. The functions computed on data streams may be order-dependent or order-independent [7]. One of the main goals of streaming

V. Kumar ()

Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India

D. K. Sharma

GLA University, Mathura, U.P., India

e-mail: dilip.sharma@glau.ac.in

V. K. Mishra

S.R.M. Group of Professional Colleges, Lucknow, U.P., India

algorithm is to use as little memory as possible. Note that available memory is a hard limitation on what's possible to process. A memory limitation is when a dataset won't fit into the available memory of system, i.e., a 6 gigabyte dataset to fit in 4 gigabytes of available memory. There is no way to load data at once and process it without using a workaround. In this scenario, one needs to rely on workarounds like processing the data in batches that do fit into memory. This requires using strategies to handle big and streaming data [9]. A program-bound mostly limits how quickly the program may be executed. Program-bound is also a limitation which affects the way of processing data; however, the program bound isn't a hard limitation. There are two primary ways a program can be bound:

- CPU-bound: CPU-bound program is dependent on the CPU to execute quickly. The cores can be used intelligently to run the program faster.
- I/O-bound: I/O-bound program is dependent on external resources, like files on disk or network services, to execute quickly. The faster the accessibility of external resources, the faster the program.

Most work on massive data tries to scale up the computation. One approach is to optimize the data, within permissible time limits, as encountered. Working with larger datasets or data streams, understanding these bounds and making the program more efficient to deal with them, is critical and important. The in-memory computing is the fastest way to compute and override the disk seek-time. The memory access time (measured in nanoseconds) is one million times faster than disk-access time (measured in milliseconds).

The big data challenges are ingestion, processing, retention, communication and various others. The similar challenges are for big data streams, out of which high rate of ingestion and high processing requirements are major challenges. The ingestion is done with the help of some tool or programming language using library/module/package. Each tool, language or even library has its own advantages and disadvantages. Legacy tools provide stability but they are not best suited for handling and processing big data streams. In case of streaming data, the frame may demand larger memory, e.g., the data stream from the Large-Hadron-Collider Computing Grid (LCG), now Worldwide LHC Computing Grid (WLCG). The four main detectors, namely alice, atlas, cms and lhcb at the LHC collectively produced 40 GB/s [3]. It generates 140 terabytes of data per day and 25 petabytes per year [1]. There are various tools readily available, out of which Spark handles big datasets (100 GBs to multiple TBs) but it requires expensive and scalable hardware. These tools even lack rich features, like pandas, which have features like high quality data cleaning, exploration and analysis.

1.1 *Background and Motivation*

The python library, Pandas, is a modular tool used for ingestion and processing data as data-frames. We observed that the data-frames, in pandas, use inbuilt default

data types for ingestion which takes large memory. We addressed this downside of pandas and proposed the techniques to handle and optimize big data streams using pandas objects and methods to address the mentioned. We implemented the proposed techniques on python, a programming language. We considered one medium-sized publicly available dataset “museum.csv” [4] as a stream window, assuming there are other windows whose sum does not fit in memory at once. We also considered one large-sized publicly available dataset “yelp_review.csv” [5] for experimentation. We focused on techniques to ingest big data stream when memory is not enough. The datasets “museum.csv” is 10.2 MB on disk and “yelp_review.csv” is 3.53 GB on disk. These files when read as data-frames in pandas, the size returned was 7.4 MB and 361.3 MB, respectively, which looks good but this is only pointer storage size which is 8 bytes per pointer. The original memory footprint pandas returned was 46.8 MB and 4.9 GB, respectively, which includes the sum of variable sized objects along with sum of pointer size. This means that about 4656 ($4.9 * 1024 - 361$) MB is required to store the actual Python strings for the object type data-columns for yelp dataset. This led to a direction to overcome the downside of data-frames using different techniques to optimize memory footprint.

1.2 Overview of the Paper and Contributions

The optimization techniques devised and applied to minimize the data-load on to the memory. A big data is considered for preprocessing optimization. The overall optimization achieved on different dataset fount to be 80% on comparison.

1.3 Arrangement of Sections in Article

This article is arranged in five sections. First section discusses the background and prerequisites. Section 2 derives the conclusions made from literature survey and related work. Section 3 proposes the architecture and algorithm. Section 4 shows the experimentation and results. Section 5 derives conclusions based on experiments performed and put some light on future directions.

2 Literature Survey and Research Gap

The TCP flow consists of packets which are multiple, consecutive, numbered and time-stamped. In case of out of order transmission, the retransmission is done. Provided that the memory space should be smaller than the number of distinct TCP flows; the problem is to estimate the number of flows that have out of order packets at any time [13]. Volume and velocity challenges are related with scalability issue.

The authors categorized scalability into vertical and horizontal scalability. Scalable computation using scale up (for vertical scalability) and scale out (for horizontal scalability) does not resolve problem complexity [2]. An architecture proposes [17] combining stream processing with complex event processing. The architecture also analyzed the information in real time for water management. Predictive maintenance is the requirement for industry, like railway and wind turbines; Industry 4.0 fulfills this with the help of big data and IoT and stream processing [15]. Batch and stream processing have many applications but in separation. A hybrid approach is required for applications in cloud computing and internet of things. Hybrid Distributed Batch-Stream (HDBS) architecture [14] is proposed for detecting anomaly in real-time data. Architecture, “FineStream,” proposed for stream processing and claims to achieve $10.4 \times$ price-throughput ratio. “FineStream” is a window-based and CPU-GPU Integrated Architecture [20]. Data stream processing systems (DSPSs) achieves real-time analytics. Recent researches are focusing on optimizing the system latency and throughput. A paper [21] surveys such researches in the direction of optimization in computation, query deployment and stream I/O. A recent work compares distributed stream management systems (DSMSs) with respect to their strengths and limitations. The work concludes that “Flink” is good for complex stream processing, “Storm” provides better latency with fewer restrictions and “Spark” has a bigger community [16]. A work [19] proposes fast incremental model tree with the drift-detection (FIMT-DD) algorithm that address the challenge of mining real-time information of big time-series data. Another work [10] focuses on knowledge extraction through stream processing, from large scale wireless networks. The work addresses challenges and research projects in mentioned area. Mille Cheval [8] is a framework for accelerated processing of Elephant flows which increases throughput using sketch data structure and GPU. A survey [18] addresses curse of dimensionality and presents methods for reduction of big data [12]. A survey, [11] focus on the challenges of real-time DSPS and recent developments. It focuses on real-time data warehouse (DWH) and challenges of Extract, Transform and Load (ETL) process. It answers real-time approaches and tools at ETL process for DWH. Another survey [6] addresses challenges raised by number of applications generating big data at a big velocity. This big data overwhelms the existing data mining techniques, tools, methods and technologies. Paper considers three major databases: Scopus, ScienceDirect and EBSCO. The authors conclude that “not much attention has been given to the preprocessing stage of big data streams.” This statement motivated us to work on preprocessing and optimization of elephant flows, hence, our work.

3 Proposed Work

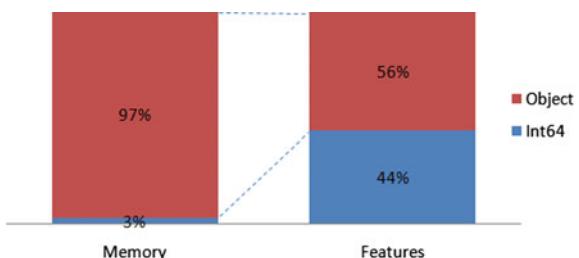
Classical data stream algorithms are time series, cash register, and turnstile. Stream items x_1, x_2, \dots, x_n describing a signal, S , is one dimensional vector and can be thought of as a function over the real, that is, $x_i = S[i]$. Despite various restrictions of classical data streaming, data sketching and statistics problems have achieved $O(1)$

passes and poly-logarithmic working space to find approximate solution. The goal is to process the input stream using a small amount of space, s , in memory. Since the stream length, m , and the universe size, n are to be thought of as big and infinite, respectively, s is supposed to be much smaller than m and n . The algorithm passes over the stream a single time or a small number of times, depending on the velocity of data and the goal processing time. The algorithm is necessarily randomized and approximated in order to be efficient. We will compute order-independent functions $f(a_1, a_2, \dots, a_m)$. A function is order independent if applying any permutations to its inputs results in the same function value.

Stream processing on high velocity is demanded in various fields of scientific applications like astrophysics, genomics, physical simulations, Large Hadron Collider (LHC), and various other fields like Internet of Things (IoT), Database Query Optimization, Satellite's data real-time processing, airplane's black box data real-time processing, social media's contents like posts, tweets, photos, manufacturing operational monitoring like dashboard, relay controls, dynamic routing algorithms, and network traffic analysis. The stream processing is in use for monitoring and analyzing operations but in the era of exabytes, the existing techniques and models are getting exhausted due to high velocity of data. The various techniques and models for stream processing are discussed below with their advantages and disadvantages. Any data that overwhelms the computing resources and compute capability is termed as big data. We considered yelp data as big data because its memory requirement for loading the data is 5 GB, whereas the system on which it is going to process have 4 GB free memory.

The true memory footprint of the data-frame is 4.90 GigaByte (GB). This means that about 4.55 GB is required to store the actual Python strings for the object data-columns. Pandas use 4892 MegaByte (MB) out of 5052 MB to represent the object data-columns. Figure 1 depicts that 44% Integer data-columns are taking just 3% of total consumed memory. This means that the large memory savings may be achieved by converting object data-columns to numeric ones.

Fig. 1 Feature memory ratio by data types for yelp



3.1 Memory Bound Approach

Memory is saved by converting within the same type (from float64 to float32 or float16 for example), or by converting between types (from float64 to int32/int16/int8). Figure 2a depicts that 78% optimization achieved by converting to best possible integer or float to accommodate the value. Actual was int64 which is optimized to either int8 or int16. The type is chosen according to the largest data in the column.

The date column, containing dates, is also considered for optimization. Figure 2b depicts that 88% optimization achieved by converting this object to datetime type. Pandas v0.15 introduced Categoricals. The category type represents the values in a column using integer values under the hood. Pandas use a separate mapping dictionary for a column containing a limited set of values. If less than 50% of the values are unique for object features, converting to the categorical type would save a lot of space. Figure 2c depicts that 90% optimization achieved. The category subtype imputes -1 for missing values.

Table 1 shows the comparative study of both datasets with their optimization ratio using above stated techniques. The overall optimization achieved was 81% on Museum dataset but only 18% on yelp dataset. The reason is that yelp dataset has string object, which cannot be optimized. Also string object is holding 66.93% share of overall memory usage. This result may be summarized as that the optimization achieved for a particular dataset solely depends on the column types and not the size.

Figure 3 shows the comparison of optimization for two different datasets. The legend shows deep size in memory. Diagram shows that for yelp dataset we achieved 78% optimization during down-casting which is equal to museum dataset, 90% optimization achieved during categorical conversion which is 8% less than museum dataset and 88% optimization achieved during typecasting which is equal to museum dataset.

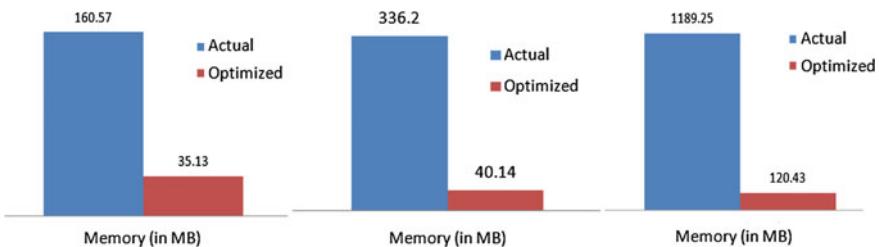


Fig. 2 a Downcasting **b** datetime conversion and **c** categorical conversion

Table 1 Comparison of optimization techniques on two different datasets

Criteria	Museum.csv	Yelp_review.csv
Dataset on disk	10.2 MB	3.53 GB
Dataset in memory (shallow)	7.40 MB	0.35 GB
Dataset in memory (deep)	46.8 MB	4.90 GB
Optimization% (downcasting)	77	78
Optimization% (typecasting)	88	88
Optimization% (categorical)	98	90
Overall optimization	81	18
Time to read from disk	NA	242 s
Space consumed for dataset	NA	7.9 GB out of 8 GB
Time to read chunks	<1 ms	377 ms
Space consumed for chunks	NA	4.7 GB out of 8 GB

3.2 Ingestion Bound Approach

Consider a case, where enough memory is not available, even after applying memory-saving techniques. The similar case is simulated with yelp dataset where one is unable to load dataset in memory and create data-frame. The yelp dataset of 4.9 GB takes 259 s to load into 3.9 GB of available memory out of 8 GB total memory using virtualization. The swapping started after 126 s which shows the disk's dual engagement. The optimal column types may be specified while reading the dataset. Programmers may help reduce swapping by freeing up dynamic memory that is no longer in use. A different strategy for working with datasets is required that don't fit into memory even after one optimizes types and filtered data-columns. Instead of trying to load the full dataset into memory, load and process into chunks.

The downside is that one has to write code that can work with just a portion of the data, store the intermediate results, and combine them at the end. While this is straightforward for simple tasks, it can be cumbersome for complex ones.

The entire dataset consumes approximately 45 megabytes of memory, by default, with the previously used approach. If only 1 MB memory is available, then each chunk should be below a specific memory threshold. Increase the number of rows until one is below 50% of decided threshold. In batch processing, task is broken down in equal parts known as bag. Each part is processed separately and combined later

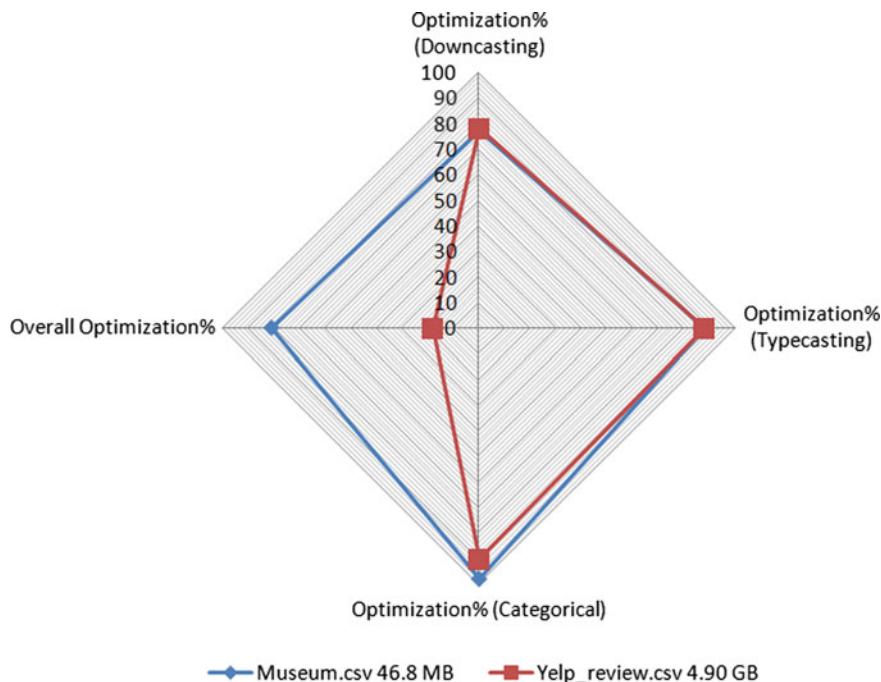


Fig. 3 Space optimization achieved

on. Parallel processing also requires data-frame/data-partition (referred as chunks here) so that many computers may work on parts in parallel. Data partition may be horizontal or vertical or both. Figure 4 shows chunk combining process where data is partitioned horizontally. Table 2 shows the comparison of two datasets on different criteria. While a separate list can be created and the values in each series object to that list can be appended, one starts to lose out on the performance advantage the pandas provides by converting back to native Python objects. In addition, each time many

Fig. 4 Chunk processing and combining process for yelp dataset

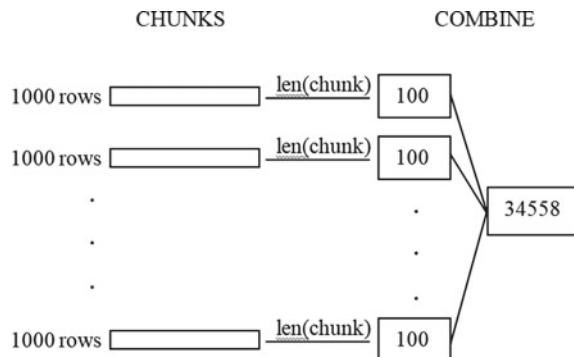


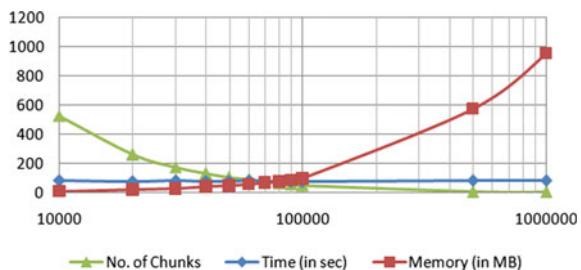
Table 2 Comparison of chunk processing on two different datasets

Criteria	Museum.csv	Yelp_review.csv
Dataset on disk	10.2 MB	3.53 GB
Dataset in memory (shallow)	7.40 MB	0.35 GB
Dataset in memory (deep)	46.8 MB	4.90 GB
Memory usage	Less than 1 MB (2%)	Less than 100 MB (2%)
Optimization% (chunking)	98%	98%
Time to read from disk	0.015 s	0.015 s
Space consumed for dataset	1 MB out of 8 GB	100 MB out of 8 GB
Time to process chunks	1.73 s	79.26 s
Space consumed by chunks	0.29 MB	53.8 MB

values to a native Python list may be appended, Python has to reallocate memory for that list, which consumes CPU time. Working with pandas objects is advantageous because of the optimizations under the hood which conserve both CPU and memory.

4 Experiments and Results

The experiments performed to evaluate the proposed techniques, results in the realization of efficacy. The hardware used for the experimentation is Intel(R) Core(TM) i3-6006U CPU@2.00 GHz, CPU@1.99 GHz dual core having 8.00 GB RAM and 64-bit operating system with \times 64-based processor. Figure 5 shows time complexity for large chunk sizes or large inputs and it is clear that the execution time remains constant if numbers of data-columns are fixed and less in number. The chunk size

Fig. 5 Execution time for chunks of different sizes

may be increased as per demand and still each chunk's memory footprint may be kept below threshold. The optimal chunk size varies and has to be calculated before fixing.

The processing time is approximately constant in case where we fix the chunk size in MB and uses large number of chunks or in case where we take less number of chunks but of large size. Table 2 compares the results of chunk processing for two different datasets on various criteria.

5 Conclusion

Velocity of big data is always time critical process. The two approaches for pre-processing data are memory bound approach and ingestion bound approach. The two approaches are good when used in respective manner, i.e., one after other. Memory bound approach is used for optimizing input on the go. The optimization takes considerable time if the flows are elephant flows. In this case, the best way is to consume data without optimization. Our contribution in this paper is to compare and contrast between the two proposed approaches. The experiments show that optimization can be achieved between 20% and 80% for big and streaming data, depending on the type of data. The future work invites work on optimization for speedup using GPU devices and heterogeneous computing.

References

- Brady HE (2019) The challenge of Big Data and data science. *Annu Rev Polit Sci* 22(1):297–323
- Budiman AR, Fanany MI, Basaruddin C (2017) Adaptive parallel ELM with convolutional features for Big Stream data. Faculty of Computer Science, University of Indonesia, Indonesia
- CERN (2021) Worldwide LHC computing grid. Retrieved Jan 07 2021, from WLCG: <https://wlcg.web.cern.ch/>
- Github (2016) Museum of modern art. Retrieved Oct 5 2020, from Github: <https://github.com/MuseumofModernArt/exhibitions/blob/master/MoMAExhibitions1929to1989.csv>
- Kaggle (2020) Yelp Rev. Retrieved Oct 7 2020, from Kaggle: <https://www.kaggle.com/yelp-dataset/yelp-dataset>
- Kolajo T, Daramola O, Adebisi A (2019) Big data stream analysis: a systematic literature review. *J Big Data* 6(1):1–30
- Kumar V, Sharma DK, Mishra VK (2020) Visualizing Big Data with mixed reality. In: System modeling and advancement in research trends. Mathura, India: IEEE, pp 85–90
- Kumar V, Sharma DK, Mishra VK (2021) Mille Cheval framework: a GPU-based in-memory high-performance computing framework for accelerated processing of Big-data streams. *J Supercomput* 77(3):1–25
- Kumar V, Sharma D, Mishra V (2021) Optimization and performance measurement model for massive data streams. In: Futuristic trends in network and communication technologies. Springer, Taganrog, Russia, pp 350–359
- Medeiros D, Neto H, Lopez M, Magalhães L, Fernandes N, Vieira A, Silva E, Mattos D (2020) A survey on data analysis on large-scale wireless networks: online stream processing, trends, and challenges. *J Internet Serv Appl* 11(1):1–48

11. Mehmood E, Anees T (2020) Challenges and solutions for processing real-time Big Data stream: a systematic literature review. *IEEE Access* 8:119123–119143
12. Mohanty S, Sharma R, Saxena M, Saxena A (2021) Heuristic approach towards COVID-19: Big data analytics and classification with natural language processing. In: *Data analytics and management*. Springer, Singapore, pp 775–791
13. Muthukrishnan S (2005) Data streams: algorithms and applications. *Found Trends Theor Comput Sci*
14. Pishgoo B, Azirani AA, Raahemi B (2021) A hybrid distributed batch-stream processing approach for anomaly detection. *Inf Sci* 543:309–327
15. Sahal R, Breslin JG, Ali MI (2020) Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. *J Manuf Syst* 54:138–151
16. Tantalaiki N, Souravlas S, Roumeliotis M (2020) A review on big data real-time stream processing and its scheduling techniques. *Int J Parallel Emergent Distrib Syst* 35(5):571–601
17. UCASE Software Engineering Research Group (2020) A stream processing architecture for heterogeneous data sources in the internet of things. *Comput Stand Interfaces* 70:103426
18. ur Rehman MH, Liew CS, Abbas A, Jayaraman PP, Wah TY, Khan SU (2016) Big data reduction methods: a survey. *Data Sci Eng* 1(4):265–284
19. Wibisono A, Mursanto P, Adibah J, Bayu WD, Rizki MI, Hasani LM, Ahli VF (2020) Distance variable improvement of time-series big data stream evaluation. *J Big Data* 7(1):1–13
20. Zhang F, Yang L, Zhang S, He B, Lu W, Du X (2020) FineStream: fine-grained window-based stream processing on CPU-GPU integrated architectures. In: {USENIX} Annual technical conference, pp 633–647
21. Zhang S, Zhang F, Wu Y, He B, Johns P (2020) Hardware-conscious stream processing: a survey. *ACM SIGMOD Rec* 48(4):18–29

Statistical Significance of Wilson Amplitude Towards the Identification and Classification of Murmur from Phonocardiogram



P. Careena, M. Mary Synthuja Jain Preetha, and P. Arun

Abstract The method of automatic recognizing of various valvular syndromes from the heart sound is a difficult job in cardiology. The features extracted from this may be time, frequency, or time-frequency domain and it bears a significant role in the automated systems to detect cardiovascular diseases. The examination of Phonocardiogram (PCG) signal provides very useful evidence related to the working of the heart. The time-domain based feature extraction techniques are logically simple and computationally less complex. In this paper, the statistical significance of Wilson amplitude at threshold levels (5–50 mV) is inspected towards the identification and classification of murmur from the Phonocardiogram. It has been noticed that the Wilson Amplitude at the threshold 5 mV is statistically more significant and it differ with a “P” value of 1.153×10^{-5} .

Keywords Heart abnormality · Box plot · ANOVA · Murmur phenotypes · Phonocardiogram · Statistical significance · Time-domain feature · Wilson amplitude

1 Introduction

The primary process to be carried out in any of the automated systems is to clearly meet its objective and this can be achieved by the correct selection of features extracted from the input given to the system. To extract these features, signal processing techniques bear a major role. These features mined by the signal

P. Careena (✉)

Department of Electronics and Communication Engineering, Amal Jyothi College of Engineering, Kanjirapally 686518, India

P. Careena · M. Mary Synthuja Jain Preetha

Department of Electronics & Communication Engineering, Noorul Islam University, Nagercoil 629180, India

P. Arun

Department of Electronics and Communication Engineering, St. Joseph's College of Engineering and Technology, Palai 686579, India

processing methods can be useful to detect, accurately traced, and to even classify a numerous problem like condition monitoring of mechanical as well as electrical arrangements by analyzing their vibration data, detection of diseases by examining various bioelectric potentials generated at cells of living things, etc. These features should own statistical significance and be efficient in computation. Basically, they belong to temporal, spectral, and spectro-temporal domains. Out of these, temporal features are mere and mathematically less complex than other domain features as they are because of no domain conversions.

Cardiovascular disease stays the number one threat to mortality and morbidity around the world. In view of the report of WHO, in 2016 almost 17.9 million people expired because of CVDs, addressing 31% of every single worldwide death. Out of this, 85% are because of cardiac infarction and cerebrovascular accidents [1]. There are so many bioelectric potentials are available and are useful to diagnose and classify numerous biological disorders. From them, Phonocardiogram (PCG) has been effectively utilizing for monitoring the heart rhythms. The techniques that incorporate the features extracted from the PCG signals have been effectively utilizing for the detection and classification of murmur [2].

A couple of strategies that includes features separated from the PCG records to address different problems of individual are given in the literature [3–12]. For murmur detection, Careena et al. [3], investigated the efficacy of the number of times a particular amplitude appears between a predefined time slot for different set values. They were reported that the signal amplitude at 20 mV set value was effectively spotting the presence of murmur when compared with other thresholds. Kobat and Dogan [4] proposed an automated system based on a feature assortment algorithm unit. The method incorporated a few supervised and unsupervised learning mechanisms. For, the former, they have considered few fundamental temporal features like RMS and peak amplitude of the pre-processed heart sound. Shen [5] proposed a method to detect various murmur qualities like musical, blowing-like, coarse, and soft. In addition to the main domain features, they have been incorporated phase space of heart signal. The features obtained from the signal have been given into artificial classifiers like SVM, KNN and Naive Bayes, and linear SVM. Jamal et al. [6], presented an automated Envelope-based computer aided auscultation framework for segmenting as well as parameter identification of PCG. They were obtained the reference information by abrupt variations of the signal amplitude computed by means of a peak detection algorithm. Sawant et al. [7], applied features like Zero crossing, energy and entropy of the energy, and the spectral features such as spectral entropy and Mel-frequency cepstrum coefficients (MFCCs). They have utilized the coefficients of optimal Tunable Quality Wavelet Transform (TQWT) based decomposition for feature extraction. Alkhodari and Fraiwan [8] introduced a system for diagnosing valvular heart diseases from the pre-processed heart sound by considering CNN and CNN bi-directional long short-term memory-based systems. Tuncer et al. [9], proposed a tent pooling (TEP) based strategy to classify heart sound. They also created a feature generation model by merging Petersen graph pattern (PGP) and TEP. The feature selection was performed via Iterative Neighborhood Component Analysis (INCA) and these features were given into classifiers like SVM, Decision

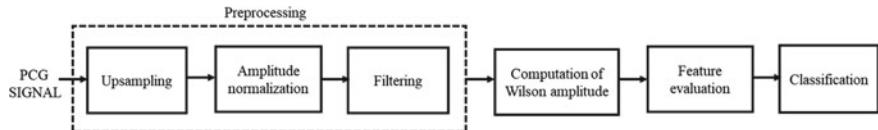


Fig. 1 Block schematic of the methodology

Trees, Linear Discriminant, and Bagged Tree classifiers. To segment S1-S2 hums, the instantaneous signal energy of the PCG had been utilized by Deperlioglu [10]. A resampled energy method was used to compute the energy and this information had been directly applied into a classifier known as stacked autoencoder network. Lin et al. [11], considered the slow oscillation elements holding the frequency energy distributions in the levels <100 Hz, 100–300 Hz, and >300 Hz for murmur detection followed by level of examining of the stenosis. After the feature separation via Empirical Mode Decomposition (EMD), pooling processes, and 1D convolutional method, patterns of features were mined from frequency relationship factor. The selected feature patterns were used to train the convolutional neural network. For diagnosis of cardiac disorders, Velusamy and Ramasamy [12] proposed a method by joining wrapper-based Boruta feature selection and embedded algorithm by combining classifiers like KNN, Random Forest, and SVM. During each iteration, the algorithm detected the substantial parameters by equating its Z-score with that of the erratically hopped prints of shadow features.

The statistical significance of Wilson amplitude (WA) towards the recognition of murmur and classification of its phenotypes from the Phonocardiogram signal gathered from the University of Washington heart sound data base [13] is examined in this paper. The main highlight of this paper is the statistical significance of Wilson amplitude at various threshold levels is tested and the separability given by this to detect and to classify different types of murmur like systolic, diastolic, and continuous murmur is also evaluated. The methods, mathematical computations and particulars of data repository utilized for the analysis are provided in Sect. 2. The statistical significance and the separability presented by the WA at various set points to spot the presence of murmur and its different phenotypes is explored in Sect. 3.

2 Methodology

The block diagram of the actions incorporated in the estimation of Wilson amplitude is exposed in Fig. 1.

Prior to the scheming of WA, the Phonocardiogram signals are pre-processed. The stages included in the pre-processing are upsampling by the factor two (for suitable sampling frequency), the normalization of amplitude between “+” and “-” unit level to regulate amplitude, and a high pass filter (to reject spectral factors beneath 10 Hz). As specified before, the WA is estimated to inspect its capability to detect

the murmur and then to classify murmur phenotypes from the PCG. The heart sound after normalization and up-sampling is presented in Eq. 1,

$$P_n(t) = \frac{P_i(t)}{\max|P_i(t)|} \quad (1)$$

where “ $P_i(k)$ ” is the PCG ranging from $1 \leq n \leq N$, $\max|P_i(k)|$ is the normalized signal, “ N ”, the total samples count and the sampling rate is “1/fs”. The Wilson amplitude is the instances that the distinction amidst two progressive magnitudes go over a specific limit [14]. It is mathematically represented as shown below,

$$\text{Wilson amplitude} = \sum_{t=1}^{N-1} f(|P_{n(t-1)} - P_{n(t)}|) \quad (2)$$

$$f(P) = \begin{cases} 1, & \text{if } P \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The analysis has performed on a sum of 40 heart sound samples (10 samples from each category) taken from the University of Washington data repository. The parameters of the heart sound are duration (more than 8 s for each record) and sampling rate (125 ms). The wave character of PCG resultant to normal and phenotypes of murmur like systolic, diastolic, and continuous murmur acquired from dataset are exposed in Fig. 2a-d.

The wave pattern of normal and the phenotypes of murmur are fully contradictory relative to the other. They differ as far as in terms of voltage and uncertainty appearances. For example, the signal characteristics of continuous murmur (Fig. 2d) is extra distorted and their mean voltage is relatively more when compared to normal (Fig. 2a) as well as murmur phenotypes (Fig. 2 b, c). The wave shape of PCG records analogous to normal and murmur acquired from two data set are shown in Fig. 2a-d. The wave pattern of normal heart sound and murmur are completely different from each other. They differ in terms of their amplitude and randomness characteristics.

The statistical significance of WA at various mV level set points (5, 10, 20, 30, 40, and 50) is verified for their capability to discriminate standard heart sound, murmurs like systolic, diastolic, and continuous murmur via Kruskal-Wallis one-way ANOVA. The separability provided by the features to discriminate normal heart sound from different types of murmur is qualitatively evaluated by Box-Whisker plot. All the analysis is performed in Matlab®.

3 Results and Discussions

In this work, the significance of WA at threshold of mV range like 5, 10, 20, 30, 40, and 50 is surveyed on the pre-processed PCG. The numerical values and range of

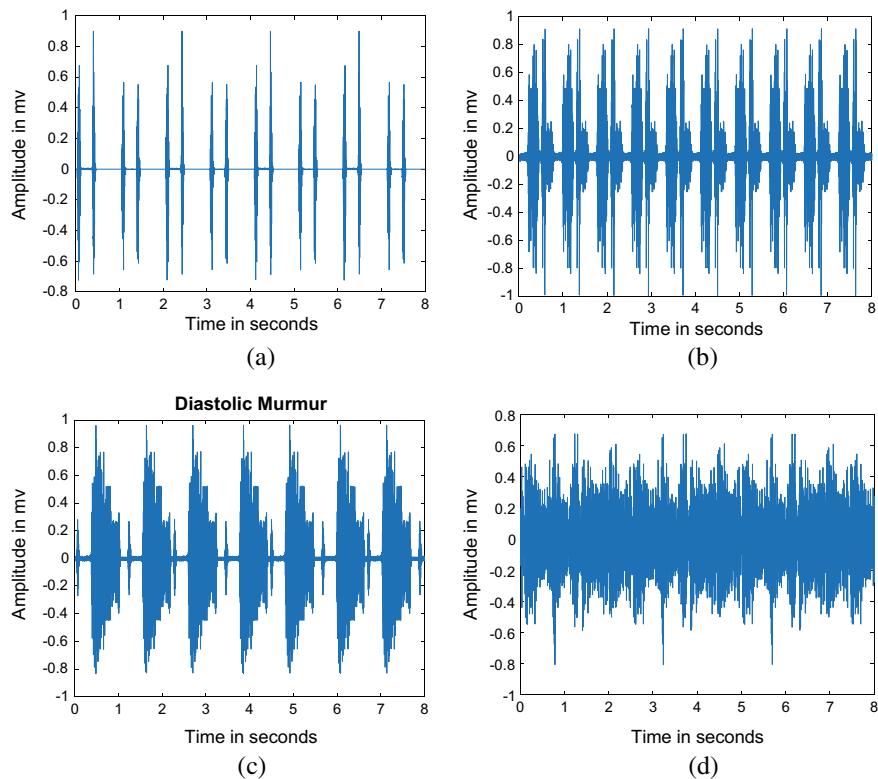


Fig. 2 Wave characteristics of normal and phenotypes of murmur **a** standard heart signal **b** systolic murmur **c** diastolic murmur **d** continuous murmur

WA at the above thresholds corresponding to normal and different classes of murmur like systolic, diastolic, and continuous murmur is supplied in Tables 1 and 2.

Table 1 The range of WA of normal heart sound and different types of murmur

S. No.	Feature	Types of heart signal			
		Normal	Sys. murmur	Dias. murmur	Cont. murmur
1	WA_5mV	14,215–16,557	13,283–93,400	27,100–116,452	92,834–119,327
2	WA_10mV	2038–3077	634–33,469	3093–46,611	11,212–16,515
3	WA_20mV	33–44	0–2671	20–5223	278–459
4	WA_30mV	0	0	0	0
5	WA_40mV	0	0	0	0
6	WA_50mV	0	0	0	0

Table 2 The numerical values of WA of normal heart sound and different types of murmur

S. No.	Feature	Types of heart signal	Sys. murmur	Dias. murmur	Cont. murmur
		Normal			
1	WA_5mV	15,538.00 ± 1093.99	37,292.75 ± 24,937.21	79,222.00 ± 46,141.17	106,435.92 ± 12,521.25
2	WA_10mV	2598.67 ± 503.66	11,228.33 ± 11,735.05	28,478.50 ± 22,472.60	14,088.58 ± 2563.89
3	WA_20mV	38.83 ± 5.42	696.67 ± 966.32	3055.08 ± 2686.82	371.92 ± 85.76
4	WA_30mV	0	0	0	0
5	WA_40mV	0	0	0	0
6	WA_50mV	0	0	0	0

Table 3 Kruskal–Wallis one-way ANOVA test—“ H ” and “ P ” values of features

S. No.	Feature	Chi-Square value (H)	Probability value (P)
1	WA_5mV	25.61	1.153×10^{-5}
2	WA_10mV	20.21	0.0002
3	WA_20mV	10.83	0.0127

It is observed from Table 1 that for 5 mV threshold, the range of different classes of heart sound are 14,215–16,557, 13,283–93,400, 27,100–116,452 and 92,834–119,327. For 10 mV threshold of Wilson amplitude, it is 2038–3077, 634–33,469, 3093–46,611, and 11,212–16,515. The range of values at 20 mV threshold for different classes of heart sound is 33–44, 0–2671, 20–5223, and 278–459. From Table 1, it is understood that, in case of continuous murmur, the WA at thresholds 10 and 20 mV is less than that of all other classes. For threshold levels of 30, 40, and 50 mV, there is no contribution of WA for murmur identification and for classifying the phenotypes.

From Table 2, it is noted that, the numerical values for 5 mV threshold level is $15,538.00 \pm 1093.99$, $37,292.75 \pm 24,937.21$, $79,222.00 \pm 46,141.17$ and $106,435.92 \pm 12,521.25$. For 10 mV threshold, the numerical values are 2598.67 ± 503.66 , $11,228.33 \pm 11,735.05$, $28,478.50 \pm 22,472.60$ and $14,088.58 \pm 2563.89$. For a threshold of 20 mV, the numerical values of all the different classes of heart sound as per the chronological order of representation in the table are 38.83 ± 5.42 , 696.67 ± 966.32 , 3055.08 ± 2686.82 , and 371.92 ± 85.76 . For all other threshold levels, the numerical value is zero.

The statistical significance of the features is inspected via Kruskal–Wallis one-way ANOVA and are specified in Table 3.

As given in Table 3, the H values attained by ANOVA test are 25.61, 20.21, and 10.83 for WA at threshold levels of 5 mV, 10 mV, and 20 mV, respectively. Chi-Square values are more prominent than the basic value of 12.838 for a degree of default rank of 0.05. The H values are outside the basic value. So that the above-mentioned features are appropriate to detect or to characterize at least one category of heart sound such as normal, systolic murmur, diastolic murmur, and continuous murmur. The “ H ” value of WA at 20 mV threshold is close to the critical value than other features. Hence it does not offer better separability than other thresholds. Wilson Amplitude at 5, 10 and 20 mV threshold conforming to the heart signal vary with a “ P ” value of 1.1536×10^{-5} , 0.0002 and 0.0127, respectively. The separability provided by the features to separate normal heart sound from different types of murmur like systolic, diastolic, and continuous is qualitatively evaluated by Box-Whisker plot. The Box-Whisker plot of WA at the above-mentioned thresholds for normal and different kinds of murmur are provided in Fig. 3a–d.

enlarge this page 12pt In the Box-Whisker plot of Wilson amplitude at thresholds 5, 10, and 20 mV, the box conforming to normal PCG is lie down adequately at a distance

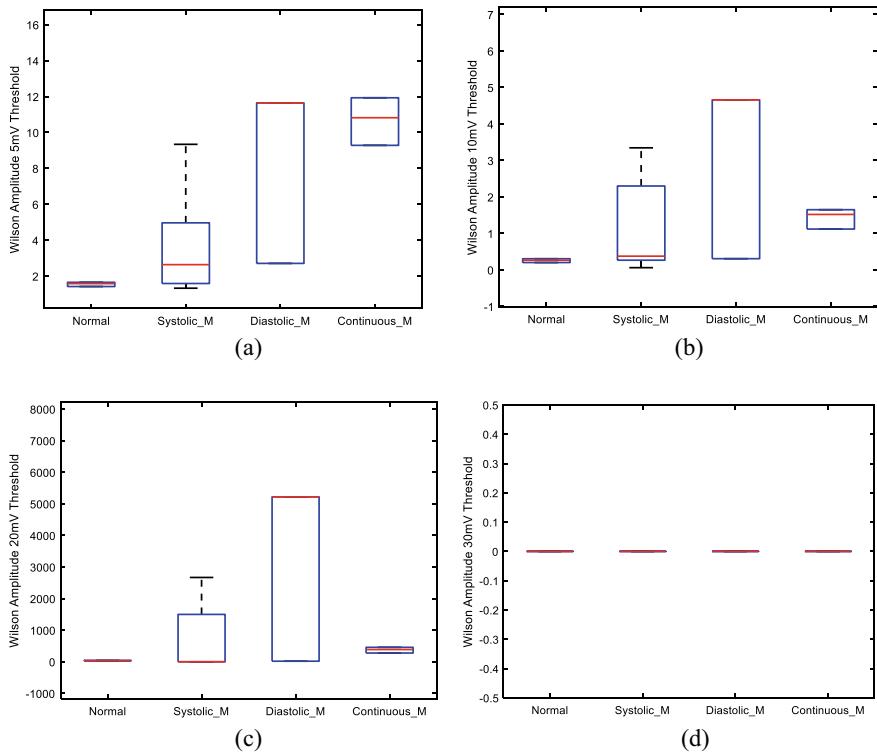


Fig. 3 Box Whisker plot of WA of normal heart sound and different types of murmur **a** Wilson amplitude at 5 mV threshold, **b** Wilson amplitude at 10 mV threshold, **c** Wilson amplitude at 20 mV threshold, **d** Wilson amplitude at 30 mV threshold

spatially from the boxes equivalent to continuous murmur. From the above discussions, it is inferred that the WA at 5 mV threshold is statistically more significant than WA at other threshold levels.

4 Conclusions

The statistical significance of Wilson amplitude (WA) at 6 differing set points in mV like 5, 10, 20, 30, 40, and 50 is tested and the separability given by this to detect and to categories different types of murmur like systolic, diastolic, and continuous murmur has been investigated. The statistical significance of the WA was verified for their capability to separate normal heart sound, systolic murmur, diastolic murmur and continuous murmur via Kruskal–Wallis one-way ANOVA and the separability provided by the features has been evaluated qualitatively by Box-Whisker plot. It was observed that the Wilson Amplitude at 5 mV threshold are statistically more

significant and differ with a “*P*” value of 1.153×10^{-5} . That means, the Wilson amplitude at other thresholds used in this study has serious constraints. The method incorporating Wilson amplitude at 5 mV threshold can be effectively utilized to detect the presence of murmur and to classify the phenotypes. The technique comprising the Wilson amplitude as a feature may resolve the problems associated with the manual auscultation.

References

1. WH Organization (2017) Cardiovascular diseases. www.who.int/mediacentre/factsheets/fs317/en/, last accessed 2019/08/13
2. Samanta P, Pathak A, Mandana K, Saha G (2019) Classification of coronary artery diseased and normal subjects using multi-channel phonocardiogram signal. *Biocybern Biomed Eng* 36:426–443
3. Careena P, Preetha M, Arun P (2020) Effectiveness of Wilson amplitude for the detection of murmur from the PCG records 656. *Lecture Notes in Electrical Engineering*. Springer, Singapore
4. Kobat MA, Dogan S (2021) Novel three kernelled binary pattern feature extractor based automated PCG sound classification method. *Appl Acoust* 179:1–9
5. Shen CH (2021) Feature extraction and classification of heart murmurs based on acoustic qualities. *IRBM* (2021)
6. Jamal N, Ibrahim N, Shaabani M, Mahmud F, Fuad N (2021) Automated heart sound signal segmentation and identification using abrupt changes and peak finding detection. *Procedia Comput Sci* 179:260–267
7. Sawant NK, Patidar S, Nesaragi N, Acharya UR (2021) Automated detection of abnormal heart sound signals using Fano-factor constrained tunable quality wavelet transform. *Biocybern Biomed Eng* 41:111–126
8. Alkhodari M, Fraiwan L (2021) Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings. *Comput Methods Programs Biomed* 200:14–24
9. Tuncer T, Dogan S, Tan R, Acharya UR (2021) Application of Petersen graph pattern technique for automated detection of heart valve diseases with PCG signals. *Inf Sci* 565:91–104
10. Deperlioglu O (2021) Heart sound classification with signal instant energy and stacked autoencoder network. *Biomed Signal Process Control* 64:1–9
11. Lin C, Wu J, Kan C, Chen P, Chen W (2021) Arteriovenous shunt stenosis assessment based on empirical mode decomposition and 1D-convolutional neural network: clinical trial stage. *Biomed Signal Process Control* 66:1–10
12. Velusamy D, Ramasamy K (2021) Ensemble of heterogeneous classifiers for diagnosis and prediction of coronary artery disease with reduced feature subset. *Comput Methods Programs Biomed* 198
13. Xi X, Tang M, Miran M, S and Luo Z (2017) Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable sEMG sensors. *Sensors* 17:1–20
14. University of Washington (2021). Heart sound and murmur. Available online: <https://depts.washington.edu/physdx/heart/demo.html>. Accessed 21 March 2021

Comparative Analysis on Machine Learning Methodologies for the Effective Usage of Medical WSNs



Shivani G. Dharmale, Snehal A. Gomase, and Sagar Pande

Abstract Due to its small scale, low cost, and ease of deployment, wireless sensor networks (WSN) are one of the most encouraging innovations for certain real-time implementations. WSN can alter progressively as a result of extrinsic or intrinsic factors, necessitating a devaluing valueless network overhaul. Traditional WSN methods are directly coded, making it difficult for networks to adapt continuously. Machine learning methodologies may be used to respond appropriately in such situations. Machine Learning is the method of a machine learning system learning from its observations and acting without the need for human interaction or reprogramming. A study of machine learning methods for WSNs is provided, spanning the years 2014 to 2019. We showcased numerous ML-based methodologies for WSNs in this study, along with their benefits, disadvantages, and variables that affect network lifetime. ML methodologies for synchronization, obstruction management, mobile sink planning, and power generation are also discussed. Eventually, we provide a methodological overview of the sample, as well as the explanations for using a specific ML methodology to solve a problem in WSNs and a summary of the remaining problems. The application of certain machine learning methodologies has been explained through the scenario of identifying anomalies in the medical WSN. The dataset utilized in this scenario is in real-time and obtained from Physionet.

Keywords Wireless sensor networks (WSN) · Machine learning (ML) methodologies · Medical WSNs · Anomalies identification

S. G. Dharmale · S. A. Gomase

Department of Electronics & Telecommunication Engineering, Dr. Rajendra Gode Institute of Technology & Research, Amravati, MH, India

S. Pande (✉)

School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India

1 Introduction

The Wireless Sensor Network (WSN) is a series of different distributed sensors. It is widely utilized for two techniques: surveillance and tracking. In WSN monitoring techniques, the single node detector aims to monitor an animal, an adversary, a person, traffic-flow, and so on, while in WSN surveillance applications, the node detector aims to control the climate, animal, or patient activity, and protection identification, among other items. Design and execution, localization, aggregation of information and sensor fusion, energy-conscious routing and grouping, planning, protection, and service quality are some of the challenges that users face when building a methodology utilizing WSN technologies. With the fast advancement in Micro Electro Mechanical Systems (MEMS) technologies, large-scale detectors in this area will be deployed in the immediate future. Large-scale WSN, on the other hand, invariably incorporates a huge volume of information into WSNs, which must be stored, distributed, and obtained. Owing to the detector's restricted resources and bandwidth, sending all information to a ground station for analysis and inference is simply impractical. As a result, Machine Learning methodologies must be implemented in WSNs. Recursively learning the characteristics of the system, ML methodologies change their actions quickly. Various ML methodologies are utilized in WSNs, including neural networks, fuzzy logical systems, evolutionary methodological systems, deep learning-based systems, and swarm intelligence-based systems. A category of artificial intelligence technology is machine learning, which inevitably generates activities utilizing example information without being explicitly programmed. The machine learning methodology allows for the construction of complex frameworks solely from evidence, without the necessity for skilled human interaction. The approaches derived from machine learning methodologies are more effective, affordable, and scalable as a result of this is free of intervention characteristics. Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four types of artificial learning strategies. These methodologies have the potential to dramatically minimize information communications thus maximizing the distributive properties of WSNs. Routing is the method of transferring information from a cluster sender head to a cluster receiver head or vice versa in a WSN. In contrast to mobile ad-hoc networks or wireless networks, routing in WSNs is a difficult process. There are three various techniques in routing procedure such as proactive techniques, reactive techniques, and hybrid technology. These can be elaborated as follows:

Proactive Technique in Routing Procedure. It's known as server-oriented, and it works by disseminating routing data regularly across all detector heads in the network to ensure reliable and precise routing tables.

Reactive Technique in Routing Procedure. It is on-demand such that the paths are found only whenever a node wishes to deliver a package. This procedure only recognizes the minimum number of hops needed to attain the target.

Hybrid Technology in Routing Procedure. It blends the most successful elements of constructive and reactive methods. When constructive routing is utilized within

a cluster and reactive routing is utilized across clusters, a hybrid routing approach may be utilized.

Minimal computational and storage requirements, automation and self-organization, power efficiency, usability, configuration meeting the requirements of traffic flows, and tolerance for in-network information aggregation are all considerations that must be considered when designing the routing procedure. In a WSN, cell power capacity and exchangeability power are the most significant considerations that impact routing. As a result, we are focusing on power-efficient routing in WSN, which is the primary aim of researchers in WSN to conserve energy. In WSN implementations, energy-efficient routing is important for expanding network lifespan. Clusters are used by the majority of routing procedures to increase network lifespan and power efficiency. LEACH, PEGASIS, and TEEN are power-efficient cluster procedures.

This article is organized into various sections. Section 2 deals with the discussion of various aspects of the research that are represented in current research articles. Section 3 deals with the discussion of various methodologies that are utilized for WSN based networks. Section 4 deals with the discussion of the conclusion and future works based on the utilization of machine learning methodologies.

2 Related Work

WSNs (Wireless Sensor Networks) are a common research topic these days. A WSN with a huge number of nodes is configured in a hostile area. They can be used for a range of mission-critical purposes, including health care, military surveillance, and civilian use. These networks have several security problems. Outlier identification is one of these problems. Outlier identification identifies information collected by certain nodes whose behavior differs from that of other nodes in a community of information. However, identifying those nodes can be challenging. Dwivedi et al. in 2018 [1] reviewed Outlier identification approaches focused on ML are debated, with the Bayesian Network (BN) appearing to be superior to other methodologies. The conditional dependence of the participating nodes in WSN can be calculated using the BN classification methodologies. This approach may also evaluate the importance of missing information. Khan et al. in 2017 [2] studied ML techniques that were introduced as a term. This study aims to resolve structural concerns in WSNs. As can be seen in this article, various attempts have culminated in the resolution of many architecture problems in WSN using several ML techniques. When using ML-based methodologies in WSNs, several restrictions must be considered, such as the network application's limited origins, which must track distinct activities, and also other operational as well as non-operational considerations. Due to its small scale, low cost, and ease of deployment, WSN is one of the most encouraging innovations for certain real-time technologies. WSN can alter progressively as a result of extrinsic or intrinsic issues, necessitating a devaluing valueless network overhaul. Conventional

WSN methods are directly coded, making it difficult for clusters to adapt spontaneously. ML methodologies may be used to react appropriately in such situations. ML is the method of a machine learning system learning from its interactions and acting without the need for human interaction or reprogramming. Kumar et al. 2019 [3] surveyed ML strategies for WSNs are discussed in this paper, which spans the years 2002–2013. We present numerous ML-based methodologies for WSNs, along with their benefits, disadvantages, and parameters that affect the lifetime of the network, in this study, which spans the years 2014 to 2018. ML methodologies for synchronization, obstruction management, mobile sink planning, and energy generation are also discussed. Eventually, it provides a methodological overview of the sample, as well as the explanations for using a specific ML methodology to solve a problem in WSNs and a summary of the remaining problems.

WSNs are usually utilized for surveillance and collecting raw detector information for eventual routing to a ground station in complex situations in task-related ecosystems. A host of technical difficulties must be solved to deploy WSNs in real-world settings. Standard strategies designed for a particular task find it impossible to respond to complex issues that are beyond the reach of the initial task. ML methodologies that can manage complex scenarios with an effective learning experience have recently been implemented in WSNs as a response to this challenge. The survey conducted by Kim et al. in 2020 [4] identified that Wide training times and datasets are required to achieve reasonable efficiency, which results in high power consumption, which is not ideal for resource-constrained WSNs. In the research, there have been articles on the use of ML strategies in WSNs. Nevertheless, there have been several articles on the use of DL methodologies in WSNs. Current advances in ML methodologies for WSNs are discussed in this study, with a focus on DL methodologies. The DL methodologies developed for different WSN technologies, as well as their DNN architectures, are discussed. WSN has gotten a lot of coverage in current times. Accumulating sensed data, converting information to the ground station in a power-efficient manner, and extending the network lifespan are all major problems in WSNs. Detector nodes in WSNs have a limited amount of storage. As a result, one of the main architecture problems in WSNs is reducing the amount of power used at the detector nodes. As a result, a range of routing strategies has been built to allow optimal use of the detector nodes' restricted power. In terms of power consumption, hierarchical routing procedures are the most well-known. Hierarchical routing protocols save a lot of power by utilizing a clustering strategy to capture and disseminate information. Yet, Bhanderi et al. in 2014 [5] mentioned that the WSNs must inevitably store, send, and obtain a vast volume of data from large-scale detector networks. Owing to the detector's restricted resources and bandwidth, sending all information to a ground station for analysis and inference is simply impractical. As a result, ML methodologies must be used in WSNs. These methodologies have the potential to dramatically reduce information communications by accurately using the distributive property of WSNs. The major purpose of this review article is to show that machine learning is a viable solution to a variety of complex decentralized challenges in WSNs, especially power-efficient routing.

WSNs keep track of changing conditions in real-time. Extrinsic variables or the device developers themselves are to blame for this complex behavior. Detector networks often use deep learning methods to respond to certain situations, avoiding the necessity for wasteful overhaul. ML also stimulates a slew of realistic strategies for maximizing resource use and extending the network's lifetime. Alsheikh et al. in 2014 [6] presented a comprehensive examinational study of ML approaches utilized to solve common problems in WSN from 2002 to 2013. The benefits and drawbacks of each suggested methodology are weighed against the various issues at hand. This article also has a comparison map to assist WSN programmers in designing ML technologies that are acceptable for their implementation problems. WSN are often used in research and architecture systems, military encroachment identification technologies, and civil technologies. WSN architecture may be technology-specific, posing a variety of problems and limitations. Power-aware routing is a perfect area for an analyzer to focus on when it comes to WSNs. Owing to the restricted power supply of detector nodes, data transfer, packet bandwidth, and information replication become crucial issues for routing algorithms to increase the network's lifespan. Prajapati et al. in 2018 [7] reviewed and discussed numerous WSNs problems with the ML-based method As well as various ideas proposed by various writers. Finally, future studies will focus on the positioning of mobile sinks and adaptive clustering for more power-efficient routing.

The software-defined network (SDN) is a networking model that was designed to offer greater functionality and address the shortcomings of existing network designs like WSNs. When SDN is introduced into a WSN, it results in SDWSN. Nevertheless, owing to the intrinsic limitations of SDN and WSN, SDWSN is confronted with a multitude of issues, including network and classification of internet traffic flow. Many methods have been suggested, including the use of ML, yet there are still so many problems that must be solved. Thupae et al. in 2018 [8] presented a review on the problems and methods of TC in SDWSN utilizing ML The aim is to recognize current procedures and problems to suggest solutions to improve these. The thesis conducted an analysis of recent research on TC in the study focusing on the aspects of business networks, SDN, and WSN, and reporting results. The results suggest that supervised or unsupervised learning was used in the methods to TC utilizing ML. Besides, TC faces problems such as power efficiency, shareable test results, and architecture. As a result, the use of machine learning to identify traffic flow in SDWSN is only in its initial stages, and it would continue to evolve to correctly classify regular and irregular traffic. Mamdouh et al. in 2018 [9] surveyed the various challenges that can affect both IoT and WSNs, as well as the ML strategies that have been created to combat them. The IoT is a network that allows digital computers, cameras, machines, and other entities to interact with one another without requiring human interaction. The IoT's core developing blocks are WSNs. Both the IoT and WSNs have a wide range of important and non-critical technologies that affect nearly every area of everyday life. Regrettably, these networks are vulnerable to a variety of potential problems. As a consequence, IoT and WSN protection has become critical. Consequently, the issue is compounded by the power constraints of the equipment utilized in these networks.

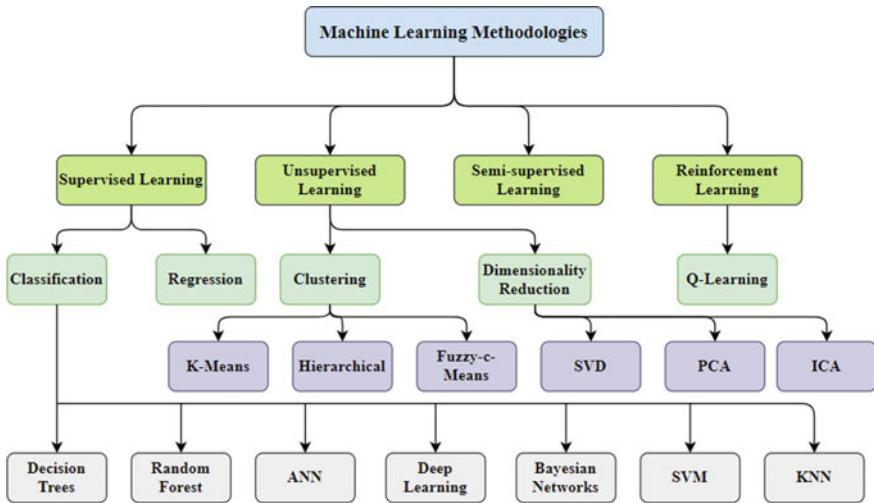


Fig. 1 Machine learning methodologies taxonomy

The literature review so far considered gives the proper information and challenges obtained with WSN (wireless sensor networks). The various solutions are generated for these challenges within WSN through machine learning as well deep learning methodologies as part of artificial intelligence for the generation of automatic solutions for the challenges of WSN.

3 Methodologies

Multiple machine training methods and their training protocols, which will aid understanding in the following pages. Besides, It included a short overview of evolutionary computation methods for WSNs. Machine Learning methods have been divided into four categories regard to learning strategies: These can be mentioned as supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (Fig. 1).

3.1 Supervised Learning Methodologies

One of the most significant information analysis methods in machine learning is supervised learning. It provides the machine a series of inputs and results, and it learns how to relate them during teaching. A method from an input vector with the best estimate of result y can be found after the training phase. The generation of the framework that represents interactions and correlation connections between the

various feature inputs and predicted objective outcomes is one of the key responsibilities of supervised learning methodologies. Localization, coverage issues, exception and fault identification, routing, MAC, information collection, scheduling, obstruction management, goal monitoring, incident identification, and power harvesting are all issues that supervised learning solves in WSNs. Regression and grouping are two types of supervised learning. Logic-based technologies like decision trees and random forests, perceptron-based technologies like artificial neural networks and deep learning, mathematical learning technologies like Bayesian model and SVM, and example-based technologies like KNN methodologies are all examples of classification. Regression is a form of supervised learning in which a series of attributes is used to estimate a quantity. The regression framework's parameters may be either continuous or quantitative. Regression is a very basic machine learning technique that forecasts reliable outcomes with the fewest possible residuals. The linear regression can be represented mathematically as mentioned in Eq. 1.

$$Y = g(X) + \varepsilon \quad (1)$$

Localization, networking problems, information collection, and power generation are all problems that regression is used to address in WSNs.

To improve readability, decision trees (DT) are a form of supervised machine learning method for classification based on a series of if–then guidelines. Leaf nodes and decision nodes are the two groups of points in a DT. Decision nodes represent the points in which the option of choices is considered, while leaf nodes indicate the outcomes in a specific direction of decision-making. DT generates a training framework based on decision rules extracted from training information to simulate a category or goal. The DT's key benefits are that it is straightforward, that it eliminates uncertainty in decision-making, and that it provides for a rigorous analysis. WSNs use DTs to resolve problems like interconnection, anomaly identification, information collection, and sink node destination address.

The random forest (RF) methodology is a supervised machine learning methodology that uses a set of trees, each of which provides a classification. The construction of an RF organizer and the estimation of outcomes are the two steps of the RF methodology. For larger databases and heterogeneous information, RF fits well. The outliers are correctly predicted using this method. A huge quantity of DTs can result from arbitrarily choosing a portion of training data and alienating parameters for each node of the tree. Because of the consistency of training data and overly sturdy DTs, the RF classifier has a lower recall rating as compared to many other implement ML classifiers. Due to the expletive dimensionality issue and strongly clustered results, current recommendation methodologies are facing major problems. For classifying highly spectral information, the RF classifier would be the most suitable tool. The RF methodology has been used to resolve a variety of problems in WSNs, including the coverage and MAC procedures. An artificial neural network (ANN) is a supervised ML methodology following the framework of a mankind neuron for categorizing the results. ANN interacted with a huge number of neurons that store data and deliver correct outcomes. Layers are usually used in ANN, with nodes linking the layers

and each node providing an active role. ANN efficiently classifies complicated and non-linear information collection, and unlike other classification systems, it has no input constraints. ANN is used in a variety of real-time WSN implementations, even though it has a higher computing specification.

Deep learning (DL) is a subsection of ANN and is a supervised machine learning method for recognition. Information learning interpretation techniques for multi-layer implementations are known as DL techniques. It is made up of small non-linear components that convert the interpretation from the lowest to the highest layers to obtain the best effect. It is based on mankind's nervous systems' connectivity designs and data processing. The main advantages of DL are its ability to derive high-level attributes from information, its ability to effort with or without tags, and the fact that it can be learned to achieve lots of goals. Computational biology, network analysis socially, data analytics, biomedical imaging, speech identification, and handwriting identification are just some of the applications. The benefits of DL have motivated WSN experts. Outlier and malfunction identification, routing, information quality prediction, and power generation are all problems that DL has discussed in WSNs.

The Support Vector Machine (SVM) is a supervised machine learning classifier that determines the best hyperplane to categorize information. Utilizing hyperplane to coordinate person analysis, SVM conducts the best classification. The bulk of the training information is obsolete once a boundary has been identified, and a collection of data aids in the boundaries recognition. Help vectors are the nodes that are utilized to determine the boundary. The best category identification from a provided collection of information is provided by SVM. As a result, the amount of attributes found in the training information has little bearing on the framework complexity of an SVM. As a result, SVMs are well adapted to active learning involving a significant range of attributes compared to the variety of training examples. Implementing SVM for WSNs also solved problems in WSNs such as localization, connection challenges, fault recognition, routing, and obstruction management.

In regression and classification, K-Nearest Neighbor (KNN) is the most simple and direct lazy, example-based learning methodology. As input from the attribute space, the KNN training dataset is used. The distance between specified training data and the testing data is frequently used in KNN classification. The Euclidean, Hamming, Canberra, Manhattan, Minkowski, and Chebychev distance functions are all used in the KNN methodology. The KNN methodology's sophistication is determined by the magnitude of the input information, with optimum efficiency achieved when the information is of the same length. This method looks for potential anomalies in the function space and determines them. Decreases the dimensionality of the data as well. The KNN methodology is utilized in WSNs for anomaly identification, malfunction identification, and information collection.

3.2 *Unsupervised Learning Methodologies*

There is no unlabeled output correlated with the input in unsupervised learning; even the framework tries to derive associations from the results. Unsupervised learning methods were utilized to classify a series of related trends, minimize complexity, and identify irregularities in the results. Unsupervised learning makes important benefactions to WSNs by solving problems such as networking, outlier recognition, routing, and information collection. Unsupervised learning is further classified into grouping algorithms such as K-means, hierarchical, and fuzzy-c-means, as well as dimension reduction algorithms such as PCA, ICA, and SVD.

From the considered dataset, the k-means methodology will efficiently shape a specific quantity of groups. Firstly, k random positions are deemed, with the remaining data points being allocated to the closest centers. After all of the data points in the dataset have been covered by groups, each group's centroid is readjusted. In each replication, the group's centroid improvements, and the methodology is replicated until there are no further improvements in the centroid of all groups. The k-means methodology has a time complexity of $O(n * k * r * d)$, where n indicates the total number of datapoints, k indicates the total number of deemed centroids, r indicates the number of replications, and d indicates the number of features in the considered dataset. The whole process is considered with the distance metrics.

The hierarchical grouping strategy divides related nodes into groups with a top-down or bottom-up sequence. In top-down hierarchical grouping, also known as divisive grouping, a large single division is broken iteratively before one group for each occurrence is found. Bottom-up hierarchical grouping, also known as agglomerative grouping, attributes each occurrence to a group using density functions. There is no necessity for previous knowledge of the number of groups in the hierarchical grouping method, and it is simple to enforce. This clustering approach has an $O(n^3)$ worst-time complexity and an $O(n^2)$ worst-space complexity. Information collection, synchronization, mobile drain, and power generation are some of the issues that hierarchical grouping is utilized to resolve in WSNs.

Bezdek introduced fuzzy-c-mean (FCM) grouping, also known as soft grouping, in 1981, utilizing fuzzy concepts to allocate observations to one or more groups. Groupings are defined using this methodology based on similarity measures such as strength, range, and communication. The methodologies can be used for one or more correlation tests, depending on the implementations or data sets. To determine the best group centers, the methodology iterates through the groups. For intermixed datasets, FCM produces the best grouping in comparison to k-means. It, like k-means grouping, necessitates previous awareness of the group scale. The FCM has a greater computation time than other grouping methods, and it is mostly determined by the number of groups, measurements, data points, and replications. This grouping technique is utilized in a variety of areas, including pattern identification, image segmentation, computational biology, and data analytics, among others. FCM is a methodology that can be utilized to resolve several problems in WSNs, including translation, accessibility, and smartphone sink.

Singular value decomposition (SVD) is a dimension reduction methodology that utilizes a factorization matrix methodology. This term refers to the transformation of a matrix into a result of matrices. The SVD algorithm can effectively reduce the data dimensions of a provided attribute space. The optimum low-rank presentation of the information is guaranteed by SVD. In WSNs, SVD is utilized to address problems such as routing and information collections.

Principle component analysis (PCA) is a dimension decrement approach to multi-variate statistical analysis function derivation. To minimize the dimensions of the dataset, the PCA blends all of the data and removes the least relevant data from the function space. PCA generates a correlation matrix of observed parameters as its output as principal components. PCA is often utilized in regression as well as in the identification of irregularities in statistics. Detectors in WSNs constantly capture data from their surroundings and send it to the ground station. PCA may diminish the dimensions of data in WSNs, either at the detector level or at the group head level, reducing transmission overheads. It prevents the obstruction issue by reducing buffer surplus at detector nodes or group heads in event-driven implementations. PCA is used in many WSN methodologies, including translation, fault prediction, information collection, and target detection.

Independent component analysis (ICA) dissolves multi-variate results into discrete sub-components and discovers a new basis for information representation. Non-Gaussian findings make up the sub-components in this situation. ICA is a more effective strategy than PCA, or to put it another way, it's a more advanced variant of PCA. ICA, unlike PCA, is capable of removing higher cognitive dependencies. ICA looked at information from several sources, including online information, visual images, psychometric assessments, business analytics, and social networking. The findings of several program information are time-series or a sequence of simultaneous findings.

3.3 Semi-supervised Learning Methodologies

The majority of information in real-world implementations is a mix of classified and unlabeled. Unsupervised learning methodologies operate well with unlabeled information, while supervised learning methodologies work well with labeled data. Semi-supervised learning was applied to operate with information that contained both classified and unlabeled information. Semi-supervised grouping for slightly labeled information, restricted grouping for both labeled and unlabeled information, regression for unlabeled information, and dimensions deduction for labeled information are all used. Semi-supervised learning has two different objectives: predicting labels on unlabeled information in the training dataset and predicting labels on possible research information. Semi-supervised learning is classified into two types based on these objectives: transductive learning and inductive semi-supervised learning. Learning that is partially supervised. Inductive semi-supervised learning trains a function $g : X \rightarrow Y$ such that f is assumed to be a strong predictor of upcoming

information, while transductive learning predicts the exact labels for given unlabeled information. Natural language analysis, online content detection, voice identification, spam detection, video monitoring, and protein sequence recognition, among other real-time implications, all benefit from semi-supervised learning. WSNs have lately used this learning method to address challenges like localization and fault identification.

3.4 Reinforcement Learning Methodologies

The Reinforcement Learning (RL) methodology learns by communicating with the surroundings and collects data to take specific actions. RL maximizes efficiency by evaluating the best outcome from the given scenario. One of the design-free reinforcement learning approaches is Q-learning. In Q-learning, each entity interacts with the surroundings and generates a state-action-rewards series of observations. $R(S, A)$ is a compensation matrix in which A and S represent a series of acts and states, accordingly. In Q-learning, the agent's behaviors are represented by the matrix $Q(S, A)$, which is proportional to the dimension of R with zero primary values. The present phase of the entity and the potential next phase are represented by the rows and columns of the matrix, accordingly. For all potential acts in the next phase, the operation principle that updates each element of the matrix Q with the total of the corresponding value in matrix R and the training variable is compounded by the highest benefit of Q .

4 Methodology for the Proposed Application

The proceeding scenario is considered: Sensing devices are connected to the patient's body for surveillance and communicate the measured physiological indicators to the sink of the network. The sink can be referred to as a base station or a smartphone. With larger memory and processing capabilities along with power supplies at its discretion, this particular base station is used for the analysis of data on the acquired records to look for abnormalities or raising alerts if the patient reaches a critical stage, or for the later usage, the records might be stored.

The gathered records are expressed in the form of a matrix A_{ij} where i denotes the time instance at which the record is collected and j denotes the collected physiological indicator measures. The structure of the matrix is represented as mentioned in Eq. 2.

$$A = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \quad (2)$$

To identify anomalous records, the proposed framework will utilize a classification methodology. A record is a set of several physiological indicator measurements collected at the same instance of time, i.e., a row in the matrix A as mentioned in Eq. 2. The correlation between forecasted and true indicator values will then be measured using a regression technique. This is done to ensure that the discrepancy between forecasted and true indicator values does not surpass a certain threshold, which in this case is 10%. If a report surpasses this limit, a correlation analysis is performed to distinguish between an anomalous report and the patient approaching a critical state.

An ideal scenario was considered for explaining the necessity of machine learning methodologies in the WSN aspect, that is nothing but identification of anomalies in medical WSNs based records. This mechanism has two phases, the first phase deals with classification, and the second phase deals with regression. The first phase is to classify the records into normal or abnormal categories. The second phase deals with the identified abnormal record used for identifying the parameter based on the threshold. This will allow us to undertake additional correlation analysis to differentiate among incorrect results and the patient approaching a critical stage. The proceeding qualities will be discussed in the remainder of the article such as heart rate, pulse oxygen saturation, pulse, body temperature, and respiration rate.

The methodology will be broken down into two stages: The first stage, a framework will be created to categorize the reports, and then the methodology will be put into action. The data will then be sent into the framework as inputs, where they will be rated as normal or abnormal categories. The proceeding classification methods will be used to classify the reports: SVM, Random Forest, and K-Nearest Neighbor. The results will be compared to see which methodology did better. We'll use regression methods like Linear Regression and Additive Regression to undertake additional correlation analysis and discriminate among incorrect reports and the real critical condition after an abnormal report has been detected. These regression techniques' outcomes will be compared to find which one gives the best performance. To considerably minimize the complexity of the dataset, classification is performed on it, so regression does not have to be applied to every characteristic for each occurrence. More detailed explanations can be obtained from the mentioned references [10] (Fig. 2).

5 Experimental Results

PhysioNet [11], an online repository of collected physiological signals, provided us with the data for this study. The resemble dataset will be used, which includes 121 recordings with a total of 12 characteristics. The proposed framework utilized the WEKA tool [12] to analyze the outcomes to assess the performance and efficiency of the various methodologies [13]. The following are the findings acquired after using the various categorization methodologies.

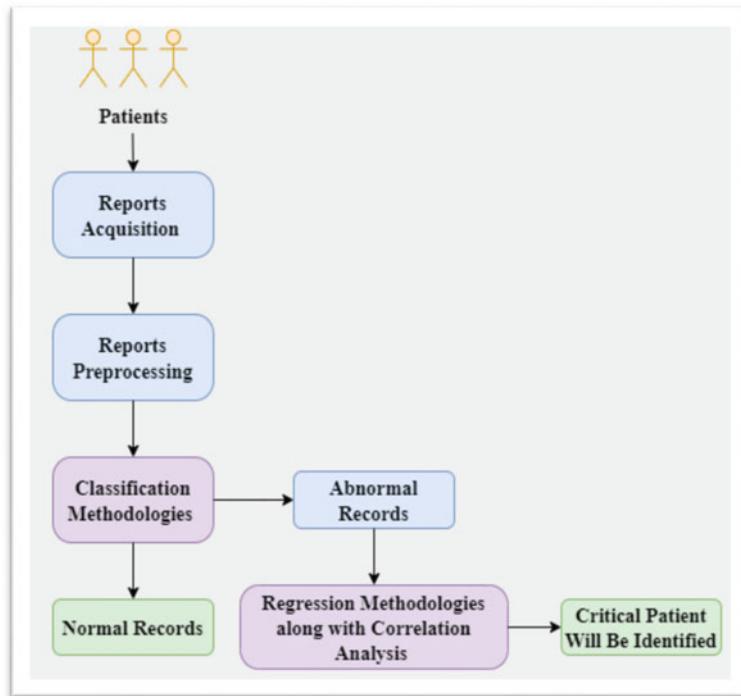


Fig. 2 Flow chart for the proposed methodology



Fig. 3 SVM ROC curve

The ROC (Receiver Operating Characteristics Curve) of SVM (Support Vector Machine) was presented in Fig. 3. The ROC (Receiver Operating Characteristics Curve) of KNN was presented in Fig. 4. The ROC (Receiver Operating Characteristics Curve) of Random Forest was presented in Fig. 5. The ROC curve shows how a

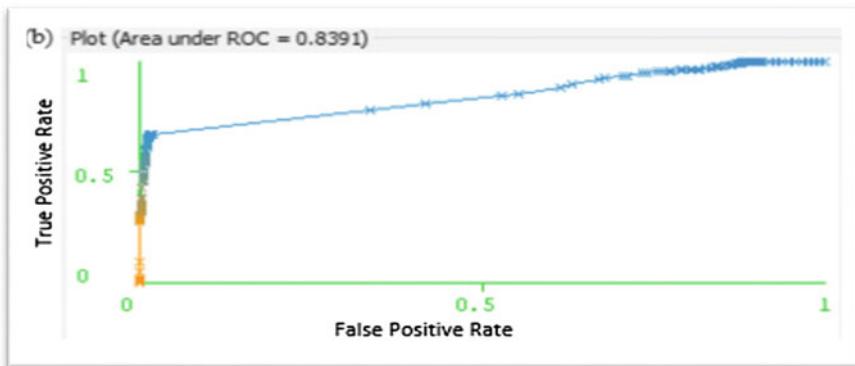


Fig. 4 KNN ROC curve



Fig. 5 Random forests ROC curve

binary classifier system performs when the discriminating threshold is changed. At various thresholds, the graph displays the true positive rate versus the false positive rate. As a result, the ROC curve depicts sensitivities as a function of specificities.

The mean absolute error for each classification methodology is shown in Fig. 6. Random Forests, together with the KNN method, deliver decent results once again, but only for tiny datasets like these. Even while KNN misclassifies considerably more cases than SVM, the mean error for both KNN and Random Forests methodologies is substantially lower [14–16]. We now go on to the regression phase of the proposed framework after using these classification techniques. The outcomes of using various regression techniques are displayed in Fig. 7.

The conceptual architecture of Additive Regression with KNN as the base learner provides the least mean error of all the applied techniques. Additionally, of all the methods used, this method delivers the best correlation coefficient. Lastly, Tables 1 and 2 offer comparative run-times of the different methodologies. Table 1 deals with classification methodologies and it demonstrates that KNN consumes the least amount of time to develop a classification model, following by SVM, and

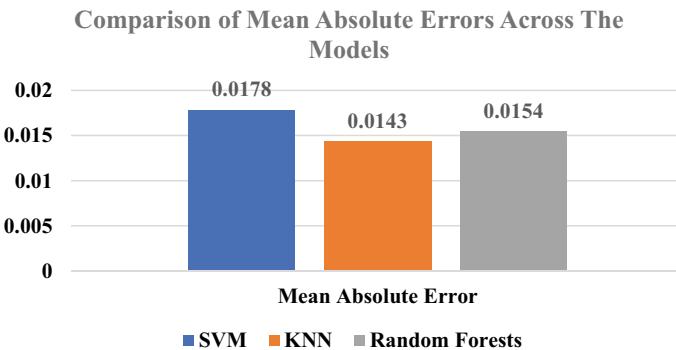


Fig. 6 Comparison of mean absolute errors across the classification models

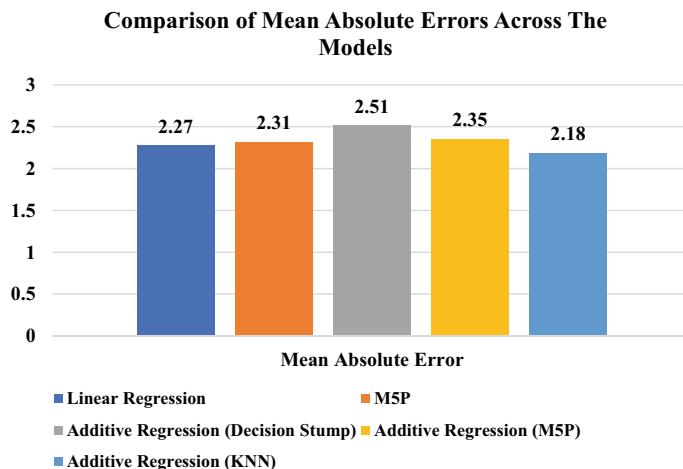


Fig. 7 Comparison of mean absolute errors across the regression models

Table 1 Run times of classification methodologies

Algorithm	Times (in s)
SVM	0.23
KNN	0.03
Random forest	1.45

Table 2 Run times of regression methodologies

Algorithm	Time (in s)
Linear regression	0.24
M5P	0.79
Additive regression (decision stump)	0.34
Additive regression (M5P)	2.12
Additive regression (KNN)	4.92

lastly Random Forests. It's important to know that the KNN methodology had the most misclassifications. Random Forest methodology consumes longer than SVM methodology. Table 2 illustrates the run times of regression methodologies in the same way. Linear Regression consumes the least duration to develop the model, whereas Additive Regression (with KNN) consumes the greatest time. It is crucial to remember, nevertheless, that Additive Regression (with KNN) had the lowest mean error.

To summarize the outcomes, we can state that Random Forest methodology gives the greatest overall classification performance, whereas Additive Regression (with k-NN) delivers the greatest overall regression performance for various workloads.

6 Conclusion and Future Scope

We can fairly assert that ML methodologies can play a key role in developing a failure and anomalies identification framework for utilization in medical WSNs based on the results of the studies. For anomaly identification in medical WSNs, the proposed framework blends the Random Forest methodology for classification aspects with Additive Regression methodologies for prediction aspects. For anomaly identification, this methodology uses analysis based on both geographical and temporal information. The proposed framework undergone to the test on a genuine medical dataset obtained from reputable data sources, and it was discovered that both methodologies outperform other prior studies represented methodologies. With the current increasing computing powers of computing machines and WSNs' comparably quick adaption in many fields such as medicine as well as health care, one can only assume that the application of ML methodologies in these fields is set to increase and establish its presence known even more.

It has introduced published currently ML-based methodologies for WSNs in this study. For the sake of clarity, we've quickly mentioned different machine learning methodologies. Localization, anomaly identification, malfunction node identification, filtering, information collection, MAC procedures, synchronization, obstruction management, power generation, and sink node route evaluation are several of the problems in WSNs discussed by ML methodologies. It has also spoken about subjects like synchronization, obstruction management, and power recycling that haven't been addressed in any previous study of articles. Besides, the ML-based methodology for WSNs has been compared and outlined in tabulated form. The mathematical charts outlined the influence of recent studies on ML-based algorithms for WSNs, as well as recommendations on which ML strategies to use to solve specific WSN problems. Eventually, it has touched on a few of the unresolved questions.

References

1. Dwivedi RK, Pandey S, Kumar R (2018) A study on machine learning approaches for outlier detection in a wireless sensor network. In: 2018 8th International conference on cloud computing, data science & engineering (confluence). IEEE, 2018, pp 189–192
2. Khan, ZA, Samad A (2017) A study of machine learning in a wireless sensor network. *Int J Comput Netw Appl* 4(4):105–112 (2017)
3. Kumar, DP, Amgoth T, Annavarapu CSR (2019) Machine learning algorithms for wireless sensor networks: a survey. *Inf Fusion* 49:1–25
4. Kim T, Vecchietti LF, Choi K, Lee S, Har D (2020) Machine learning for advanced wireless sensor networks: a review. *IEEE Sens J*
5. Bhandari M, Shah H (2014) Machine learning for wireless sensor network: a review, challenges and applications. *Adv Electron Electr Eng* 4:475–486
6. Alsheikh MA, Lin S, Niyato D, Tan H-P (2014) Machine learning in wireless sensor networks: algorithms, strategies, and applications. *IEEE Commun Surv Tutor* 16(4):1996–2018
7. Prajapati J, Jain SC (2018) Machine learning techniques and challenges in wireless sensor networks. In: 2018 second international conference on inventive communication and computational technologies (ICICCT). IEEE, pp 233–238
8. Thupae R, Isong B, Gasela N, Abu-Mahfouz AM (2018) Machine learning techniques for traffic identification and classification in SDWSN: a survey. In: IECON 2018—44th annual conference of the IEEE industrial electronics society. IEEE, pp 4645–4650
9. Mamdouh M, AI Elrukhsi M, Khattab A (2018) Securing the internet of things and wireless sensor networks via machine learning: a survey. In: 2018 International conference on computer and applications (ICCA). IEEE, pp. 215–218
10. Pande SD, Bhagat VB Hybrid wireless network approach for QoS. *Int J Recent Innov Trends Comput Commun* 4:327–332
11. "Physionet," <http://www.physionet.org/cgi-bin/atm/ATM>
12. "Weka data mining tool," <http://www.cs.waikato.ac.nz/~ml/weka/>
13. Divya K, Sirohi A, Pande S, Malik R (2021) An IoMT assisted heart disease diagnostic system using machine learning techniques. In: Hassanien AE, Khamparia A, Gupta D, Shankar K, Slowik A (eds) Cognitive Internet of medical things for smart healthcare. Studies in systems, decision and control, vol 311. Springer, Cham. https://doi.org/10.1007/978-3-030-55833-8_9
14. Pande SD, Khamparia A (2019) A review on detection of DDOS attack using machine learning and deep learning techniques. *Think India J* 22(16), ISSN: 0971-1260
15. Pande S, Khamparia A, Gupta D, Thanh DNH (2021) DDOS detection using machine learning technique. In: Khanna A, Singh AK, Swaroop A (eds) Recent studies on computational intelligence. Studies in computational intelligence, vol 921. Springer, Singapore. https://doi.org/10.1007/978-981-15-8469-5_5
16. Pande S, Gadicha AB, Prevention mechanism on DDOS attacks by using multilevel filtering of distributed firewalls. *Int J Recent Innov Trends Comput Commun* 3(3):1005–1008. ISSN: 2321-8169

Convolutional Neural Networks for Malaria Image Classification



Kanchan M. Pimple, Praveen P. Likhitkar, and Sagar Pande

Abstract Image processing applications are becoming more popular day by day across various domains in real-time aspects. This directly or indirectly having a great influence on artificial intelligence-based applications. These applications are widely using in agriculture and bio-medical images to built automated frameworks in the recognition of various diseases. Not only that, image recognition, image to text transformation, and object recognition are also popular applications. To bring these applications into reality, convolutional neural networks (CNN) acts as a backbone. Even if a lot of popular CNN architectures also are in usage, still researchers are looking for more effective frameworks. These CNN architectures play a vital role in advanced concepts like transfer learning. The CNN working aspects are mentioned more clearly in this article. The concept of customized CNN is explained through the application of malaria cell images. This customized CNN model attained a model accuracy of 95.91% and validation accuracy of 94.52%.

Keywords Convolutional neural network · Image processing · Convolutional layer · Pooling layer · Fully-connected neural (FCN) network · Fully-connected (FC) layer

1 Introduction

Deep Learning has had a huge influence on numerous scientific disciplines in the latest times. It has resulted in remarkable advancements in speech and image identification, as well as the ability to train artificial agents to overtake mankind in popular games based on deep learning such as Go and ATARI, as well as the creation of imaginative latest images and audio. Even before the occurrence of deep learning, most of

K. M. Pimple · P. P. Likhitkar

Department of Electronics & Telecommunication Engineering, Dr.Rajendra Gode Institute of Technology & Research, Amravati, India

S. Pande (✉)

School of Computer Science Engineering, Lovely Professional University, Phagwara, Punjab, India

these activities were thought to be difficult for machines to resolve, even in various studies related to science fiction. Evidently, this technology has a lot of applications in medical imaging. The related work contains a variety of presentations on the subject, varying from brief lessons and evaluations to blog articles and Jupyter notebooks to whole manuscripts. They all distribute a distinct determination and provide a unique perspective on this rapidly changing subject. Litjens et al. [1], for instance, produced excellent literature review articles by reviewing over 300 articles in their publication. Nevertheless, many other notable projects have emerged since then—apparently daily—making it challenging to write a literature review article that keeps up with the contemporary speed in the domain. Nevertheless, deep learning is rapidly gaining traction in other areas of medical image processing, and the manuscript falls short on issues like image reconstruction, for instance. Although having a general understanding of key processes in the domain is essential, real execution is just as vital in moving the field forward. As a result, researches like Breininger et al. brief's tutorial [2] are extremely useful for introducing the subject on a compiled code. Their Jupiter notebook model builds an immersive browser experience for implementing basic deep learning concepts in Python. In conclusion, they believe that the subject is too complicated and develops too rapidly to be adequately covered in a formal statement.

Digital image processing is the method of manipulating image data in digital format using the software. The implication of signal processing methods to the field of images, which can be considered as two-dimensional channel signals such as pictures or video, is known as image processing. Scanning or expanding an image with different kinds of functions, as well as other ways to obtain data from the images, is characteristic of image processing. It is one of the most commonly utilized applications for digital image processing. It also entails using software to analyze and manipulate photographs. Three steps are involved in the processing of images:

1. Images can be imported using optical devices such as a scanner or a camera, or they can be processed digitally.
2. This phase may involve image enhancement and data summarization, or the images can be analyzed to uncover rules that aren't visible to the naked eye. It implies that the image is been altered or evaluated.
3. This phase is concerned with getting the image processing result as an output. The end outcome will be an image that has been altered in some way or a report based on the interpretation of the results of the input images.

These steps can be identified in the following flowchart of digital image processing as represented in Fig. 1.

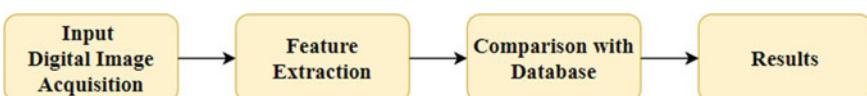


Fig. 1 Flowchart of digital image processing

Under computer science engineering, digital image processing is a very famous and quickly expanding field of development. Its expansion is fueled by innovative advancements in digital imaging, computer processing, and mass storage systems. For their suitability and accessibility, domains that have historically used analog imaging are now transitioning to digital circuits. Medicine, video processing, image processing, remote sense processing, and security monitor processing are also good instances. These roots of information generate a massive amount of digital image information every day, much more than can ever be checked systematically. Essentially, image processing is the manipulation of a two-dimensional signaled image by the software. An image or several attributes or traits associated with the image may be the consequence of image processing. The majority of image processing methodologies treat an image as a two-dimensional signal to which standard signal processing methodologies are applied. Certain vital applications based on image processing are computer vision, face recognition, video processing, remote sense processing, biomedical image processing, and analysis, biometric authentication, identification of signatures, character identification, and so on.

As the technology has been improving there is a lot of variation has been identified in methodologies utilized for various image processing applications. The methodologies include right from machine learning to deep learning. In current aspects, deep learning methodologies showing significant impact in identification, feature extraction, reliability of the processing. Particularly, the convolutional neural network (CNN) part of deep learning plays a vital role in the processing of images or videos. The popular software considered for image processing-related applications are MATLAB, Python and the popular platform on which these are executed are Jupiter notebook, spider, and Colab.

The present article is fragmented into various sections. Section 2 deals with the discussion of various literary aspects of recent works considered for this article. Section 3 deals with various popular deep learning-based methodologies considered for the processing of image-based applications. Section 4 deals with the conclusion and future scope of the related work.

2 Related Work

Razzak et al. in 2018 [3] presented an overview of, risks that are involved in biomedical image processing with the aid of deep learning methodologies. The healthcare industry is unlike any other industry across the globe. Customers contemplate a high quality of care and services, nevertheless of price, in this high-priority industry. Despite consuming a large portion of national resources, the healthcare industry has failed to meet society's assumptions. Medical professionals are usually the ones who examine the explanations of medical information. Because of rationality and the sophistication of the images, a medical professional's ability to clarify images is severely restricted; large differences occur among specialists, and exhaustion occurs as a result of their huge load. Deep learning is seen as offering innovative and precise

results for biomedical images and is seen as a crucial technique for potential application developments in the healthcare industry, succeeding in its achievement in many other real-world approaches. The deep learning framework and its effectiveness when utilized for biomedical image segmentation and classification are discussed in this article. The article highlights with a critique of the difficulties of deep learning methodologies in biomedical images as well as an active research problem.

Maier et al. in 2019 [4] provided the basics of deep learning methodologies in biomedical image processing in this article. This article also covers the reasons for the high popularity of the utilization of deep learning, then followed by reviewing the various fundamental concepts related to perceptron and simple neural networks that lead to understanding deep neural network concepts. These concepts are also discussed by considering the various trends based on image processing aspects. Kamarilis et al. in 2018 [5] reviewed the concepts and basic architectures of convolutional neural network (CNN) to deal with disease identification with plant leaf images. This article also includes various data sources, data disparity, pre-processing of the data, importance of data augmentation, and various metrics related to performance. Agrawal et al. in 2021 [6] intends to advantage the banking system by re-inventing a skilled check-based financial transactions software that involves automated system involvement. The proposed framework based on CNN and IDRBT datasets is considered. This framework attained an accuracy of 99.14%.

Cresson in 2018 [7] presented a model for DL approaches to be used with RS imagery and geographic information. The solution provided is based on two widely utilized open-source libraries: Orfeo ToolBox for RS image processing and TensorFlow for enhanced performance numerical computing. It can utilize deep neural networks without limiting the size of the images and is estimating effectiveness nevertheless of system setup. Wang et al. in 2020 [8] goal of this paper is to impart a thorough overview of the latest developments in image super-resolution utilizing deep learning techniques. In particular, the current research on SR approaches can be categorized into three groups and have supervised SR, unsupervised SR, and field-specific SR. It also goes over many other key topics like publicly accessible benchmark datasets and performance assessment measurements. Lastly, we will end this survey by emphasizing many future instructions and key questions that the neighborhood can discuss in the future.

Hegde et al. in 2019 [9] demonstrated how white blood cells are classified into 6 categories such as lymphocytes, monocytes, neutrophils, eosinophils, basophils, and abnormal cells. The model compares classical image processing techniques with deep learning methodologies to classify various WBCs. The model reviewed neural network-based classification outcomes and achieved an accuracy of 99.8%. The model utilized intensive training and transfer learning methodologies of convolutional neural networks to classify the various categories of WBCs. For intensive training CNN, an accuracy of about 99% was achieved.

Minaee et al. in 2021 [10] provided a detailed review of writing, involving a wide range of groundbreaking research for semantics and feature extraction, such as completely convolutional neural networks, encoder-decoder structures, multiscale and pyramid-based techniques, recurrent neural networks, attention models, and

generative models in confrontational situations. The framework look at the similarities, strengths, and problems of these models based on deep learning methodologies, as well as the most commonly utilized datasets, reports results, and explores exciting future study aspects in this field. Bhattacharya et al. in 2021 [11] focused to summarise the most recent studies on DL methodologies for COVID-19 biomedical image processing. The article then goes on to give an explanation of DL and its methodologies in healthcare in the previous few years. Following that, 3 utilize categories from China, Korea, and Canada to demonstrate DL methodologies for COVID-19 biomedical image processing. Ultimately, the article discusses several difficulties and concerns related to DL implementations for COVID-19 biomedical image processing, which are intended to spur more research into epidemic and disaster management, resulting in intelligent, healthy communities.

Affonso et al. in 2017 [12] explored how photos can be used to classify the performance of wood boards. It contrasts the utilize of DL, specifically CNN, with a mixture of texture-based attribute extraction methods and classical methodologies such as DT Induction methodologies, Neural Networks, K-Nearest Neighbors, and SVM. DL methodologies applied to image processing jobs have obtained predictive performance compared to conventional classification methods, according to published studies, especially in high-complex situations. Their encoded extracting features procedure is one roof of the objectives mentioned.

This section discussed various aspects that have been advanced in the past few years in the domain of image processing with deep learning methodologies. Also, it discussed improvements of the conventional deep learning methodologies to improved deep learning methodologies. The latest technology in CNN is transfer learning, and its importance is also explained.

3 Methodological Discussion

Improvements in the major concepts such as Computer Vision with Deep Learning technologies have been built and mastered over time, largely one whole methodology called Convolutional Neural Networks (CNN). These networks that are widely utilized in machine learning techniques for image processing and object recognition have a benefit over feedforward networks (FFN) in that they can regard featuring locality. CNN is a Deep Learning methodology that can receive input in the format of an image, allocate significance to specific facets or objects in the received input image, and distinguish between them. When compared to other classification techniques, the amount of pre-processing needed by a CNN is significantly less. While filters are hand-engineered in primitive techniques, CNN can understand this filtration with a sufficient amount of training. The structure of a CNN is influenced by the organization of the Visual Cortex and is similar to the pattern association among various neurons in the human being's brain. Individual neurons can only react to

stimuli in a small area of the field of vision called the Receptive Field. A compilation of these fields may be stacked on top of each other to covering the full visual field.

Reasons for CNN preferred over FFN Networks

There are various reasons for having priority for CNN over FFN networks especially while dealing with images or videos-based applications. The most popular reasons are mentioned as follows:

1. A bigger filter results in a less filtered image that means less data is transferred through the FC layer to the output layer. This results in a lower signal-to-noise (SN) proportion and a larger bias, yet it also eliminates the fitting problem by reducing the no. of parameters in the FC layer. This is a situation where the bias is high but the variance is low.
2. A small-scale filter results in a bigger filtered image that means more data is transferred through the FC layer to the output layer. Due to that the no. of parameters in the FC layer is enhanced, this results in a higher SN proportion and reduced bias, but it also increases the risk of overfitting. This is a situation where the bias is low but the variance is high.
3. If both CNN and FFN Networks have the same no. of hidden layers and similar architecture, a CNN would surpass an FCN network, particularly in image or video-based applications.
4. When conducting a classification model on incredibly simple binary images, the technique may show an average precision, however, when it moves to complex images with pixel dependencies across, the technique will have next to no accuracy.
5. Through the implementation of related filters, a CNN can effectively obtain the dependencies concerning spatial and temporal aspects in an input image. Because of that the reduced no. of parameters intricated and the reusability of weights, the structure does a smoother fitting to the input image dataset. Specifically, the network can be trained to better interpret the complexity of the input image.

Various Layers in CNN Architecture

The popular layers that act as building blocks of convolutional neural networks (CNN) are:

1. Input Layer
2. Convolutional layer
3. Pooling Layer
4. Fully Connected (FC) Layer.

The input layer is responsible for attaining the input in various formats of images that also include images with various color spaces. Some of the popular and existing color spaces of images are RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), HSB (Hue, Saturation, Brightness), HSL (Hue, Saturation, Lightness), CMYK (Cyan, Magenta, Yellow, and Black), and Grayscale. The CNN's job is to

condense the images into a format that's simpler to process while preserving important characteristics for accurate prediction. This is critical when designing a structure that is capable of learning characteristics while still being scalable to large datasets.

The Kernel is the component that performs the convolution functioning during the initial portion of a Convolutional Layer. With some kind of stride value, the kernel proceeds to the right until it aggregates the entire width. Proceeding on, it uses the same value to hop down to the left of the image and continues the procedure till the complete image has been navigated. The Kernel has the same dimension as the input image in the scenario of files with different channels. The outputs of matrix multiplication are summarized with the bias to produce a smothered single depth channel convolutional feature result. The main goal of convolution functioning is to retrieve high-level attributes from the input image, such as corners. There is no need to restrict CNN to just one layer of convolution. The first CNN is traditionally responsible for acquiring Low-Level attributes such as corners, color, orientations, and so on. The structure adjusts to the High-Level attributes as well as the additional layers, making us a network that has a complete comprehension of the considered image dataset. The procedure produces two kinds of outcomes as one of its outcomes in which the dimensionality of the convolved attribute is lowered in comparison to the input, and the other of its outcome in which the dimensionality is either enhanced or unchanged. This is accomplished by using valid padding in the earlier scenario and similar padding in the latter scenario.

The Pooling layer, like the Convolutional Layer, is beneficial for reducing the convolved spatial complexity of attributes. Throughout dimensionality deduction, the computational capacity necessary to process the information is reduced. It's also helpful for retrieving rotational and positioning irreducible significant characteristics, which helps keep the training process of the framework running smoothly. There are about three kinds of pooling operations. They are max pooling, avg pooling, and min pooling. But, majorly used operations are max pooling and avg pooling. Max pooling refers to maximum pooling and this operation identifies maximum values of the section of the image shaded by the kernel. Avg pooling refers to average pooling and this operation identifies average values of the section of the image shaded by the kernel. Min pooling refers to minimum pooling and this operation identifies minimum values of the section of the image shaded by the kernel. Max Pooling works as an error elimination as well. It removes all error action potentials and does de-noising and dimensionality deduction at the same time. Avg Pooling merely reduces dimensionality as an error elimination mechanism. As a result, one can conclude that Max Pooling outperforms Avg Pooling.

The specific layer of a CNN is made up of the convolutional and the Pooling operations containing layer. Based on the input image complexity, the no. of such layers can be expanded more and more to capture even more low-level particulars, although at the expense of more computing capacity. One can successfully be allowed the framework to learn the attributes after following through the above-mentioned procedure. After that, we'll flatten the result and feeding it to a standard FFN network for the aspect of classification. Using an additional FC layer is a low-cost methodology of learning non-linear configurations of high-level attributes represented by

the result generated by the layer using convolution operation. In that time, the FC layer is learning a presumably non-linear relation.

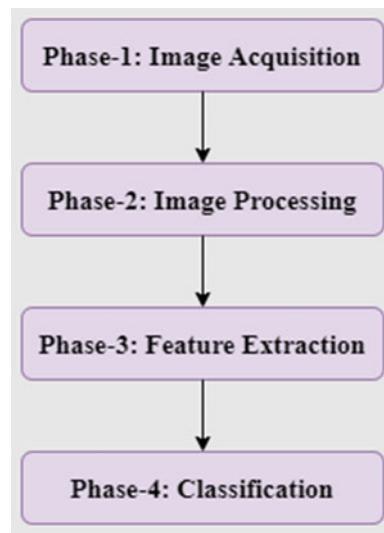
One will flatten the image into a directed graph now that one could transform it into a format appropriate for the ML Perceptron. Every epoch of training uses back-propagation to feeding the flattened result to an FFN network. The framework can differentiate between dominant and some low-level attributes in images throughout a sequence of iterations and classification utilizing the Softmax methodology. There are a variety of CNN structures accessible, all of that has played a role in developing methodologies that dominate and will continue to dominate AI in the coming years. Some of them are LeNet, AlexNet, VGGNet, GoogleNet, or Inception, ResNet, and ZFNet.

Proposed Methodology

The generalized flowchart for the classification of images can be mentioned as mentioned in Fig. 2. The generalized architecture for the classification of images consists of primarily four stages and those stages are Image acquisition, image processing, feature extraction, and Classification. Image acquisition mainly deals with obtaining the required image data set from a reliable source. Once, the image dataset is obtained, then image processing techniques need to be employed on the image dataset to remove the noises and to obtain all the images in the same size in terms of pixels.

When all the images are processed using image processing techniques then the processed images need to pass into the proposed CNN model. In that model, two phases will occur. In the first phase, features will be extracted from the images using convolution and pooling layers. Once features are extracted, the features will be passed into a flattening layer to obtain all the features into a single column. Then, it

Fig. 2 Generalized flowchart for the classification of images



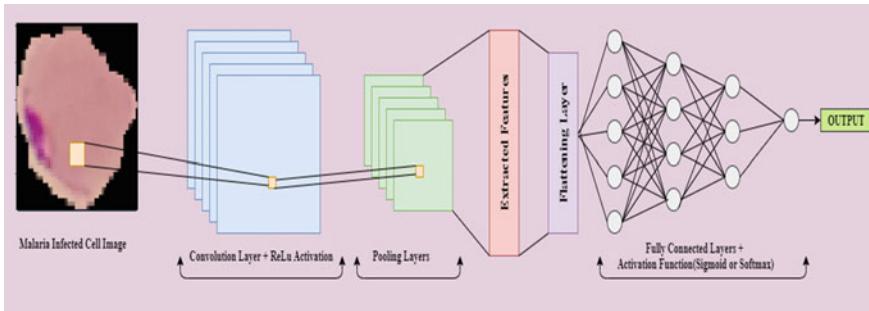


Fig. 3 Generalized CNN model for image classification

enters into the second phase. The single column of extracted features will be passed into fully connected layers or dense layers to get the classification. The generalized CNN model used for the classification of images can be mentioned as represented in Fig. 3.

Dataset Description

The malaria image dataset was obtained from the open-source provided by the National Institutes of Health which is a department of Health & Human Services of the U.S [13]. This dataset consists of 27, 588 images on a whole with classes of infected as well as uninfected images. This dataset is the most balanced as both the classes i.e., infected as well as uninfected have an almost equal number of instances.

4 Experimental Results

The customized CNN model was proposed to classify the malaria cell images into infected cells or not infected cells as mentioned in the previous section. The proposed CNN model consists of various layers as mentioned in the previous section such as convolution layer, pooling layer, fully connected layer along with the epochs, optimizer function, loss function, and dropout parameter. Drop out parameter used to avoid the overfitting of the model. In the case of underfitting, the number of epochs can be increased to obtain a better model. The proposed model architecture can be represented as mentioned in Fig. 4. The total parameters of the model are 826,529. All of those parameters are trainable parameters and none of the parameters are non-trainable parameters. The parameters considered for this model can be represented as mentioned in Table 1. The training and validation accuracies obtained through the proposed model are about 95. 91% and 94.52% respectively. The training and validation losses obtained through the proposed model are about 12.74% and 16.65% respectively. Comparison of the model training accuracy and the model validation accuracy is mentioned in Fig. 5. Along with that, the comparison of the model training loss and the model validation loss is mentioned in Fig. 6.

```

Model: "sequential"
-----
Layer (type)          Output Shape         Param #
=====
conv2d (Conv2D)        (None, 126, 126, 16)    448
max_pooling2d (MaxPooling2D) (None, 63, 63, 16)    0
dropout (Dropout)      (None, 63, 63, 16)    0
conv2d_1 (Conv2D)       (None, 61, 61, 32)     4640
max_pooling2d_1 (MaxPooling2 (None, 30, 30, 32)    0
dropout_1 (Dropout)    (None, 30, 30, 32)    0
conv2d_2 (Conv2D)       (None, 28, 28, 64)     18496
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 64)    0
dropout_2 (Dropout)    (None, 14, 14, 64)    0
flatten (Flatten)      (None, 12544)        0
dense (Dense)          (None, 64)           802880
dropout_3 (Dropout)    (None, 64)           0
dense_1 (Dense)         (None, 1)            65
=====
Total params: 826,529
Trainable params: 826,529
Non-trainable params: 0
=====
```

Fig. 4 The architecture of the proposed CNN model

Table 1 Parameters information that utilized in the proposed model

Parameter	Parameter value/parameter method
Optimizer function	Adam
Loss function	Binary cross-entropy
Epochs	7
Dropout	0.3
The activation function in intermediary layers	ReLU
The activation function in the final layer	Sigmoid

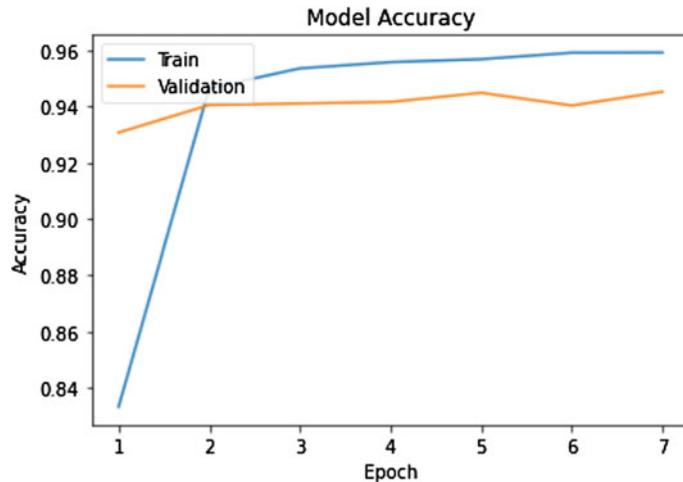


Fig. 5 Comparison of training and validation accuracies of the proposed model

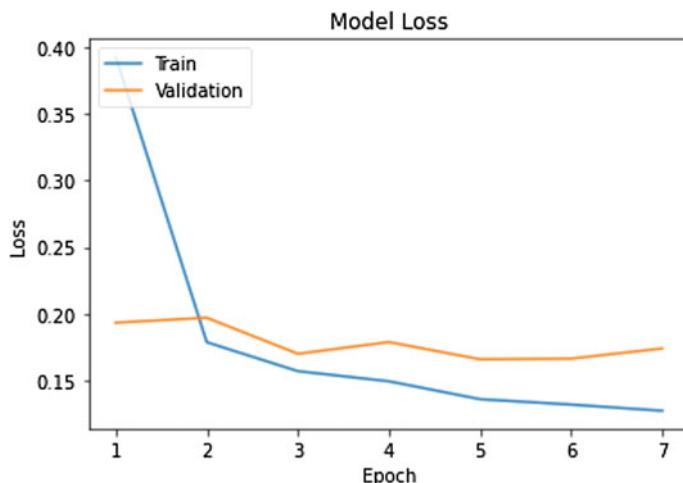


Fig. 6 Comparison of training and validation losses of the proposed model

5 Conclusion and Future Scope

The applications based on images or videos are being developed in artificial intelligence. These applications are majorly based on image and video processing. Such scenarios can efficiently deal with the aid of deep learning, especially, CNN. CNN is also a typical structure of Deep learning which has an efficient impact on applications based on images as well as on videos. A varied structure of CNN has been evolved and the complexity and the computational capacity of these models are also

enhanced but these do not affect the accuracy or precision of the result, on the other hand, the accuracy and precision for those models are enhanced. Overfitting is always a problem while dealing with deep learning models which are also similar to CNN models. But one can overcome this issue by proper tuning of hyper-parameters. The future aspect based on these CNN models is made through transfer learning which is generating much more impact on these applications.

For explaining the CNN architecture as well as working through an example of classification of malaria cells using images. The performance of the proposed model is fine but it can be improved by enhancing the layers, epochs primarily. The performance of the model can also be improved by considering the transfer learning techniques such as VGG-16, Inception, ResNet, and so on. Practically, it proved that the transfer learning models attaining better performance when compared to the customized CNN models.

References

1. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M et al (2017) A survey on deep learning in medical image analysis. *Med Image Anal* 42:60–88
2. Breininger K, Würfl T (2018) Tutorial: how to build a deep learning framework. <https://github.com/kbreininger/tutorial-dlframework>
3. Razzak MI, Naz S, Zaib A (2018) Deep learning for medical image processing: overview, challenges and the future. In: Classification in BioApps, pp 323–350
4. Maier A, Syben C, Lasser T, Riess C (2019) A gentle introduction to deep learning in medical image processing. *Z Med Phys* 29(2):86–101
5. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: a survey. *Comput Electron Agric* 147:70–90
6. Agrawal P, Chaudhary D, Madaan V, Zabrovskiy A, Prodan R, Kimovski D, Timmerer C (2021) Automated bank cheque verification using image processing and deep learning methods. *Multim Tools Appl* 80(4):5319–5350
7. Cresson R (2018) A framework for remote sensing image processing using deep learning techniques. *IEEE Geosci Remote Sens Lett* 16(1):25–29
8. Wang Z, Chen J, Hoi SCH (2020) Deep learning for image super-resolution: a survey. *IEEE Trans Pattern Anal Mach Intell*
9. Hegde RB, Prasad K, Hebbar H, Singh BMK (2019) Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. *Biocybern Biomed Eng* 39(2):382–392
10. Minaee S, Boykov YY, Porikli F, Plaza AJ, Kehtarnavaz N, Terzopoulos D (2021) Image segmentation using deep learning: a survey. *IEEE Trans Pattern Anal Mach Intell*
11. Bhattacharya S, Maddikunta PKR, Pham Q-V, Gadekallu TR, Chowdhary CL Alazab M, Jalil Piran M (2021) Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey. *Sustain Cities Soc* 65:102589
12. Affonso C, Rossi ALD, Vieira FHA, de Leon Ferreira ACP (2017) Deep learning for biological image classification. *Expert Syst Appl* 85:114–122
13. “NIH”, <https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets>

Identification of Characters (Digits) Through Customized Convolutional Neural Network



Swati C. Tawalare, Nikhil E. Karale, Sagar Pande, and Aditya Khamparia

Abstract Digitalization is showing a greater impact across the fields. The improved digitalization has brought various technicalities to object detection. In this aspect, the proposed framework deals with the identification of characters. This paper highlights the issue of identification of characters or digits in the CHARS74K data from handwriting, printed, natural image samples. This framework also illustrates the study performed on handwritten character recognition that describes the various manuscripts of various languages such as Urdu, English, Devanagari, Arabic. A customized CNN model has been proposed for identifying the various forms of digits in the dataset and the precision, the recall has been mentioned for individual digit forms.

Keywords Languages · OCR · Feature extraction · CNN · Classification

1 Introduction

Handwriting processing has become a challenging topic in which historical journals, paleographers, psychologists have been targeted for several decades. The script of personal [1] and psychological characteristics is also being examined. When the learning cycle begins with a replica of forms from a sample journal, each individual learns their literary style according to a personal interest in writing or combining template models. The handwriting of each individual is, however, unique [2], as well as an appropriate biometric behavioral method, can be used. Handwritten recognition approaches are typically classified into [3] text-independent and

S. C. Tawalare · N. E. Karale

Department of CSE, DRGIT&R, Amravati, India

S. Pande (✉)

School of Computer Science Engineering, Lovely Professional University, Phagwara, Punjab, India

A. Khamparia

Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India

[4] text-dependent approaches. Written data of the same textual content requires text-dependent methods for comparison. However, text-dependent techniques allow writer recognition, regardless of semantic content. Similarly, the types of recognition are differentiated by online [5] and offline [6], based on handwritten acquisition approaches.

Offline approaches employ digital handwriting images and use mathematical or structural attributes derived from handwriting images locally or globally. In addition to character forms, online techniques often take advantage of online attributes such as pressure and number of strokes to accurately describe an author. Here, we consider the offline data manuscripts and work on the detection of a specific character in the manuscript using deep learning and Image processing. The image processing framework is designed to detect, classify, track and predict specific features in images. Such frameworks must train to exhibit the features of their pictures in a stable and immutable way. The new study indicates that the activities implicitly acquired by deep learning models capable of detecting abstract standards invariant of the diversity of applications, such as vision, understanding, and ambiguity. Therefore, studies investigated the endless opportunities over the use of neural networks in different machine accessing activities, especially on challenges with categorization. In various applications, deep learning models have proved highly successful, in particular in visual data activities. Automatic identification of characters is some of the most significant computer vision and machine learning fields. It mainly corresponds to the perpetuation of the automated system and enhances connectivity with the human-computer in different technologies. As a robust content extraction and SVM, a high-end classifier for the identification of text recognition on a template sheet was used by the machine learning design based on the deep convolutional algorithm. Identifying digit numbers from unbalanced image classification is extremely challenging, as it has vast character variations. In this framework, a deep CNN model was used for digit recognition of the CHARS74K dataset.

This paper is organized as follows. Section 2 describes an overview of the paper. Section 3 describes various datasets and their language manuscripts. Section 4 includes various methodologies. Section 5 includes the results of the proposed method. Finally, ends with a conclusion.

2 Related Work

Hannad et al. 2016 [7] suggested a suitable solution that is dependent on tiny segments of the script. The handwriting model is fragmented into a group of small frames, and the pieces in each frame are described by histograms of well-known sculptural definitions. LPQ, LBP, and LTP are also used. The gap between all the various pixels of the fragments obtained by the two frames is calculated by two different samples. Scores of 94.89% were reached by the framework tested by the 411 IFN/ENIT Repository by the corresponding editors. Later, the study also extended [8] to examine HOG's efficiency as a depicter of small bits of literature. Al-aadeed [9] indicated a variety of

geometric parameters. These possibilities have path, tortuosity, and thickness which strengthen the conventional network and spatial characteristics. A Top-1 detection score of 70.08% for Arabic studies of 1017 authors, using KDA, was examined for both English as well as Arabic studies in the QUWI repository.

SVM is indeed one of the scholars' classifiers [10] that was used in the identification of scripted numbered symbols in the Indian language and Arabic. Cleber and Washington W. Azevedo [11] have suggested an integrated MLP-SVM method to consider the cursive handwriting aspect of the framework. Specialized local SVM also enhanced MLPs' capacity to detect similar characteristics. For cursive text classification [12], a combined KNN-SVM approach was used. To enhance KNN's output in handwritten characters, SVM has been added. Nasien [13] suggested a method for the identification of handwritten English characters. For attribute extraction and Support vector machine, FCC is used as an algorithm.

Hertel [14] has a fascinating method in utilizing the coevolutionary components which have been recently taught from the ILSVRC-12 database, showing that CNNs can upgrade standard extraction functions and also achieve a test failure rate of 0.459% for the MNIST database. It also has been shown how even the most related trend is focused for its successful results on CNN models. However, there may be several exemptions. For instance, the pattern formed out by Wang [15] with a test failure rate of 0.349%, using the Centering—SVDD intra-scale reactive rule and SIFT functionality. Furthermore, Zhang [16] developed a 0.42% test failure rate without improvement for predicting original data properties for deep learning training.

The failure rate about an MNIST was 0.45 and 0.33% for a group of the better-known complex systems in [17], which depends upon reinforcement training to build CNNs frameworks. Also, a 0.63% error rate was achieved at DEvol [18], using a genetic algorithm. Baldominos [19] published a paper in 2018 in which the network layer developed with linguistic progress and obtained a 0.368% test failure rate despite an increase in the volume of input and progressively extended this outcome through the network development of the CNNs [20] group up to 0.281%. Bochinski [21] which designs consisting of the group has developed using a genetic framework and has documented a test failure rate of 0.26%.

3 Datasets Description

A collection of data with suitable data is often an essential condition for quality testing to be usable for predicting the model and for getting better accuracy. Datasets such as MNIST, PE92, CEDAR, HCL2000, CENPARMI, UCOM, etc. are freely accessible for the languages like English, Urdu, Chinese, numerical digits, Arabic, etc. Some of the datasets and their description are mentioned.

Character name	Isolated	Initial	Middle	Final
Meem				
Seen				
Yaa				
Noon				

Fig. 1 Sample UCOM dataset

3.1 UCOM

This dataset, Fig. 1, contains Urdu language analysis. Both character and writer recognition can be used for this dataset. The dataset comprises 53,248 characters and 62,000 calligraphically scanned words of 300 dpi. This dataset has been created by 100 various writers, each of whom has written 6 pages A4-sized. The assessment of the datasets is dependent on 50 images of manuscript lines as train datasets with an error rate of 0.004–0.006% reported.

3.2 Cedar

The famous CEDAR dataset was created by researchers who belong to the University of Buffalo in the year 2002 and is one of the initial big man-written catalogs. Images have been calibrated at 300 dpi in CEDAR. The Sample character images were in Fig. 2.

3.3 IFN/ENIT

It is Arabic's most famous manuscript repository. It was created at the Braunschweig technical university, Germany in 2002 for recognition methods of Arabic handwriting. The collection of 26,459 handwritten pictures showing the names of the Tunisian cities and towns. These pictures are 212.211 characters from 411 editors.



Fig. 2 Sample CEDAR dataset

أولاد حفوز	أولاد حفوز	أولاد حفوز
أولاد حفوز	أولاد حفوز	أولاد حفوز
أولاد حفوز	أولاد حفوز	أولاد حفوز
أولاد حفوز	أولاد حفوز	أولاد حفوز

Fig. 3 Sample IFN dataset

This dataset, Fig. 3, has now been extensively used since its inception by researchers to classify Arabic characters effectively.

3.4 CHARS74K

In 2009, the University of Surrey scholars published Chars74k. The database comprises 74,000 pictures of Kannada (Indian) and English scripts. The archive



Fig. 4 Sample CHARs74K dataset

provides scenes from Bangalore city, India. 1922 photos were taken of labels, hoardings, ads, and retail items. The individual character segments have been manually segmented and results are displayed in boundary box segmentation. For classifying items, a bag of visual letters method was used, and 62 separate classes for English and 657 classes for Kannada were finally developed. A sample of this dataset can be seen in Fig. 4.

4 Proposed Method

An algorithm is instructed in manuscript OCR on a dataset and how well the alphabets and numbers can be effectively classified. Classification is the task for studying a method on given data input and labeling it into a predetermined class or group. The

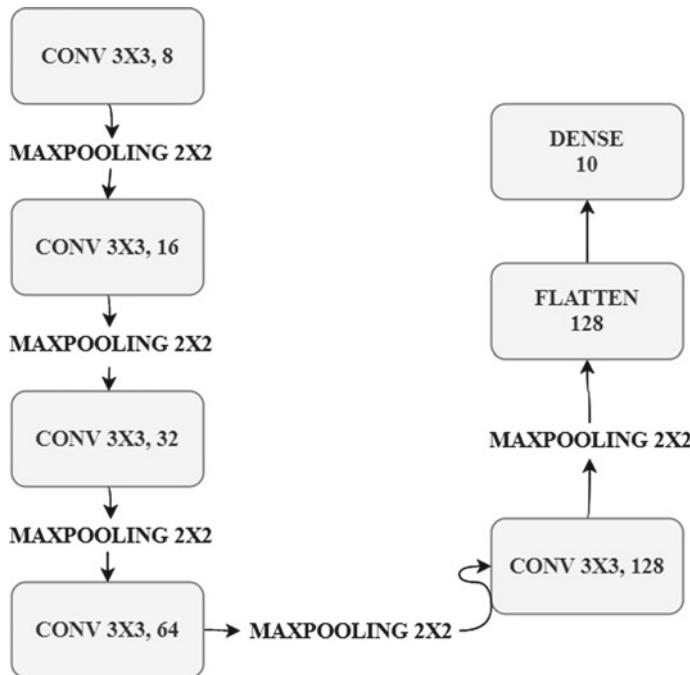


Fig. 5 Proposed CNN model

most popular classification strategies in OCR research studies from 2000 to 2019 have been explained in this section (Fig. 5).

An individual convolution component contains several samples of the matrix in almost all of the CNN architectures, so a feature vector with each surface of the matrix would be produced. For each position mostly on output and input of the layer contains a function in which the k th component of the function at the n th position will be the type of the n th position upon this input image provided by the k th matrix convolutional layer. For the convolutional network, the trainable criteria are based mostly on matrix parameters as well as layer thickness. To conduct the nonlinear mapping over the source images, an activation function has been used. The activation function of ReLU is often used in traditional deep neural networks to tackle the issue of sensitivity and tolerance. This framework also used the ReLU function in almost all of the levels excluding the output nodes in our research. The value of using ReLU over most functions is that it would be computationally effective because at any particular time just a few nodes are enabled. In the positive area, it should not saturate. ReLU converges 6 times faster than sigmoid and Tanh functions as well as the issue of gradient descent could also be solved by the ReLU function. In certain situations, ReLU is chosen because it is enriched in the lower region, rendering the gradient zero for such a region.

An additional batch normalization step precedes each layer in the CNN. Batch normalization validates each layer inputs, such that the issue of internal covariate change falls. To improve the consistency of a network, by deducting the sample average and varying by the sample standard deviation, batch normalization validates the outcome of a preceding activation function. The pooling layer among convolutions would be accessible for most CNN models. In essence, this layer limits the number of network parameters and computations. It also regulates overfitting by gradually growing the network spatial size. Although there is min-pooling, average pooling, max-pooling is required in most programs. Even as the name indicates, mostly the max value from a defined matrix can be taken out by max-pooling. This framework has been using the mini-batch gradient-based method for weight updating. This method is a subset of the technique for gradient descent which divides the test set into small amounts often used to measure prototype uncertainties and to modify statistical parameters. The mini-batch gradient-based method compromises stochastic gradient reliability and batch gradient descent performance.

5 Results and Discussion

The framework focused on the CHARS74K dataset by using a customized CNN model. Performance for a collective model of the CHARS74K dataset can be seen in Table 1 and also for individual digits, the precision and recall have been analyzed.

As earlier discussed, the CHARS74K dataset is a combination of printed fonts, handwritten fonts, and natural image fonts. The accuracy for the collective model based on their type was given in Table. 2.

Table 1 Performance of model on CHARS74K dataset

Digit number	Precision (%)	Recall (%)
0	98.78	98.86
1	95.25	96.94
2	98.58	97.85
3	95.62	96.23
4	98.63	97.52
5	97.28	98.29
6	98.16	97.82
7	98.10	98.46
8	99.52	98.31
9	97.39	96.85

Table 2 Accuracy for different digit forms

Digits type	Accuracy (%)
Printed digits	98.97
Handwritten digits	86.25
Natural image digits	88.63

6 Conclusion

The proposed CNN is designed for the classification of various digit representations. A higher precision has been obtained by the proposed CNN architecture. Chars74K and Single Digit. Also, it is found that the precision of handwritten characters and regular photo digits is enhanced by training the characteristics from printed script digit pictures. So, we could use this approach to detect the places (house or office) number, pin code sorting, and credit or debit card details recognition when provided with a great segmentation approach.

References

1. Fisher J, Maredia A, Nixon A et al (2012) Identifying personality traits, and especially traits resulting in violent behavior through automatic handwriting analysis. In: Proceedings of Student-Faculty Research Day, CSIS, Pace University, New York, USA. Al-Maadeed, S (2012) Text-dependent writer identification for Arabic handwriting. *J Electr Comput Eng* 2012, p 13
2. Srihari SN, Cha S-H, Arora H et al (2002) Individuality of handwriting. *J Forensic Sci* 47(4):1–17
3. Siddiqi I, Vincent N (2010) Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. *Pattern Recognit* 43(11):3853–3865
4. Bensefa A, Paquet T, Heutte L (2005) A writer identification and verification system. *Pattern Recognit Lett* 26(13):2080–2092
5. Shrivram A, Ramaiah C, Govindaraju V (2014) Data sufficiency for online writer identification: a comparative study of writer-style space vs. feature space models. In: 2014 22nd International conference on pattern recognition (ICPR), Stockholm, Sweden, 2014, pp 3121–3125
6. Gope B, Pande S, Karale N, Dharmale S, Umekar P Handwritten digits identification using MNIST database via machine learning models, IOP Conf Ser: Mater Sci Eng 1022
7. Hannad Y, Siddiqi I, Merabet YE et al (2016) Arabic writer identification system using the histogram of oriented gradients (hog) of handwritten fragments. In: Proceedings of the Mediterranean conference on pattern recognition and artificial intelligence, Tebessa, Algeria, 2016, pp 98–102
8. Al-Maadeed S, Hassaine A, Bouridane A et al (2016) Novel geometric features for off-line writer identification. *Pattern Anal Appl* 19(3):699–708
9. Liu C-L, Suen CY (2009) A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters. *Pattern Recogn* 42:3287–3295
10. Azevedo WW, Zanchettin C (2011) A MLP-SVM hybrid model for cursive handwriting recognition. In: Proceedings of international joint conference on neural networks. California, USA, 2011, pp 843–850
11. Shen L-L, Ji Z (2009) Gabor wavelet Selection and SVM classification for object recognition. *Sci Direct* 35:350–355

12. Nasien D, Haron H, Yuhaniz SS (2020) Support vector machine (SVM) for English handwritten character recognition. In: 2010 Second international conference on computer engineering and applications
13. Hertel L, Barth E, Käster T, Martinetz T (2015) Deep convolutional neural networks as generic feature extractors. In: Proceedings of the 2015 international joint conference on neural networks, Killarney, Ireland, 12–16 July 2015
14. Wang D, Tan X (2016) Unsupervised feature learning with C-SVDDNet. *Pattern Recogn* 60:473–485. [CrossRef]
15. Zhang S, Jiang H, Dai L (2016) Hybrid orthogonal projection and estimation (HOPE): a new framework to learn neural networks. *J Mach Learn Res* 17:1286–1318
16. Baker B, Gupta O, Naik N, Raskar R (2017) Designing neural network architectures using reinforcement learning. In: Proceedings of the 5th international conference on learning representations, Toulon, France, 24–26 April 2017
17. Davison J (2018) DEvol: automated deep neural network design via genetic programming. 2017. Available online: <https://github.com/joeddav/devol>. Accessed 4 May 2018
18. Bochinski E, Senst T, Sikora T (2017) Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In: Proceedings of the 2017 IEEE international conference on image processing, Beijing, China, 17–20 Sept 2017, pp 3924–3928
19. Baldominos A, Saez Y, Isasi P (2018) Evolutionary convolutional neural networks: an application to handwriting recognition. *Neurocomputing* 283:38–52. [CrossRef]
20. Baldominos A, Saez Y, Isasi P (2018) Model selection in committees of evolved convolutional neural networks using genetic algorithms. In: Intelligent data engineering and automated learning—IDEAL 2018; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2018, vol 11314, pp 364–373
21. Baldominos A, Saez Y, Isasi P (2019) Hybridizing evolutionary computation, and deep neural networks: an approach to handwriting recognition using committees and transfer learning. *Complexity* 2019:2952304. [CrossRef]

Breast Cancer Detection Using Image Processing and CNN Algorithm with K-Fold Cross-Validation



Pruthvi Tilekar, Purnima Singh, Nagnath Aherwadi, Sagar Pande, and Aditya Khamparia

Abstract Breast cancer is the most well-known kind of disease in lady overall representing 20%, everything being equal. That brought 1.68 million new cases and 522,000 Passings. One of the serious issues is that ladies regularly disregard the indications, which could cause more unfriendly consequences for them accordingly dropping down the endurance potentials. In created nations, the endurance rate is albeit high, yet it is a space of worry in the agricultural nations where the 5-year endurance rates are poor. In India, there are around 1,000,000 cases each year and the five-year endurance of stage IV bosom disease is about 10%. Accordingly, it is vital to recognize the signs as ahead of schedule as could be expected.

Keywords Image processing · Convolutional neural network (CNN) · Transfer learning · Deep learning · Invasive ductal carcinoma (IDC)

1 Introduction

The thought is to utilize pathology test pictures and characterize them as IDC (+) and IDC (−). Precisely recognizing and classifying bosom malignant growth subtypes is a significant clinical errand, and robotized techniques can be utilized to save time and lessen error. The obsessive tests incorporate pictures of the tissues, the undertaking

P. Tilekar · P. Singh · N. Aherwadi · S. Pande (✉)

School of Computer Science Engineering, Lovely Professional University, Phagwara, Punjab, India

P. Tilekar

e-mail: tilekar.12008745@lpu.co.in

P. Singh

e-mail: singh.12010242@lpu.co.in

N. Aherwadi

e-mail: aherwadi.12002658@lpu.co.in

A. Khamparia

Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India

is to prepare a PC to utilize these pictures and react on whether the individual is IDC (+) or IDC (-). Since it is a clinical field issue it is significant that affectability of the yield ought to be high [1].

Our information includes pictures with the classes composed on information record name, subsequently, we would have to extricate the class name from it and make a segment to store. We likewise need to part the dataset into the preparation set, approval set and testing set. Testing set for checking how great the model chips away at totally concealed information and approval set to check and stay away from under-fit or over-fit, they will likewise assist with choosing the best model. One hot encoding will be done in classes segment so it could work better with our model. Picture handling step is additionally needed to diminish the pixel range from 0–250 to 0–1. After CNN model to be utilized to foresee the class, CNN makes a successful design the 2D construction of the picture; subsequently, it would be the awesome use, taking into account that we are working with the portraits. We have implemented testing for dataset accuracy, sensitivity and specificity by applying CNN, image augmentation and transfer learning.

2 Literature Review

Reference [1] Creator Anuj Kumar Singh and Bhupendra Gupta proposed a Max-Mean and Least-Variance procedure for lump acknowledgment. Test results show the viability of the methodology. Recognition stage is trailed by division of the tumor area in a mammogram picture. This methodology utilizes basic picture handling procedures, for example, averaging and thresholding through visual investigation, obviously this strategy is effective in sectioning the malignancy locale of mammogram. Alongside division, pixels of malignant growth locale are additionally recognized. Their strategy is basic and quick due to utilizing essential picture preparing methods. Their strategy can likewise be useful in other clinical imaging applications, design coordinating, and include extraction. The primary downside of this technique is the manual determination of limit boundary and size of averaging channel. For future degree, they are willing to work on decreasing the steadfastness on boundary to make our technique versatile to various pictures.

Reference [2] Yousif M. Y Abdallah proposed a framework, where addresses the picture handling strategies contrast improvement, clamor diminishing, surface investigation and parceling estimate. The mammography pictures kept in great to monitor the quality. Those techniques plan to increase and sharpen the picture force and kill commotion from the pictures. The collection factor of expansion relies upon the setting tissues and kind of the bosom sores; henceforth, a few injuries gave preferable improvement over the rest because of their thickness. The computation speed investigated results were $96.3 \pm 8.5 (p > 0.05)$. The outcomes showed that the bosom sores could be improved by utilizing the proposed picture improvement and division strategies. In this examination, they offered new procedures, which would help in the location of the tumor in mammography. In view of the consequences of

the division and improvement, they finished up the two techniques would improve the pictures analytic worth. Future works need to zero in on and study identification of the all-bosom peculiarities utilizing other imaging modalities.

Reference [3] Authors present a Computer-Aided Diagnosis (CAD) approach for the determination and characterization of patients into three conditions (harmful, favorable, and ordinary) from pixel mammogram pictures. For the grouping task, they investigate and look at three remarkable classifiers: Support Vector Machine (SVM), k-Nearest Neighbor (K-NN), and Random Forest (RF) to examine their precision in dynamic. Likewise, they examine the impacts of pre-prepared mammogram pictures prior to entering the classifier, which brings about higher successful order. The proposed approach intends to mechanize the order and division measure in mammogram examination. The sorts of information that should be characterized incorporate typical, amiable, and threatening ailments. The impacts of AI calculations have started to be investigated in a few application spaces and the clinical field is one of them. In this unique situation, we have tried three directed anticipated models SVM, K-NN, and RF to decide their general exactness to effectively group mammogram picture circumstances, where RF accomplished the most noteworthy precision measurements for multi-class and twofold arrangement, just as by utilizing improved and crude pictures. Moreover, this work has portrayed the high effect of picture pre-handling arrangements for improving precision in the grouping interaction.

3 Proposed Methodology

3.1 CNN (*Simple Benchmark Model*)

This method takes data picture and assigns significance to the different objects or points in picture and then separates them from each other. Pre-designed convolutional network is significantly less by diverging course of action estimations. With handmade un-refined steps channels or credits are comfortable.

Convolutional network design looks same like a representation of human brain neuron structure and it is animated by using visual-crust relation. Neuron response to over-hauls in the bounded region of field is known as reception field. Entire visual local is covered. Well images are cross-section of pixels, so we smoothing the images of 3×3 pixels into the 9×1 pixels and it will be feuded to multi-level. In occasions of staggeringly central equal images, the procedure maybe shows specific precision score while figure performance of classes. It will have essentially zero exactness as compared to complex images which are having pixel conditions get evaluated [4].

Convolutional network will get adequate conditions which are spatial and temporal in the image with the help of channels or credits. Planning is playing better fitting in the image database. In that capacity, the association can be set up to fathom the refinement of the image better.

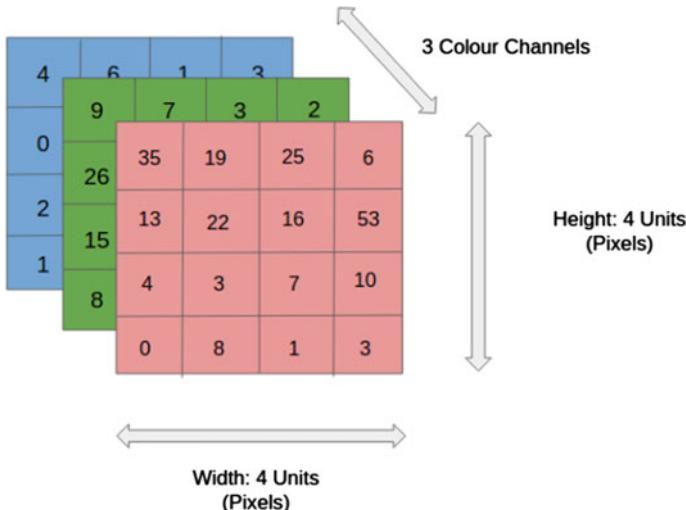


Fig. 1 RGB scale

Fig. 1 depicts that RGB images are disconnected by its three concealing levels that are: Red, Green and Blue. We have different concealing levels in which images exist grayscale, R-G-B, H-S-V, C-M-Y-K, etc. We can imagine how computationally focused things can get once the images show up to estimation, if we say 8 k (7680×4320). The piece of the convolutional network is to diminish the images into construction which are less difficult for quantification, not losing the features those are essential to get a nice scale in. That is critical when we have to design a designing which is not only worthy for understanding features, what's more versatile to huge datasets [5].

Where we have dimension as 5:5:1 it given by green coloring. The initial layer of the layer is addressed by Kernel layer and it's given by yellow coloring. Here P is assumed as a 3:3:1 network. The Kernel moves on various occasions by virtue of Step Length = 1 (Non-Stepping), through the piece of P playing out framework increases the action of floating in P and piece of P. The channel moves to step length of P in restricted way. It processed it with left to till it completes total picture completely. Keeping in memory the RGB coloring format it moves ahead with picture stepping. Product is taken out from the framework till P_n as in $([P_1, Z_1]; [P_2, Z_2]; [P_3, Z_3])$ and every one of the outcomes are added to inclination which provides a crushed single-profundity CNN [6].

The use of the method is removing the verifiable criteria like edges, from the data picture. ConvNets need not be limited to simply a solitary Convolutional Layer. Most probably, low-level details are obligated, and priority given to high-level details gives us similar dataset how it would look like [7].

There are two sorts of results to the movement: one in which the convolved incorporate is decreased in dimensionality when appeared differently in relation to

the data, and in second the dimensionality is enlarged with respect to previously or is unchanged. It would be done by Valid Padding.

3.2 *Image Augmentation*

The presentation of profound learning neural organizations frequently improves with the measure of information accessible. Information increase is a method to falsely make new data is created from existing data. It ends up with applying region specific tactics to algorithm from preparation of data that make new and different group of methodology.

Picture information increase is maybe the most notable sort of information expansion and includes making changed variants of pictures in the preparation dataset that have a place with a similar class as the first picture. Changes incorporate a scope of tasks from the field of picture control, like movements, flips, zooms, and substantially more.

The plan is to extend the preparation dataset with new, conceivable models. This implies, varieties of the preparation set pictures that are probably going to be seen by the model. For instance, a flat flip of an image of a feline may bode well, in light of the fact that the photograph might have been taken from the left or right. A vertical flip of the photograph of a feline does not bode well and would presumably not be proper given that the model is probably not going to see a photograph of a topsy turvy feline [8].

Thusly, unmistakably the decision of the particular information increase procedures utilized for a preparation dataset should be picked cautiously and inside the setting of the preparation dataset and information on the difficult area. Furthermore, it is very well maybe helpful to explore different avenues regarding information increase techniques in disengagement and in show to check whether they bring about a quantifiable improvement to demonstrate execution, maybe with a little model dataset, model, and preparing run.

Present-day profound learning calculations, for example, the convolutional neural organization, or CNN, can learn highlights that are invariant to their area in the picture. By the by, expansion can additionally help in this change and can help the model in learning highlights that are likewise invariant to changes, for example, left-to-option to through and through requesting, light levels in photos, and that is only the tip of the iceberg [9].

Expansion is not used for testing the dataset but only for extending the dataset size for further training purpose. This is not same as data planning, as example, picture resizing should be performed with all the existing datasets.

3.3 Transfer Learning

Here transfer learning is used as move learning and is related to issues, for instance, play out numerous undertakings learning and thought drift and is not exclusively a space of study for the knowledge. Coincidentally, move learning is notable in significant learning given the colossal resources expected to get ready significant learning models or the gigantic and testing datasets on which significant learning models are taken for granted.

Move learning we can say it as move learning potentially works in significant learning if the model features acquired from the essential endeavor are general. This cooperation will overall work if the features are general, which implies sensible to both base and target tasks, as opposed to unequivocal to the base task [10].

This kind of move learning used in significant learning is called inductive trade. This is where the degree of possible models (model tendency) is restricted in a profitable way by using a model fit on a substitute anyway related endeavor.

3.4 K-Fold Cross-Validation

K-fold CV gives a model with less inclination contrasted with different strategies. In K-fold CV, we have parameters ‘X’. This boundary chooses the number of folds the dataset will be partitioned. Each overlap gets opportunity to shows up in the preparation set ($X - 1$) times, which thusly guarantees that each perception in the dataset shows up in the dataset, consequently empowering the model to gain proficiency with the basic information circulation recover.

The estimation of ‘X’ utilized is by and large between 5 or 10. The estimation of ‘X’ ought not to be excessively low or excessively high. In the event that the estimation of ‘X’ is excessively low (say $X = 2$), we will have a profoundly one-sided model. This case is like that of parting the dataset into preparing and approval sets, consequently the predisposition will be high and difference low. In the event that the estimation of ‘X’ is huge (say $X = n$ (the quantity of perceptions)), at that point this methodology is called Leave One Out CV (LOOCV). For this situation, inclination will be low, yet the difference will be high and the model will over-fit, bringing about the model to flop in summing up ridiculous set.

Another methodology is to rearrange the dataset only once before parting the dataset into k folds, and afterward split, to such an extent that the proportion of the perceptions in each class stays as before in each crumple. Furthermore, the test set does not cover between subsequent prominences. This methodology is called Stratified K-fold CV. This methodology is valuable for imbalanced datasets.

4 Result Analysis

4.1 Dataset Description

The problem statement involves using images that are obtained during pathology tests to detect whether the patient is IDC positive or negative. Histopathology images are the images of tissues that are obtained during pathology tests; therefore, these images will act as inputs for the problem [10]. The dataset that contains histopathology images for breast cancer is present on Kaggle at: <https://www.kaggle.com/paultimothymooney/breast-histopathology-images>. The dataset contains 198,738 of negative IDC and 78,786 of positive IDC, subsequent; it is a decent dataset with enough information for our assignment. The first dataset comprised of 162 entire mount slide pictures of breast cancer (BCa) examples examined at $40\times$. In any case, the information that I have chosen contains pictures that are trimmed from the first dataset, i.e., it contains patches of locales where the IDC happens, making it more explicit to our concern. Each fix's document name is of the organization: u_xX_yY_classC.png—>model 10253_idx5_x1351_y1101_class0.png. Where u is the patient ID (10253_idx5), X is the x-organize of where this fix was edited from, Y is the y-arrange of where this fix was trimmed from, and C demonstrates the class where 0 is non-IDC and 1 is IDC. Figure 2 represent Simple CNN model confusion matrix where used model is sequential model. Figure 3 represents transfer learning confusion matrix where the used model is VGG-19 (Figs. 4 and 5).

Fig. 2 Confusion matrix
CNN

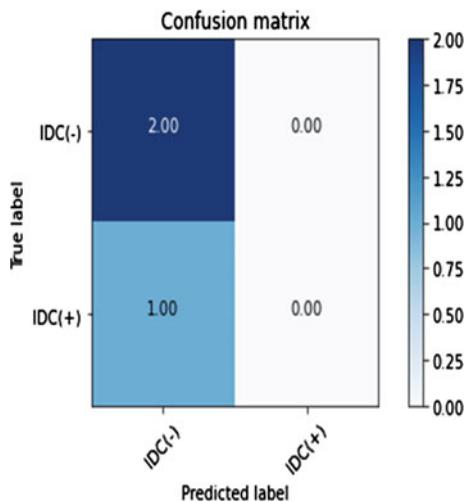


Fig. 3 Confusion matrix for transfer learning

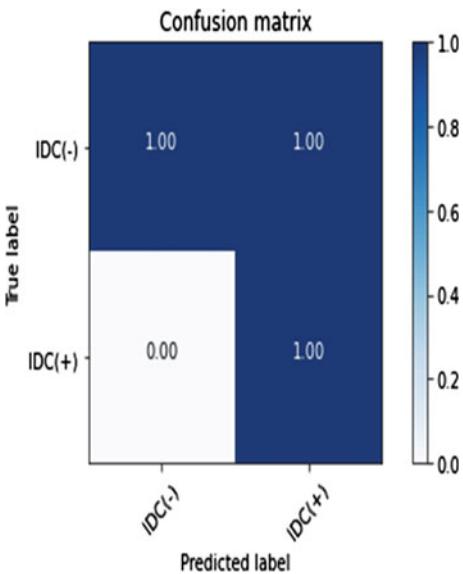
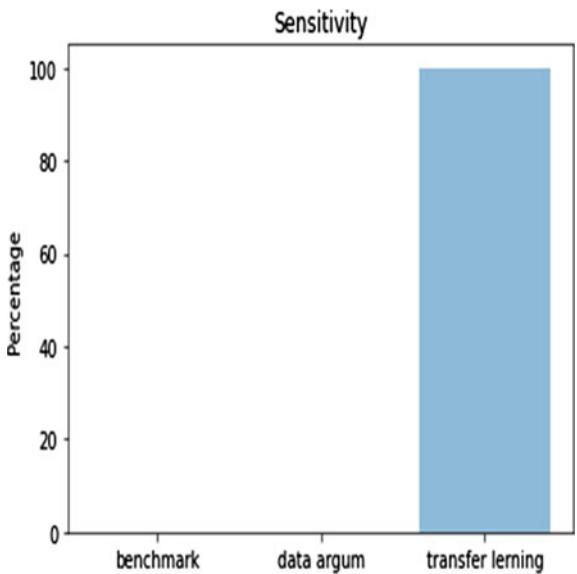


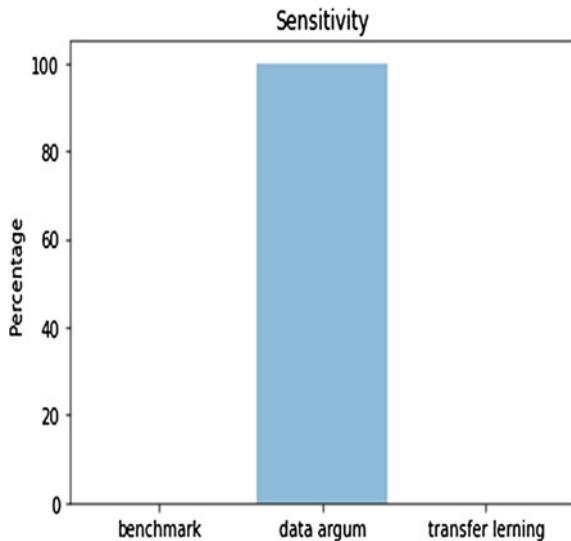
Fig. 4 Sensitivity performance metrics



5 Conclusion

The proposed approach means to mechanize the characterization and recognizable proof cycle of bosom disease in mamogram investigation. The sort of information we are utilizing is having positive for IDC and negative for IDC. We are getting

Fig. 5 Sensitivity for benchmark model



89.99% exactness for negative IDC in future, and we will try to have a completely programmed framework with picture handling and order of the mammogram cases dependent on the methods. It could assist radiologists in IDC understanding cycle as a proper non-intrusive apparatus.

References

1. A novel approach for breast cancer detection and segmentation in a mammogram (2015). *Procedia Comput Sci* 54:676–682
2. Breast cancer detection using image enhancement and segmentation algorithms (2018). *Biomed Res* 29(20)
3. Viswanath H, Guachi-Guachi L, Thirumuruganandham SP (2019) Breast Cancer detection using image processing techniques and classification algorithms
4. Pratiwi M, Jeklin Harefaa A, Nandaa S (2015) Mammograms classification using gray-level co-occurrence, matrix and radial basis function neural network. In: International conference on computer science and computational intelligence (ICCSCI 2015)
5. Devakumari D, Punithavathi V (2018) Study of breast cancer detection methods using image processing with data mining techniques. *Int J Pure Appl Mathe* 118(18):2867–2873
6. Zhou X, Li C, Mamunur Rahaman MD A comprehensive review for breast histopathology image analysis using classical and deep neural networks. <https://doi.org/10.1109/ACCESS.2020.2993788>
7. Sreedevi S, Sherly E (2014) A novel approach for removal of pectoral muscles in digital, mammogram. In: ICICT 2014
8. Guzmán-Cabrera R, Guzmán-Sepúlveda JR, Torres-Cisneros M, May-Arrijoa DA, Ruiz-Pinales J, Ibarra-Manzano OG, Aviña-Cervantes G, González Parada A (2013) Digital image processing technique for breast, cancer detection. *Int J Thermophys*, pp 1519–1531

9. Tomar RS, Singh T, Wadhwani S, Bhadoria SS (2009) Analysis of breast cancer using image processing techniques. In: IEEE 2009, pp 251–256
10. Chaturvedi A, Kumar P, Rawat S (2016) Proposed noval security system based on passive infrared sensor. In: InCITe, IEEE 2016

Pilot Decontamination Using Sector Base Method in Massive MIMO System



Dikshit Kalyal, Paras Chawla, and Rajpreet Singh

Abstract In this research has discussed the pilot contagion termination in the 5G communication known as massive number of multiple input and multi output system. In this Massive MIMO system have issue pilot contagion and to reduce this issue in this research has discussed a proposed solution that used the less number of pilots reused in current cell and every base station in cell is divided into three sectors and in every sector used different pilot sequence to terminate the interference for their users in cell for communication in uplink or downlink, after sectoring of the cell, in every sector users are communicating with their sectored antenna array for all the cells and in channel coefficient of every sector users data are transmitted accordingly time division duplexing mode where channel reciprocity is possible. For approximation of medium coefficients and finding, the transmitted data of different users of different sectors used MMSE in approximate message algorithm.

Keywords Pilot contagion · Massive MIMO · AMP algorithm · Channel approximation

1 Introduction

In the massive mimo system in 5 g communication used huge network system and this system have large capacity of handling users than LTE system basically this system is hundred times faster than LTE and have more spectral efficiency, this system uses huge numbers of antenna on single base station that serves the users in their range [1]. In this system when the cluster of cells uses same number of pilots or other words when these pilots sequences are non-orthogonal to each other, then problem occurs

D. Kalyal (✉) · P. Chawla · R. Singh
Chandigarh University, Gharuan, Mohali, India

P. Chawla
e-mail: hod.ece@cumail.in

R. Singh
e-mail: rajpreet.ece@cumail.in

known as pilot contagion and interference also created between the cells inter cell and intra cell interference, but when numbers of antenna are increased in the base station while using orthogonal pilots then interference is reduced and pilot contagion is also terminated in some extent. It depends upon how much pilot repeated factor, if the pilot repeated factor is small then we have less orthogonal pilot's otherwise large pilots [2].

- The main contribution of this paper is to decontaminate pilots in massive mimo system using orthogonal pilots.
- Reuse less number of pilots in cell for pilot contamination termination and testing on randomly deployed and systematic less number of users.
- For these used sector-based pilot based decontamination method discussed below:

We have discussed pilot contagion problem and use a sector-based algorithm and make a try to use less number of pilots in between the base station and cluster to make every sector of base station orthogonal. In this method we have divided into two phases, in phase I, we have divided cell into sectors and apply on them orthogonal pilots and in phase II, we have applied Amp algorithm for approximated the transmitted signal, in AMP only need to approximate the channel and transmitted signal [3].

In every sector, we use orthogonal pilots to make sectors orthogonal from other neighbour sectors of other cells as well as of same cell. After that, we have used the approximate message passing algorithm on the channel coefficient of the sector-based method. Find out the parameters' values that will be shown in graph below of every sector user in figure shown below pilot contagion problem due to usage of same pilots in both cells and signal of cell1 user and cell is received by cell2 of user in uplink and downlink [4].

1.1 Pilot contamination

When the user transmits their data and pilots to the base station then the base station in single antenna received lot of unknown users pilot signals. If the received pilot is same as the neighbour base station user then pilot contagion problem occurs or if the neighbour user in same cell uses the same pilot then also pilot contagion or contamination (Fig. 1).

Occurs between users. As an example, if received pilot signal is $R_p \in C^M$.

$$R_p = \sum_{i=1}^L \sum_{k=1}^K H_{ijk} X_{ik} + N_j \quad (1)$$

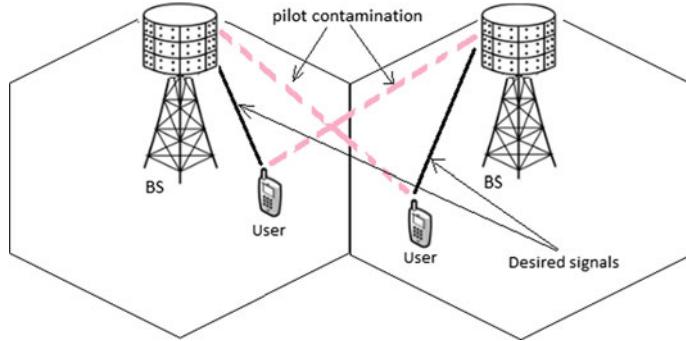


Fig. 1 Massive mimo system for $L = 2$ and $K = 1$ in each [3]

where H_{ijk} is the channel with i th cell j th antennas and k th users X_{ik} is the transmitted signal from k th users in i th cell and N_j is the additive white Gaussian noise is added to the channel which has zero mean and sigma square variance $(0, \sigma^2 I_m)$.

$\rho_{ik} = E(|X_{ik}|)^2$ Is the power of signal? τ_p Is the sample of each coherence block pilot's symbol [5].

The pilots $\emptyset_{ik} \in C^{\tau_p}$. $\emptyset_{ik}^H \emptyset_{ik} = \tau_p$ In addition, $\sqrt{\rho_{ik}}$ is the signal power?

$$R_p = \sum_{k=1}^K \sqrt{\rho_{ik}} H_{jk} \emptyset_{jk} + \sum_{\substack{i=1 \\ i \neq j}}^L \sqrt{\rho_{ij}} H_{jil} \emptyset_{jl} + N_j. \quad (2)$$

In this equation, first term is desired pilot, 2nd term is inter-cell pilot and 3rd term is noise.

For estimating the channel, we need the pilots and the approximated channel equation is given below

$$\begin{aligned} H^{\text{est}} = & \sum_{k=1}^K \sqrt{\rho_{ik}} H_{jk} \emptyset_{jk} \emptyset_{jk}^* + \sum_{\substack{i=1 \\ i \neq j}}^L \sqrt{\rho_{ij}} H_{jil} \emptyset_{jl} \emptyset_{jk}^* \\ & + \sum_{l=1}^L \sum_{\substack{k=1 \\ l \neq j}}^K \sqrt{\rho_{il}} H_{jil} \emptyset_{jl} \emptyset_{jk}^* + N_j \emptyset_{jk}^* \end{aligned} \quad (3)$$

In this H approximation equation first term is the desired pilot term and 2nd term is intra-cell pilots, 3rd term is inter-cell pilots and 4th term is additive white Gaussian

noise is added in channel and $\llbracket \emptyset_j \rrbracket$ k^* is the multiplied orthogonal pilot with received pilots signal to approximate the channel [6].

2 Literature Review

Various authors proposed different methods for pilot decontamination in channel estimation in massive mimo system. In [7, 8] discussed channel estimation techniques and some methods of estimation in uplink and downlink and how much number of pilots are used for estimation. In [9] told regarding pilot contamination problem and channel reciprocity, [10] told about signal detection in uplink, pilot contamination and some methods. In [11] discussed FDD, orthogonal pilots and user scheduling in massive mimo, [12] discuss pilot based sub-spaced approach for de contagion, [13] give idea of ZF and MMSE algorithms for estimation and user division into edge and centre, [6] shows SPRS scheme and weighted graph scheme used for termination of pilot contamination. In [14, 15] Amp algorithm for data detection from channel MMSE estimator for channel estimation [16].

3 System Model

In this model, we have discussed K number of users in the cell M numbers of antenna in cell and L are the number of base stations in cell and one base station in every cell, every base station is divided into three sectors. Sector1, Sector2 and Sector3 and in sector1 have m_1 numbers of antenna and k_1 number of users, similarly for sector2 m_2 numbers of antenna and k_2 number of users and in sector3, m_3 numbers of antenna and k_3 number of users in one cell, so the total numbers of antenna in one base station are the sum of all sector antennas in current base station and for users sum of all users in all sector of current cell.

$$\text{Sector1} = m_1, k_1$$

$$\text{Sector2} = .m_2, K_2$$

$$\text{Sector3} = m_3, k_3$$

$$\text{Base station} = \text{sector1} + \text{sector2} + \text{sector3}$$

$$\text{Number of } M = m_1 + m_2 + m_3$$

$$\text{Number of } K = k_1 + K_2 + k_3$$

H is the channel coefficient for i -th base station j -th antennas and k -th users in cell

S denoted the number of pilots, Φ pilot's $1 \leq i \leq S$ and pilots $(1 \leq S \leq K)$. We randomly assign pilots Φ_{plk} to user (l, k)

Channel coefficient \mathbf{H} for $(l, k, m) = \mathbf{H}(l_1, k_1, m_1) \dots l_L, k_K, m_M$ for every sector in cell.

$$H = \text{large-scale fading coefficient} * \text{small-scale fading coefficient.}$$

$$H = \sqrt{(\beta l, k, m)} * G(l, k, m) \quad (4)$$

$\sqrt{(\beta l, k, m)}$ is the large-scale fading coefficient and $G(l, k, m)$ is the small-scale fading coefficient for l cells k users and m numbers of antenna in each cell. H is $M \times K$ matrix for small-scale component and G has path-loss and shadowing effect in large-scale fading coefficient in channel.

Channel coefficient for sector1

For $i = 1: L, j = 1:m1, k = 1:k1$

$H1 = H$ matrix ($m1, k1$)

Channel coefficient for sector2

For $i = 1: L, j = 1:m2, k = 1:K2$

$H2 = H$ matrix ($m2, K2$)

Channel coefficient for sector3

For $i = 1: L, j = 1:m3, k = 1:k3$

$H3 = H$ matrix ($m3, k3$).

$H(1, 2, 3)$ are the channel coefficient matrices of sectors in cell with L base station and m, k antenna and users?

3.1 Model

Initialize $L, (K, M) k1, k2, k3, m1, m2, m3;$

For $k1, m1$ and $k2, m2$ and $k3, m3$;

% Divide sector wise base station;

$\Theta = 0 < 2\pi/3 < 4\pi/3 < 2\pi$;

If theta in between $0 < 2\pi/3$;

Then sector1

Create channel for sector 1

Else if theta in between $2\pi/3 < 4\pi/3$;

Then sector 2

Create channel for sector 2

Else if theta in between $4\pi/3 < 2\pi$;

Then sector3

Create channel for sector 3;

Create data

Apply AMP algorithm on channels and estimate data using AMP, End, End.

4 Methodology

4.1 Amp Algorithm

In Approximate message passing algorithm, we approximate the message from channel and find SNR and SER symbol error rates by using monte-carlo simulations [1]. We have proposed to combine LMMSE with Massive MIMO architecture is to serve tens of users by employing hundreds of antennas and received equation is

$$Y = Hx + w$$

where the channel coefficient $H \in Cm \times k$ users for every sector has its elements sampled from $NC(0,1/m)$, $m \gg k$, $y \in Cm$ is the received signal, AWGN noise components w are i.i.d with $N \in (0, \sigma^2)$; regarding the transmitted signal is x , we only presume that it's zero mean and finite variance σ^2 [14].

Before including the AMP algorithm, we should be well aware of two facts:

1. MMSE approximation probability of $(x|y)$ to work with the exact earlier degrage the necessity of employing AMP, because for achieving the full diversity essentials anlarge set of constellation points, in which Approximate Message Passing algorithm works slowly when doing the moment of the matching process, it's not to mention problems about its impotence to converge to the lowest fixed point [15].
2. In the CDMA multiuser determination theory, MMSE determination does not mean the one working with correct earlier knowledge, but exclude the one assuming a Gaussian earlier. Therefore, use a proxy earlier for determining the transmitted x , that is assuming that $xi \sim N \in (0, \sigma^2)$ mean 0 and variance is σ^2 , the signal power is $\sigma^2 = 2$ taken for QPSK and $\sigma^2 = 10$ taken for the 16QAM.

$$Y = Hx + wi$$

$$Wi = y - Hx$$

The target function is the minimum and normalized square of wi while $xi \sim N(0, \sigma^2)$ x is the transmitted signal rt is the residual factor, k is the number of users m is the number of antenna at denotes the number of iterations for $k = 1, \dots, N$ and also for $x = 1, \dots, X$, αt is the damping factor with t iterations [16].

$$\text{Min} \|y - Hx\|^2, xi \sim NC(0, \sigma^2)$$

$$Rt = y - Hxt + k/m * \sigma^2 / (\sigma^2 + \alpha t - 1) * rt - 1 \quad (5)$$

$$At = \sigma^2 + n/m * \alpha t - 1\sigma s^2 / (\sigma s^2 + \alpha t - 1) \quad (6)$$

Initialize is $r0 = 0$, $x0 = 0$, $\alpha 0 = \sigma s^2$. In terms of complexity, it only costs 2 mn. Also, according to the second equation of the algorithm, it is converging extremely fast.

4.2 Proposed Solution

See Figs. 2 and 3.

In the proposed method, we have discussed sectorized based three pilots reused in single cell of base station. In every sector of pilots are orthogonal to each other, pilots reused in cell are p1, p2 and p3. For sectoring parameter used theta for separation of the sectors in cell, theta varies from 0 to $2 * \pi/3$ allotted for P1, $2 * \pi/3$ to $4 * \pi/3$ allotted for p2 and for $4 * \pi/3$ to $2 * \pi$ allotted for p3.

P1 is first pilot sequence $1 < S < k1$.

P2 is 2nd pilot sequence $1 < S < k2$.

P3 is 3rd pilot sequence $1 < S < k3$.

Sector1 = P1, sector2 = P2, sectpr3 = P3.

Total number of users are $K = k1 + k2 + k3$ and M is the total numbers of antenna used in every massive mimo cell. For estimation of channel coefficient of users of all the three sectors, first pilots of the channel are received at the receiver and multiply with receiver pilots if same cancel out and then get that particular channel

Fig. 2 Pilot reuse is 3 sector wise in cell

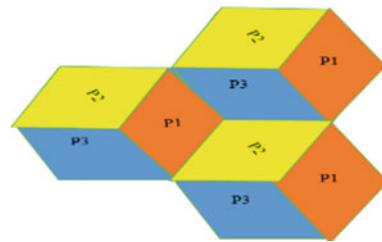
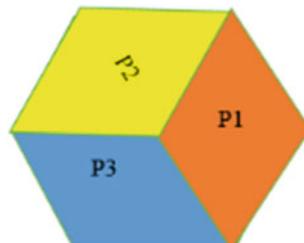


Fig. 3 Pilot reuse in single cell sector wise



that uses TDD mode for channel estimation. When channel recognized for all users then in downlink it used reciprocity system for transmission of data in channel for communication.

5 Results and Findings

In this simulation in all the three sectors of cell of users are, distributed in random in all three sectors and also simulate without random users deployment in the all three sectors of cell and measures the snr and bit error rate.

For sector1 users are deployed randomly up to 15 and without randomly deployed 15 users in sector1 and numbers of antenna used 32, results shown below in graph, for sector2 users randomly distributed up to 18 and without randomly also 18 for viewing results and numbers of antenna used also 32 and for sector3 the users are deployed randomly 25 and also without randomly are 25 and the number of antenna is used 64 in sector3 of the base station the SNR range is in between 1 and 30, red and green graph is showing the AMP algorithm used received variance power and blue shows MMSE estimator in the algorithm. If more numbers of antenna in any sector and users are less then data rates are very good in that particular sector of the base station, data rates are only affected while fast moving subscribers because if any sector has antennas on base station are equal to users than for rural users also have achieved less number of data rates and snr is also low at that time, in urban areas where population is more where deployment of antennas more in that sector of cell. For improving coverage and satisfying that all users in that area. Capacity of system is directly proportional to snr. If symbol error rate is, increase the value of snr fall down (Figs. 4, 5, 6, 7, 8 and 9).

Fig. 4 Sector1 randomly deployed users 1–15, 32 M

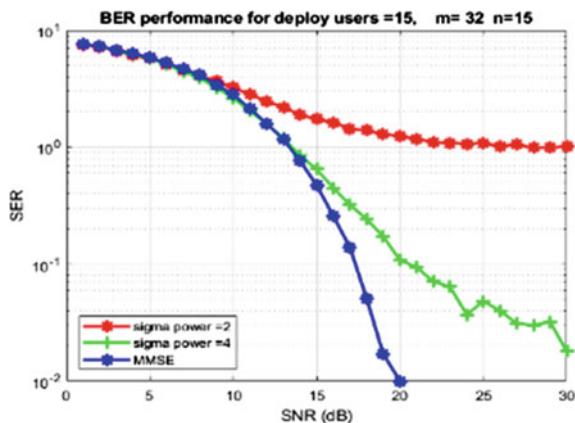


Fig. 5 Sector1 deployed users 15 and 32 M

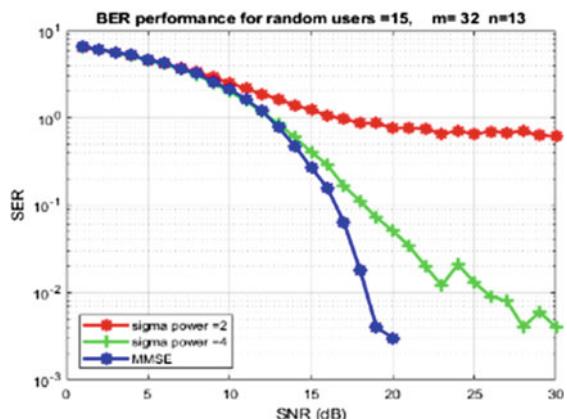


Fig. 6 Sector2 randomly deployed users 1–18, 32 M

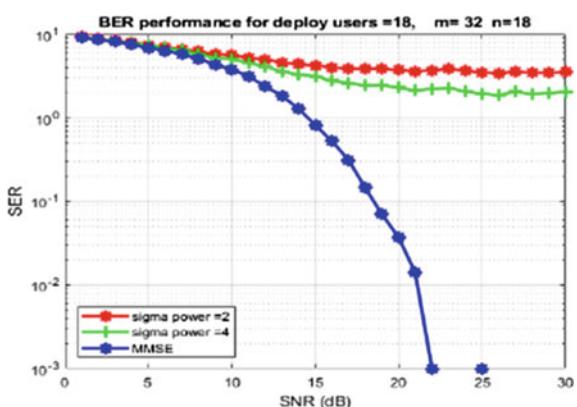


Fig. 7 Sector2 deployed users 18 and 32 M

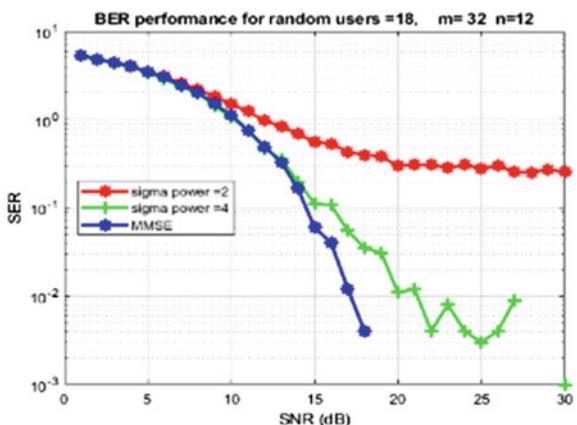


Fig. 8 Sector3 random users 1–25 and 64 M

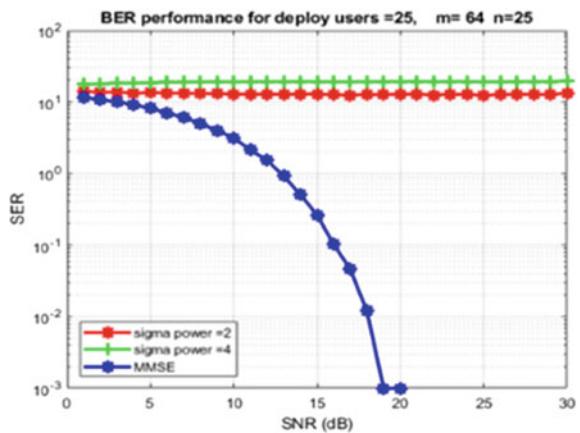
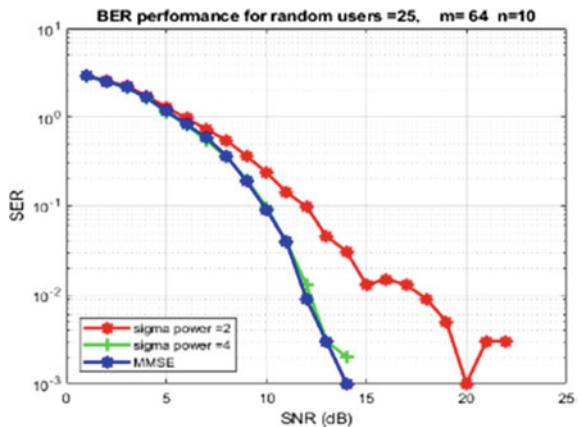


Fig. 9 Sector3 deployed users 25 and 64 M



5.1 Table

See Table 1.

Table 1 Parameters used in this proposed solution

Parameters	Values
Radius of cell	0–500 m
Total number of cells	3
Theta	0–2 * pi
K = k1 + k2 + k3	58
M = m2 + m2 + m3	128
Pilots	3

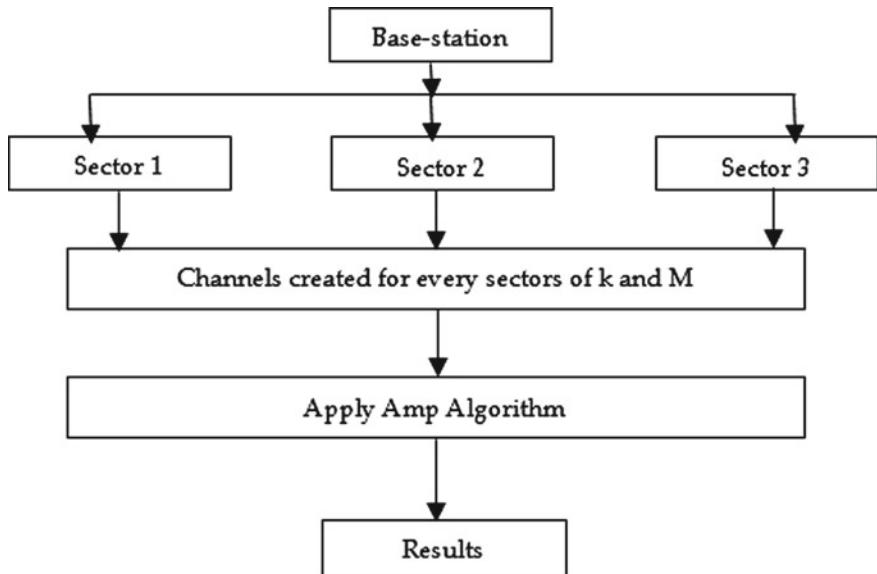


Fig. 10 Flow graph of paper

5.2 Flow Graph

See Fig. 10.

6 Discussion

In the section, we have discussed findings of this article we have discussed pilot decontamination problem. We have separates the base station into three sectors and apply different pilots in every sector of base station for termination of pilot contagion problem. In the experimental way, we deploy random users for all three sectors in cell and systematic user deployment also for observing parameters in results. Parameters value varies while systematic to randomly user's deployment observing in cell. In comparison of snr regarding random users and systematic users in cell. Systematic subscribers have good snr but in real time all the users are not in systematic during days mostly, at night most of the users are in systematic form and achieve good data rates but snr of random users is low in comparison with systematic users. Theoretically, parameters values of k or users is equal or mostly prefer less than the antennas used in base station. In case of urban areas movement of users is high than rural areas and antennas on base station are mounted in stationary, so if antennas are less than moving users faces difficulty to access desirable data rates.

7 Conclusion

In this research, discussed how can reuse less number orthogonal pilots in the cell and reduce pilot contagion problems from communication systems. For that, we have used approximate message passing algorithm for determination of data from random channels in uplink or downlink, in the proposed algorithm divide the cell into three sectors and in every sector, we have applied different pilots and then approximate the data from all three sector channels and for getting the results use MATLAB 2020b software. In the deployment of users in sectors randomly, in every sector accordingly, user snr graph and BER are changing due to changing number of users in cell.

Acknowledgements I would like to express my deep and sincere gratitude to my supervisor, Dr. Paras Chawla is professor and head of department of electronics and communication engineering and Mr. Rajpreet Singh is assistant professor and co-supervisor of department of electronics and communication engineering in Chandigarh University for giving me opportunity and providing me invaluable guidance for my research.

References

1. Zhou Y (2020) A novel way for pilot decontamination in massive MIMO multi-cell systems. *The Front Soc, Sci Technol* 2(12):104–115. <https://doi.org/10.25236/FSST.2020.021216>
2. Kalantarinejad N, Abbasi-Moghadam D (2020) Joint distance-based user grouping and pilot assignment schemes for pilot decontamination in massive MIMO systems. *Int J Commun Syst* 33(3). <https://doi.org/10.1002/dac.4216>
3. Belhabib A, Boulouird M, Hassani MMR, Nash equilibrium based pilot decontamination for multi-cell massive MIMO systems
4. Liu L, Yu W (2017) Massive device connectivity with massive MIMO. In: 2017 IEEE International Symposium on Information Theory (ISIT). IEEE, pp 1072–1076. <https://doi.org/10.1109/ISIT.2017.8006693>
5. Björnson E, Hoydis J, Sanguinetti L (2017) Massive MIMO networks: spectral, energy, and hardware efficiency. *Found Trends Signal Process* 11(3–4):154–655. <https://doi.org/10.1561/2000000093>
6. Memon SA, Chen Z, Yin F (2016) Pilot decontamination in multi-cell massive MIMO systems. In: Proceedings of the 2nd international conference on communication and information processing, pp 227–232. <https://doi.org/10.1145/3018009.3018013>
7. Khan I, Zafar MH, Jan MT, Lloret J, Basher M, Singh D (2018) Spectral and energy efficient low-overhead uplink and downlink channel estimation for 5G massive MIMO systems. *Entropy* 20(2):92. <https://doi.org/10.3390/e20020092>
8. Albatineh Z, Hayajneh K, Salameh H. B., Dang, C., & Dagmeh, A. (2020). Robust massive MIMO channel estimation for 5G networks using compressive sensing technique. *AEU-Int J Electron Commun* 120:153197. <https://doi.org/10.1016/j.aeue.2020.153197>
9. de Figueiredo FA, Cardoso FA, Moerman I, Fraidenraich G (2018) Channel estimation for massive MIMO TDD systems assuming pilot contamination and flat fading. *EURASIP J Wirel Commun Netw* 2018(1):1–10. <https://doi.org/10.1186/s13638-018-1021-9>
10. Chataut R, Akl R (2020) Massive MIMO systems for 5G and beyond networks—overview, recent trends, challenges, and future research direction. *Sensors* 20(10):2753. <https://doi.org/10.3390/s20102753>

11. Khan I, Rodrigues JJ, Al-Muhtadi J, Khattak MI, Khan Y, Altaf F, Mirjavadi SS, Choi BJ (2019) A robust channel estimation scheme for 5G massive MIMO systems. *Wirel Commun Mob Comput.* <https://doi.org/10.1155/2019/3469413>
12. Banoori F, Shi J, Khan K, Han R, Irfan M (2021) Pilot contamination mitigation under smart pilot allocation strategies within massive MIMO-5G system. *Phys Commun* 47:101344. <https://doi.org/10.1016/j.phycom.2021.101344>
13. Salh A, Audah L, Shah NSM, Hamzah SA (2020) Mitigating pilot contamination for channel estimation in multi-cell massive MIMO systems. *Wirel Pers Commun* 112:1643–1658. <https://doi.org/10.1007/s11277-020-07120-9>
14. Cakmak B, Winther O, Fleury BH (2014) S-AMP: approximate message passing for general matrix ensembles. In: 2014 IEEE information theory workshop (ITW 2014). IEEE, pp 192–196. <https://doi.org/10.1109/ITW.2014.6970819>
15. Khumalo M, Shi WT, Wen CK (2016) Fixed-point implementation of approximate message passing (AMP) algorithm in massive MIMO systems. *Dig Commun Netw* 2(4):218–224. <https://doi.org/10.1016/j.dcan.2016.08.002>
16. Zhao J, Ni S, Gong Y, Zhang Q (2019) Pilot contamination reduction in TDD-based massive MIMO systems. *IET Commun* 13(10):1425–1432. <https://doi.org/10.1049/iet-com.2018.5557>

Applications of Deep Learning in Diabetic Retinopathy Detection and Classification: A Critical Review



Preeti Kapoor and Shaveta Arora

Abstract Diabetes is the world's fastest-growing illness, causing a slew of complications. One of these conditions is diabetic retinopathy (DR) which causes retinal lesions affecting the vision. There are several levels of DR, ranging from moderate to extreme. Early detection and treatment of DR can decrease the risk of loss of vision. Presently, DR detection is a laborious and blue-collar process that needs qualified ophthalmologists to inspect digital color retinal fundus photographs. As a result, various machine vision-based methods for automatically diagnosing diseases are explored in the literature. Deep learning has emerged in recent years achieving better performance than existing methods in many areas, especially in analyzing and examining the medical images. Deep CNNs are broadly used method in analysis of medical images. To highlight the role of CNNs in DR detection and classification, various recent papers have been studied and considered. The characteristics of various available DR Datasets of colored fundus images are also discussed. The paper suggests a set of steps that will form a process such that it will be promising at all the stages of DR detection/ classification. The explanation behind the suggested set is derived by comparing the results of the papers. Since the results of any method depend on the input, therefore, to see the effects of selected preprocessing method, i.e., CLAHE, was applied to images from most widely used dataset: Kaggle dataset. The processed images were noise-free also the lesions were clearly visible.

Keywords Diabetic retinopathy · Deep learning · Transfer learning · Ensemble · Microaneurysms (MA) · Haemorrhages (HM) · Soft and hard exudates (EX) · Cotton wool spots · Neovascularization (NV) and macular edema (ME) · CLAHE

P. Kapoor (✉) · S. Arora
North Cap University, Gurugram, India
e-mail: preeti19csd006@ncuindia.edu

S. Arora
e-mail: shavetaarora@ncuindia.edu

1 Introduction

With the modern lifestyle, diabetes has become a common disease. Diabetic Retinopathy (DR) is an eye affecting disease that leads to swelling of retinal blood vessels and leakage of fluids and blood [1]. Diabetic retinopathy (DR) is a common cause of human visualization loss. An increase in number of diabetic patient's percentage from 2.8% in 2000 to 4.4% in 2030 worldwide is expected [2]. Patients have more probabilities of DR who suffer from diabetes for a long period. To avoid the risk of blindness regular eye screening is important for diabetes patients at an early stage [3]. The presence of diverse types of lesions on a retina marks the possibility of DR. These lesions are Mas, HM, SE, HE, cotton wool spots, NV, and ME. The Vision loss can be sudden or gradual like Cataract, Tractional R.D and D. R is the cause of Gradual Vision loss whereas sudden vision loss is caused by PDR or NPDR. The underlying issue with DR [4] is that it becomes incurable as it progresses, so early detection is critical. This has prompted the creation of automatic diagnostic systems to aid in the early detection of DR. Even though several attempts have been made in this direction but rating diabetic retinopathy (DR) is critical for deciding appropriate care as well as the patient follow-up. Diabetic patients can become blind as a result of DR. If observed and treated in the first phase, its impact can be reduced. For vision restoration and timely treatment, it is important to detect DR early. Automatic DR Recognition is cost-effective and saves time. Automated techniques are more competent than a labor-intensive judgment since it is prone to possibilities of miss-judgment and requires more efforts. Fundus images are used to find the precise position of lesions such as MAs, HM, SE, and HE but since photographs include other features, like the red dots as well as blood vessels, this is a difficult challenge. Preprocessing, attribute extraction/selection, choosing an appropriate classification system, and eventually assessing the findings are all part of the general mechanism for identification, segmentation, and classification.

Deep learning algorithms have demonstrated excellent success in numerous computer vision tasks and have decisively defeated conventional hand-engineered-based approaches in current years, thanks to the accessibility of large datasets to the enormous computational power provided by GPUs. Several deep learning (DL)-based techniques for analyzing retinal fundus imageries have also been developed to build automatic computer-aided diagnostic schemes for DR. Deep learning uses deep NNs to learn different tasks. CNNs and modified versions such as ensemble of CNN, blend of machine learning methods and CNNs, fully connected CNNs, and auto encoders are widely used deep learning models for automatically detecting and grading DR.

This paper reviews the techniques used for preprocessing of images, detection or classification of DR and standard parameters to evaluate the algorithms used. The present effort thoroughly reviewed 30 papers where DL algorithms are used to classify DR images. The paper also tries to find the best techniques among all and to justify the results of all the papers that were compared on the selected methods. The objective is to find a process which will be promising at all the stages of DR

detection/ classification. The paper is planned as follows: Sect. 1 briefly introduces about DR, DR detection /classification, and Deep learning methods. Section 2 gives a summary of the studied papers. Then Sect. 3 discusses the various methodologies, i.e., datasets, preprocessing techniques, training methods, classification types, and performance evaluation. Section 4 is the experimentation and discussion section which tries to justify the widely used techniques of the process. Section 5 tries to discuss the challenges which are faced during DR detection/classification. Finally, Sect. 6 concludes the findings and gives the future scope relevance of the work done.

2 Diabetic Retinopathy and Deep Learning

Diabetes is a disease that increases the amount of glucose in the blood caused by a lack of insulin. The retina, heart, nerves, and kidneys are the most affected areas. The effect of diabetes on eyes is termed as Diabetic Retinopathy causing the swelling of eye vessels and fluid leakage in eyes. If not treated or detected on time DR can lead to vision loss. Diabetic Retinopathy is of two types. In Type 1 which is also called insulin-dependent, DR is more common whereas in second type in there is no need for insulin. Worldwide people suffer from type II DR. Also, DR is more common in Females due to risk factors such as Pregnancy, Hypertension, and anemia. The Early treatment diabetic retinopathy study shows there are 4 stages in DR. Stage I is the Earliest Stage also known as Non-Proliferate D.R which occurs on the inner layers. The presence of lesions if observed on time can save many lives. As seen in Fig. 1, the disorder progresses from moderate NPDR to PDR.

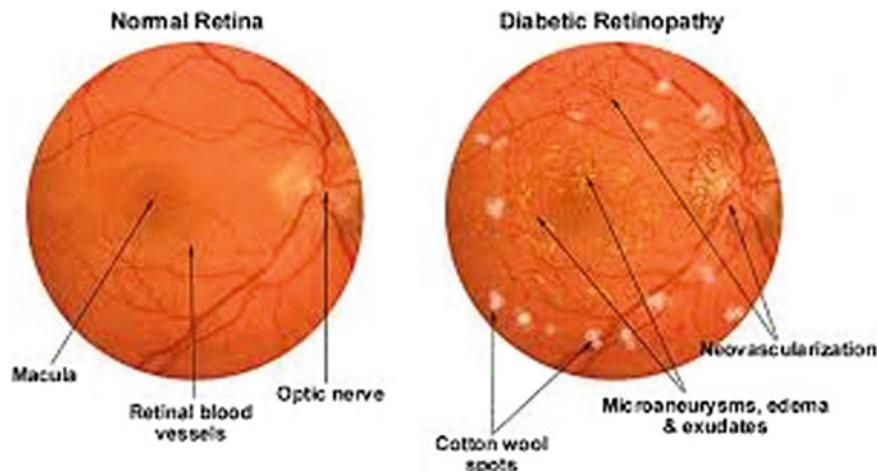


Fig. 1 Diabetic retinopathy (DR)

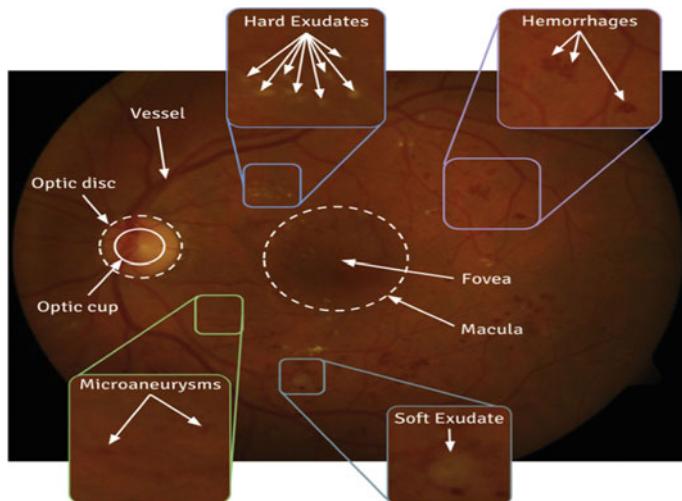


Fig. 2 Fundus photograph showing the signs of Stage I DR [2]

These lesions are:

- Microaneurysms: MAs are the earliest signs in which tiny little specks (pin-point hemorrhages appear). It is the Hallmark of N.P D.R.
- Deep Hemorrhages: red dots-blots, flame type it's a larger version.
- Hard Exudates: HE is the swollen retina, very yellow in color, distinct rings or clump shape mostly around microaneurysms caused by blood vessels leaks.
- Soft Exudates: SE occurs in outer surfaces only and part of hypertension diabetes.

Intra retinal microvascular abnormalities: IMRA beads or loops in veins are the venous changes and retinal edema which is due to dilated capillaries, fluid accumulates in inner retinal layers, cysts appear leading to retinal thickening. Figure 2 shows the earliest signs/ lesions of DR the detection.

For detection of the stage I SEVERE Nonproliferative DR: 4–2–1 Rule is followed in which the eye into 4 quadrants then.

- 4 quadrants have microaneurysms
- 2 quadrants have venous changes
- At least 1 quadrant has IRMA

The second stage is also called Proliferative DR. In this stage, neovascularization is the hall marks. There is a growth of abnormal blood vessels over optic disk and other areas plus NPDR is also there. In elderly people, the problem is known as Recurrent Vitreous where there is N.V D (optical disk) and NVE (elsewhere i.e. majorly on blood vessels) is there at this stage. Third stage, i.e., Diabetic Maculopathy is the main reason for Vision loss. It occurs post-NPDR or PDR It has macular edema. It significantly occurs at

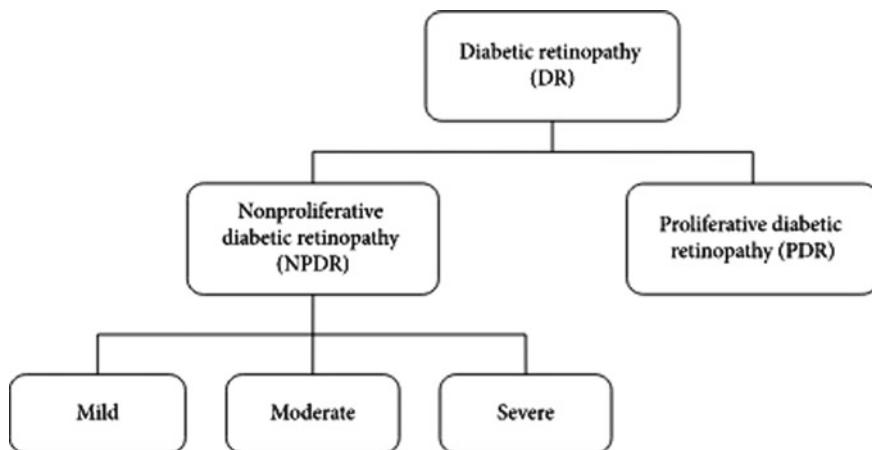


Fig. 3 Stages of DR

- at Macula (central and functional area of retina)
- 500 Um from center of Macula
- increased retinal thickness

End stage of Diabetic Eye Disease is an intricate stage which starts with Tractional R.D: old, dense, and vitreous HMs. Scar tissue or other tissue forms on the retina, pulling it away from the layer underlying, causing tractional retinal detachment. It has the potential to result in significant visual loss. People with diabetes who have severe diabetic retinopathy, or damage to the blood vessels in the retina, are more likely to develop this type. NV as well. Glaucoma is a type of eye disease that affects people of all Glaucoma is a category of eye diseases that affect the optic nerve, which has a negative impact on the health of the eyes. Figure 3 shows the stages of DR.

For feature learning and classification, DL incorporates hierarchical layers of non-linear phases. DL is a computer application for analyzing medical images. It's employed in a variety of applications, including image classification, segmentation, detection, retrieval, and registration. One of the DL algorithms is CNNs. CNNs are the non-recurrent feed-forward technique. Nowadays when CNNs are used in medical image classification provide the best results. LeNet-5, VGG16, VGG 19, and Alex Net are the traditional CNN architectures. Some of the modified versions of Deep CNNs are Inception V3, ResNet, U-net, Dense Net, Squeeze Net, and Efficient Net. Every convolution architecture is distributed into feature extraction and classification. Convolution is a mathematical operation where sum of products is performed. The successive convolutional layers are subsets of previous layers interwoven with average pooling or maximum pooling (downsampling) layers as depicted in Fig. 4. Feature maps in CNNs are the 3D matrix. Various transformation filters and pooling layers are applied on these feature maps to extract robust features.

Table 1 discusses the various CNNs algorithms which are used in the studied papers.

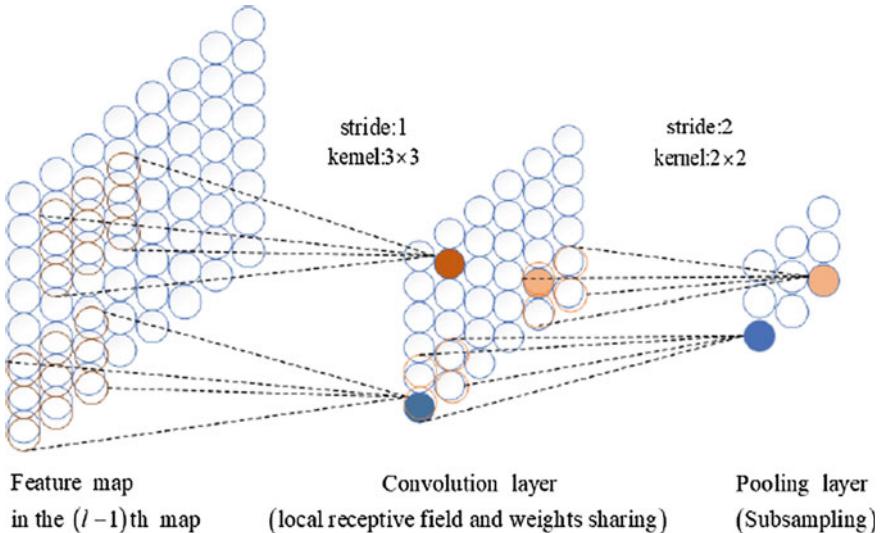


Fig. 4 An illustration of layers of Convolutional Neural Network architecture [5]

The goal of this work is to look at a few current publications in the field of DR detection and classification using DL. The outcome is expected to direct in the direction of finding a process which will improve the results as compared to existing works. The focus is to compare the studied papers on a common dataset which is most widely used. The outcome of which will be the set of methods, used at various stages of DR detection such as preprocessing method, learning technique, classification type, and performance measures for model evaluation. The methods which are chosen are compared and the results obtained in the studied papers are discussed and conclusions are drawn. Also, a sample dataset was selected from the Kaggle dataset and the images were preprocessed using HE and CLAHE for contrast enhancement and the results are shown in the paper. The paper is a critical review for doing future research in the resulted process and to cover the research gaps which are found in the survey.

3 Related Work

The role of DL is seen in various computer vision fields. Also in DR detection, the Deep NNs give better results than the existing machine learning algorithms. In [10] Meher Madhur Dharmana and Aiswarya MS, used Blob detection for feature extraction. The blob detection was done to extract the intensity, number, and radius of Blobs. These details were then fed to Naïve Bayes classifiers, ANN and SVM giving an accuracy of 83% with Naïve Bayes. Whereas in [11], Ganjar Alfian et al., proposed a deep model which in comparison with KNN, NB, SVM, DT, RF attained

Table 1 Description of various Convolutional Neural Network architectures

Paper	Architectures	Description
Mobeen-ur-Rehman et al. [6]	AlexNet	It is an introductory CNN model that uses the ImageNet dataset to train for item classification and identification
Mobeen-ur-Rehman et al. [6]	VGG16	It's a deep CNN model that employs a series of convolutional layers with small 3×3 kernels (or filters), the smallest kernel for extracting data. The use of a combination of these small filters can extract the same information as AlexNets huge filters, but with more efficiency and a less number of parameters to estimate during training
Asra Momeni Pour et al. [7]	EfficientNet	In comparison to the most popular CNNs, EfficientNets are the most efficient. The number of parameters and FLOPS are being reduced, while accuracy and speed are being improved
Alan Lands [8]	ResNet	To avoid the Vanishing gradient problem, it adds a jump link that bypasses the non-linear transform. The flow of the gradient is directed to the next stratum
Alan Lands et al. [8]	DenseNet	DenseNet contains direct connections from every layer, resulting in a more stable architecture than ResNet
Yunlei Sun [9]	LeNet	Earliest model is used for small databases and 1 D data with few layers used for identification of handwritten numbers
Xianglong Zeng et al. [4]	Inception V3	The deep CNN model Inception V3 is well-known. Inception V3 may be tailored to do many image classification tasks by using the transfer learning method

the highest accuracy. Also, the model was tested on other datasets to cross-validate again outperforming the other algorithms with an accuracy of 84%. They analyzed whether the patient will develop DR or not. In the algorithm SVM based REF (i.e., linear function) was used to get the top-ranked features. The grid search algorithm was then utilized to tune the hyperparameters.

Nowadays, Deep CNNs and modified versions of CNNs are most widely used in DR detection as they outperformed the existing methods. In [9], Yunlei Sun, the author applied LeNet to 1-D data of unrelated data. The data was converted to 1-D using various normalization methods. The model added batch Normalization to LeNet: BNLeNet, along with Adam algorithm for parameter optimization. In [12], R. Murugan et al., used the Deep CNN network giving 98% accuracy results in comparison with existing methods. Also the run time of 3.9 s (CPU) which is the least as compared to the existing state-art-methods. Also In [13], Harry Pratt et al., this papers used the deep CNN model with some predefined hyperparametres and focused on real-time batch normalization by assigning weights on the source

of amount of images from a class. In [14], D. Jude Hemanth et al., segmented the images into R, G, B components. Then the processed images were fed into deep CNN for classification into four classes. Similarly In [15], K. Shankar et al., the images were segmented using Histogram method. Then, processed images were fed into an Inception V4 which was hyper tuned using Bayesian optimization and had MLP at the classification layer. The model showed an extraordinary excellency as compared to existing CNN architectures. In [6], Mobeen-ur-Rehman et al., tried to overcome the overfitting problem of the existing variations of CNN by proposing a method which was designed from scratch, a five-layered approach where last three layers were taken to be FC neural networks and starting two layers were convolution layers. The last three layers act as ANN classifier. In [16], P. Saranya and S. Prabakaran, made an effort to improve the efficiency of Deep CNN by performing up sampling and downsampling of the images, along with optic disk segmentation. Also, Canny edge detection was used for the preprocessing. Hyperparameter tuning was done for better classification.

Also, the modified version of Deep CNNs is more explored for getting promising results in DR Detection. In [17], K. Shankar et al., used PCA for segregating the optic disc from images increasing the accuracy by 25–40%. Then histogram-based division was done to separate green from RGB for vessel segmentation. Then Synergic DL was used for classification. In [18], Md. Sanaullah Chowdhury et al., the paper focused on image preprocessing to improve luminosity and contrast enhancement. The Generator function of GAN model uses U-Net algorithm for image segmentation using parallel layers in the U-net decoder. The use of image preprocessing gave moderated results based on benchmark standards. Also, morphological opening algorithm had better performance. In [8], Alan Lands et al., used techniques like Adam optimization, Drop out Regularization for experimentation and compared the efficiency of Resnet and Densenet models on a dataset. Densenet 121 gave a better accuracy. Also, Gaussian blur subtraction was applied for image preprocessing. A website was made where the patients can upload their fundus images and get the results without internet facility. In [19], T. Shanthi and R. S. Sabeenian, proposed an Alex-Net design with Decimation with spatial reduction. The images were segmented using Green channel segmentation. In [20], Muhammad Mateen, Junhao Wen et al., used VGG-19 for detecting the DR stages. They employed GMM for vessel segmentation along with PCA and SVD for feature vector dimensionality reduction for better results.

It can also be seen that learning methods like ensemble learning or transfer learning gave better results than individual Deep CNNs. In [21], Shuqiang Wang et al., due to the lack of labeled data and dispersion of features in medical images suggested the use of ensemble-based framework of GAN in semisupervised learning which is done for the first time here. The discriminator was fed features extracted from DenseNet and the sub fundus images local features which were generated by the parallel Generators. Then the results of the multiple discriminators were weightage collected. In [5], Wanghu Chen et al., tried to overcome the problem of overfitting in deep CNNS by proposing the use of 5 Shallow CNNs with several scales. Each CNN served as a base learner, consisting of two convolutional layers, two pooling layers, and an FC

layer. These multiple CNNs extracted various vision-related features and the output of each CNN is then integrated based on major voting and mean to do classification. In [22], Gaurav Saxena et al., also used the ensemble learning. The hybrid of Res-net and Inception net was created, i.e., The inception blocks with residuals blocks, which combined the output of the module's convolution operation with the input to get better results than standalone approaches. In [23], Sehrish Qummar et al., proposed an ensemble of 5 algorithms, i.e., Resnet, inceptionV3, Xception, Dense121, Dense169 with Nesterov-accelerated Adaptive Moment and SGD for multi-class classification. The input was divided into minibatches and the CNN was trained iteratively. This method received a good performance than the existing state of matter algorithms. In [24], Hongyang Jiang, et al., found that the CAM of the individual methods and the ensemble model of REsnet152, InceptionNetV3, InceptionResnetV2 along with ADAM optimizer for the DR classification proved that integrated/ ensemble technique is better by projecting the integrated CAM showing better interpretability.

In [25], Sagar B Hathwar et al., used the transfer learning approach for fine-tuning with Inception-ResnetV2, Xception models. In comparison to a random initialized model, the models achieved good prediction performance and converged substantially faster. For rating DR severity, the highest performing model achieved a quadratic weighted kappa score (2) of 0.88, compared to 0.81 for state-of-the-art approaches. In [4], Xianglong Zeng et al., paper took into consideration both the eyes unlike single eye fundus images into Inception V3, fine-tuned using transfer learning, to be classified into two classes. The images were preprocessed by normalization and aspect ratios, along with augmentation by random flips and random changes. The models for monocular and binocular were trained and compared. The model gave a better result as compared to monocular images and also a better accuracy. In [26], Ramon Pries at el., took 16 layered architecture which was fine-tuned using transfer learning method which resembles VGG-16 architecture but unlike original idea where 90% data is given to FC layers here only 25% data is given to FC layers while 75% was kept with the Convolution layers and also instead gradient descent the Nesterov method was used. The cross-entropy replaced the MSE as the objective function. To avoid local minima and deal with small datasets multi-resolution training was done. This method took less memory and space also gave better results with the datasets in comparison to the studied method. In [7], Asra Momeni Pour et al., the images were processed and then fed to Efficient Net-B5 for binary classification. The classifier is trained on a combination of two datasets before being evaluated on the third, i.e., Messidor-2 and IDRiD → Messidor and Messidor and Messidor-2 → IDRiD. The three scaling parameters: depth, width, and image resolution were hyper tuned for the network which is pre-trained on Image net. In [27], Shaohua Wana et al., removed the noise from the databases using NLMD, i.e., nonlocal Means Denoising. The classifiers: Alex net, VGGnet, Resnet, Googlenet were trained on Image net proving to achieve better results than random initialization. In [28], Wei Zhang et al., also pertained VGG-D on image net. Then Mini batch gradient descent algorithm along with Nesterov Momentum were done to overcome the problem of overfitting.

Other than these two methods there are many techniques like Grey wolf optimization which also helps in hyper tuning the CNNs achieving good results. In [29],

Thippa Reddy Gadekallu et al., used PCA for dimensionality reduction as feature engineering and GWO for hyper tuning the Deep neural network by optimizing the feature selection and outperforming the machine learning algorithms with an accuracy of 97.3% proving CNNs a better choice for DR detection.

Labeling of lesions is done in order to make detection easier, In [30], Hongyang Jiyang et al., did multi-labeling of lesions in eye fundus images to speed up the collecting of DR fundus images with numerous lesions by having trained candidates assign annotations to lesions or other objects. The detection was then done using a combination of multi-label classification and Grad CAM. In ResNet50 three convolution layers were added instead of fully connected layer. The output of the last layer computed the Grad-CAM which was also multiplied with guided backpropagation of predicted result to give a Guided-Grad-Cam to give an outline of an exact area of each lesion. This output showed lesions in a fundus image very clearly making the process of detection easier. In [31], Qilei Chen et al., also implied the labeling of detection for better results. The papers showed the impact of small lesions on the detection of DR severity. The objective was to automate the task of small lesions detection in large images. The authors created their own labeling method which created a bounding box with location, size of box and label of lesion around the lesion. The proposed method, i.e., large FPN focused on smaller size feature maps, unlike a normal FPN, took into consideration the Po layer which contains the features for small lesions and mapped ROI on it. To improve the region proposal network of RCNN a center-focus target detection mitigated the problem of rejection suitable anchors. Then the Faster R_CNN was applied with FPN and LFPN with LFPN giving better results. Similarly in [32], Jing Ni et al., used the lesion detection to enhance the classification process by creating the four patch heat maps containing probability for respective four lesions categories finding for each pixel, by a binary patch auxiliary classifier (ResNet-101) the suggested model is trained using a dataset, which is offered as an extra input channel because it gives finer information. The paper takes the feature maps generated by Inception V3 for both eyes and concatenate them to form a single feature layer using Relu activation function. Then the individual features were fed to auxiliary classifiers: ResNet-101 and concatenated map along with patch heat maps are fed to a fully connected layer with soft max function for labeling. The SeS is used for data augmentation. The new model provided improved accuracy, according to the report than Inception V3 for single eye and increasing the pixel resolution also played a positive impact and the addition of SeS and heat maps increased the kappa score.

The summary for the various studied papers is represented in the Table 2.

4 Methodology

The dataset collection method is followed by preprocessing to enrich and enhance the photos, which is then utilized to detect and categorize DR pictures using deep

Table 2 Methods used for DR detection/classification

Author and year	Dataset	Classification type	Preprocessing technique	Deep learning algorithm	Learning technique
R. Murugan et al. [12]	ROC, IRDiD	Binary classification	Image normalization	Deep CNN	Random initialization
Mobeen-ur-Rehman et al. [6]	MESSIDOR-1	Binary classification	Histogram equalization	Alex net, VGG, Vgg-Net-vd-16 and Vgg-Net-vd-19	Random initialization
Md. Sanaullah Chowdhury et al. [18]	DRIVE	Binary classification	CLAHE, Morphological Operations, Green channel	U-Net + GAN algorithm	Random initialization
Sagar B. Hathwar et al. [25]	KAGGLE, IRDiD, DRISHTI	BOTH	Image normalization	Inception-ResnetV2, Xception	Transfer learning
Harry Pratt et al. [13]	KAGGLE	Multi-class classification	Image normalization	Deep CNN	Random initialization
Shuqiang Wang et al. [21]	KAGGLE, MESSIDOR-2 , MESSIDOR-1	Both	Image normalization	Dense Net + GAN	Ensemble Learning
Ramon Pries et al. [26]	KAGGLE, MESSIDOR-2, DR2 dataset from Department of Ophthalmology, Federal University of Sao Paulo	Binary classification	Histogram equalization	VGG-16 + ADAM optimizer and Random forest optimizer	Transfer learning

(continued)

Table 2 (continued)

Author and year	Dataset	Classification type	Preprocessing technique	Deep learning algorithm	Learning technique
Meher Madhur Dharmana and M.S. Aiswarya [10]	a database of 732 images using eye fundus photography which are labeled with five tags: no, mild, moderate, severe, proliferative as 0, 1, 2, 3, and 4	Multi-class classification	Histogram equalization	Navie Bayes Classifier, SVM, ANN	Random initialization
Ara Momeni Pour et al. [7]	Messidor-1, Messidor-2 and IDRiD	Binary classification	CLAHE	Efficient Net-B5,	Random initialization
Wanghu Chen et al. [5]	KAGGLE	Multi-class classification	Histogram equalization, image normalization	Multi-scale Shallow 5 CNNs	Ensemble technique
Hongyang Jiyang et al. [30]	Messidor-1, private data by Beijing Tongren Eye center	Multi-class classification	Pixel normalization, CLAHE	ResNet50 + Guided Grad CAM	Transfer learning
P. Saranya and S. Prabakaran [16]	IDRiD, Messidor	Multi-class classification	Canny edge detection, interpolation, Image normalization, optic disk segmentation	Deep CNN	Random initialization
Alan Lands et al. [8]	APTOs 2019 By kaggle with 3662 images	Multi-class classification	Gaussian blur subtraction	ResNet50, DenseNet 121, DenseNet169, DenseNet256 + Adam optimization	Random initialization

(continued)

Table 2 (continued)

Author and year	Dataset	Classification type	Preprocessing technique	Deep learning algorithm	Learning technique
Ganjar Alfian et al. [11]	from mendelevdata.com of Lur and Lak population collected from 133 patients containing 13 relevant features NHSS KOREA with 1000 patients: 761 healthy and 239 disease	Binary classification	Image normalization	Deep NN + RE(Recursive Feature Elimination), Chi square, ANNOVA, extra trees, SVM kernels	Random initialization
Qilei Chen et al. [31]	The dataset is collected from 500 patients with all 5 categories, containing 5198 images of 2136 × 3216resolution, KAGGLE	Multi-class classification	Image normalization	FPN, Faster R-CNN, RPN, ResNet	Random initialization
Jing Ni et al. [32]	KAGGLE	Binary classification	–	Inception V3	Selective data sampling
Yunlei Sun 2019 [9]	from 301 hospitals containing 6 million records which are filtered to 3500: DR and Non-DR	Binary classification	Image normalization	Le-Net 5, GAN + ADAM optimizer, De-convolution	Random initialization
Gaurav Saxena et al. [22]	Eye pacs and Messidor 1 and Messidor 2	Binary classification	Histogram equalization	Inception net, Resnet SVM, ANN with Extreme gradient boosting	Ensemble learning

(continued)

Table 2 (continued)

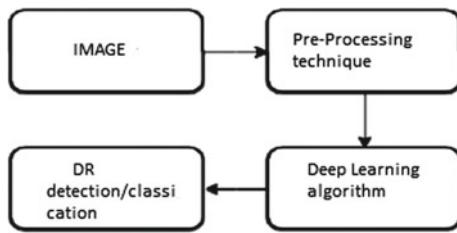
Author and year	Dataset	Classification type	Preprocessing technique	Deep learning algorithm	Learning technique
Sehrish Qummar et al. [23]	KAGGLE	Multi-class classification	Image normalization	Resnet, inceptionV3, Xception, Dense121, Dense169 with Nestrov accelerated Adaptive Moment and SGD	Ensemble learning
T. Shanthi and R.S. Sabeeenian [19]	MESSIDOR-1	Multi-class classification	Green channel	Alexnet with decimation	Random Initialization
K. Shankar et al. [17]	MESSIDOR-1	Multi-class classification	Green channel, optic disc segmentation, PCA, Histogram-based segmentation	Synergic Deep learning	Random initialization
Muhammad Mateen et al. [20]	KAGGLE	Multi-Class classification	GMM for vessel segmentation, PCA	VGGNet	Random initialization
Shaohua Wana et al. [27]	KAGGLE	Multi-class classification	Image normalization, Non local Means De noising	Alex net, VGGnet, Resnet, Googlenet	Transfer learning
Thippa Reddy Gadekallu et al. [29]	Debrecon dataset from UCI machine learning repository, MESSIDOR-1	Binary classification	Image Normalization, PCA for dimensionality reduction	GWO for hyper tuning the Deep NN	Grey wolf optimization
Wei Zhang et al. [28]	Macula-centered retinal fundus from Sichuan Academy of Medical Sciences and Sichuan Provincial Peoples Hospital	Binary classification	Histogram Equalization	Resnet and inception V3, Dense Net169, 201,,, Xception, InceptionResnetV2	Ensemble learning

(continued)

Table 2 (continued)

Author and year	Dataset	Classification type	Preprocessing technique	Deep learning algorithm	Learning technique
Arkadiusz Kwasigroch et al. [33]	KAGGLE	Binary classification	Image normalization, image correction	VGG-D algorithm With minibatch gradient descent algorithm with Nestrov Momentum	transfer learning
Hongyang Jiang et al. [24]	Beijing Tongren Eye centre	Binary classification	Image normalization	REsnet152, InceptionNetV3, InceptionResnetV2 with ADAM optimizer and CAM	Ensemble learning
Xianglong Zeng et al. [4]	KAGGLE	BOTH	Image normalization	Inception V3	Transfer learning
D. Jude Hemanth et al. [14]	MESSIDOR-1	Multi-Class classification	Histogram Equalization, CLAHE, RGB segmentation	Deep CNN	Random initialization
K. Shankar et al. [15]	MESSIDOR-1	Multi-Class classification	CLAHE, Histogram-based segmentation	Inception V4	Random initialization

Fig. 5 The process for DR classification/detection



learning. Deep learning techniques are then used to extract features and classify the photos, as illustrated in Fig. 5. These steps are explained in the following sections.

4.1 DR Datasets

This section covers the various datasets which are used in the studied papers. These are used for training, validating, and testing the proposed algorithms and do a comparison with other works. There is a variety of fundus image datasets that are public. Commonly used datasets are:

4.1.1 KAGGLE [13, 25]

It contains 88,702 high-resolution photos acquired from several cameras with resolutions ranging from 433×289 pixels to 5184×3456 pixels. Each image is assigned to one of five DR phases. Only the ground facts for training photos are provided to the public. The photographs are of low quality, and several are labeled incorrectly.

4.1.2 DRIVE [18]

The DRIVE database is the result of a Dutch diabetic retinopathy screening program. There were 400 people in this programme who were between the ages of 25 and 29. A total of 40 photos were chosen at random from these for the DRIVE database. These photos were separated into two categories: training and testing.

4.1.3 MESSIDOR [21]

The collection contains 1200 fundus color images with a 45-degree field of view. The image is 1440 960 pixels wide, 2240 1488 pixels tall, or 2304 1536 pixels tall. Based on the quantity of MA, HM, and the presence of NV, each image is assigned to one of four lesion grades (R0, R1, R2, and R3).

4.1.4 MESSIDOR-2 [26]

This database is a supplement to the Messidor dataset, which is a collection of diabetic retinopathy examinations with two macula-centered eye fundus photographs in each (one per eye). The collection contains 1748 photographs with a 45-degree field of view.

4.1.5 Indian Diabetic Retinopathy Image Dataset (IDRiD) [12]

There are 516 images in the Indian Diabetic Retinopathy Image Database (IDRiD). The retinal pictures are separated into two categories: those with DR indications and those without. The dataset contains ground truth on DR symptoms and typical retinal architecture.

4.1.6 ROC Dataset [12]

There are 100 digital fundus pictures in the Retinopathy Online Challenge (ROC) dataset. It was discovered during a diabetic retinopathy test. Topcon NW 100, Topcon NW 200, and Canon CR5-45NM with 50 degree FOV were used to capture the photographs. All of the photographs were saved in JPEG format.

Table 3 Details of DR datasets

Dataset	Image number	normal image	Mild DR	Training sets	Test sets	Image size
Kaggle	88,702 images	–	–	35,126 images	53,576 images	Image resolutions ranging from 433 × 289 pixels to 5184 × 3456 pixels are available
DRIVE	40 images	33 images	7 images	20 images	20 images	565 × 584 pixels
Messidor	1200 images	–	–	–	–	1440 × 960, 2240 × 1488, or 2304 × 1536
Messidor-2	1748 images	–	–	–	–	Different image resolution
IDRiD	516 images	–	–	413 images	103 images	4288 × 2848 pixels
ROC	100 images	–	–	50 images	50 images	Different image resolution

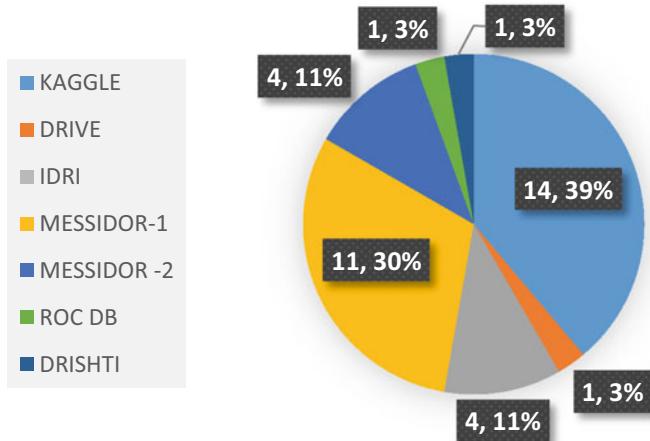


Fig. 6 Distribution of studied papers based on dataset selected

Table 3 summarizes these data sets discussing the quantity of images, categories, the training set, test sets, and the image sizes.

Many other datasets were studied which were collected by authors and labeled by experts for research. In [12] Assam medical college dataset which contained 50 images, 24 normal and 26 abnormal was used for testing the proposed work. Similarly in [26] the DR2 dataset from Department of Ophthalmology, Federal University of Sao Paulo was used for testing proposes. Also in [10] the database consisted of 732 images which were labeled with 5 tags: no, mild, moderate, severe, proliferative as 0, 1, 2, 3, and 4. The private data provided by Beijing Tongren Eye Center was used in [30, 24]. The data was processed with sensitive information elimination. Researchers of [28] considered Macula-centered retinal fundus from Sichuan Academy of Medical Sciences and Sichuan Provincial Peoples Hospital for DR classification [9, 29] and [31] also, trained and tested on the datasets collected privately for DR classification.

The study done shows that Kaggle dataset is most commonly used. Figure 6 shows how many times a dataset is used in the papers.

4.2 Preprocessing Techniques

To make an image structure stand out, it needs to be preprocessed. Depending on the image's nature, several preprocessing procedures are taken. The preprocessing steps are used to remove variations and noise, also to enhance image contrast and image quality of the eye fundus datasets. These steps also include image normalization and non-uniform illumination correction in order to remove unwanted traces and to

improve the overall performance of the proposed approaches. This section discusses the various preprocessing techniques employed in the publications examined.

4.2.1 Image Normalization

Image normalization is used to overcome the variability in fundus images. In [6] the color normalization was implemented on the dataset to get a resized 512X512 pixels set which retained the complicated features. In [4, 9, 11, 16, 29] the pixel values were normalized from [0,255] to [1, 1] for avoiding negative effects. In [5, 21, 30], the images were resized for normalization. In [23, 33] To avoid feature bias and speed up training, the photos were shrunk and mean normalized. In [25, 27] nonlocal means denoising was used to remove noise. In [24] the images were resized and the Gaussian filter was applied to pre-process the images. In the survey, 15 papers out of 30 have used image normalization for fundus images preprocessing.

4.2.2 Histogram Equalization

HE is one of the most extensively used image processing techniques for increasing image visibility and quality. It broadens the dynamic range of the target's histogram. The grey levels are distributed uniformly in the final image. CLAHE is a HE extension that is used to enhance photographs in the medical field [14]. CLAHE separates the images into nominal partitions before applying histogram equalization to each of them. This equalizes the distribution of grey values used in the image, increasing the image's hidden qualities. In [18] CLAHE was used to enhance the contrast on LAB images with Numtiles of 8 by 8 and Clip limit of 0.0001 (Fig. 7).

Also [7, 26, 30] and [15] used CLAHE for contrast enhancement. But HE is also used in [5, 6, 10, 22] to deal with excessively bright and dark background and foreground. In [14] both HE and CLAHE were used on R G B section with CLAHE giving better results as compared to HE. To strengthen edge definitions and boost

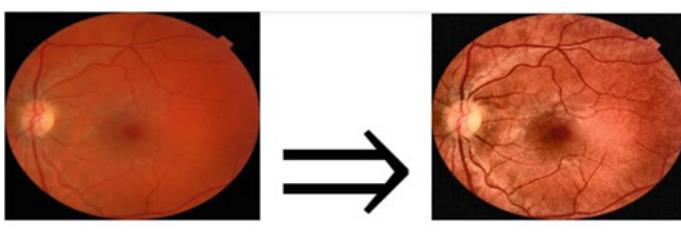


Fig. 7 The increase in image contrast with the CLAHE method [7]

local contrast in each image region, [28] used Adaptive Histogram equalization. In the study conducted 11 papers enhanced the contrast of the images with CLAHE giving better results than HE.

4.2.3 Morphological Operations[18]

Morphological Opening is defined as performing erosion followed by a dilation operation, and it may be written as $I \circledast e = (I \ominus e) \oplus e$. A binary image I is eroded by structuring components of kernel size $e \times e$ (denoted 10×10), resulting in a new binary image $g = (I \ominus e)$. If all the pixels of image I covered by the kernel are 1 (or 0), the targeted pixel $I(x, y)$ is considered 1 (or 0) in the erosion method (or 0). If one pixel covered by the kernel is 1, dilation of an image I by a structural element d yields a new binary image $g = I \oplus d$ with ones. Dilation is the polar opposite of erosion.

1. Calculating the background image.
2. Deduction of the image from the background.
3. Morphological Closing ($I \ominus e \oplus e$) is the process of doing dilatation followed by an erosion operation.

In [18] morphological operations were performed on the images to deal with luminosity issues and variations in the images performing better than image contrast enhancement.

4.2.4 Green Channel

Color as a feature can be used to separate eye features from one another. MAs and HMs appear as red spots in the RGB imagery, whereas EXs show as yellow spots. The green channel refines the photos by providing finer details of characteristics, resulting in improved results [19]. In this monochrome image, red lesions (MAs and HMs) seem dark in the green channel while white lesions (EXs) seem brilliant. The Green channel was utilized in [19] to increase the multi-class classification accuracy of a modified Alex net. The Green channel was also extracted for finer feature details and further preprocessing in [2, 17]. In the study, 3 out of 30 extracted the green channel for preprocessing.

4.2.5 Image Correction Techniques

Blurring an image by a Gaussian function for image preprocessing is known as Gaussian-blur correction. It is used to reduce image appeal and diminish detail. In [8, 33] Gaussian blur subtraction was used to proliferate the details of an image using normalized box filters.

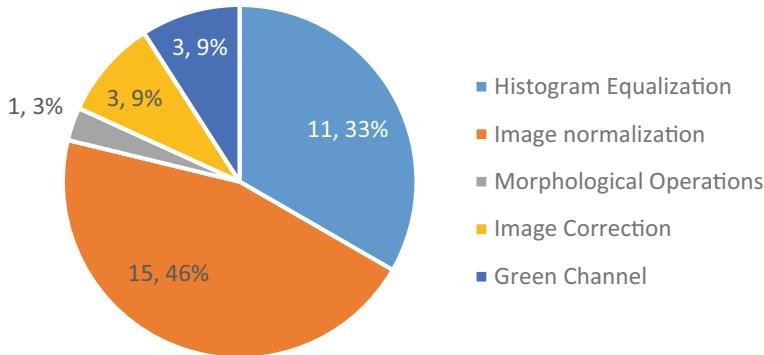


Fig. 8 Distribution of studied papers based on PreProcessing Techniques in DR detection

The statistical features of an image are altered by non-uniform illumination, which has an impact on the system's performance. In [20] the grayscale conversion technique and to estimate the image, the shadow correction approach was used, then subtracted it from the original image to extract more information from the background.

In 3 out of 30 studied papers, image correction techniques were used for more feature extraction.

Figure 8 reveals that image normalization is an essential preprocessing step. Also, for contrast enhancement CLAHE worked out to be the best.

4.3 Training Techniques

This section discusses about the various strategies of training a neural network. Neural networks are trained by random weights initialization and the weights are updated using back propagation. Some techniques to train the neural networks are discussed as follows:

4.3.1 Transfer Learning

Transfer learning is a distinct technique to random initialization in which the network weights are initialized from models pre-trained to categorize objects in the ImageNet dataset. It is a technique in which a model that has been trained for one task is used to train a model for a different task. In [25–27, 29, 33] and [4] the deep learning algorithms were pre-trained on ImageNet dataset to make DR severity grading and binary classification more effective. In Fig. 9 a block schematic of a typical transfer learning methodology is shown. This strategy involves pre-training the model on a sufficiently large, well-annotated dataset, such as ImageNet, and then fine-tuning it

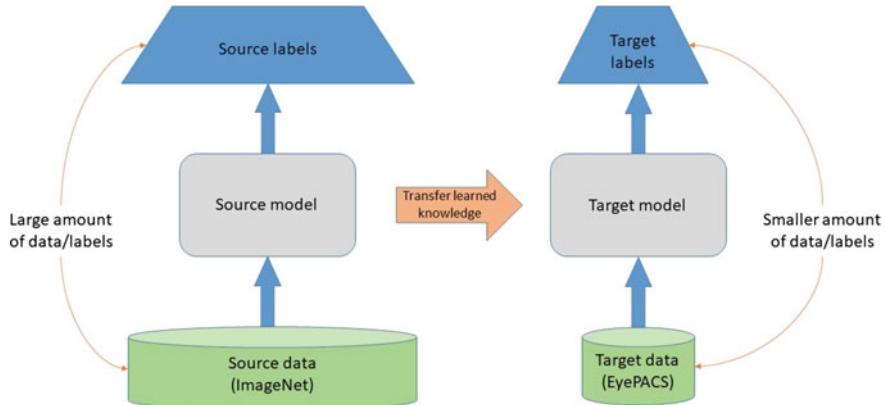


Fig. 9 Block diagram of transfer learning

on a smaller dataset relevant to the application domain. DR severity rating can be improved by fine-tuning CNNs that have already been trained.

4.3.2 Ensemble Learning

Even if deep learning models are finely trained, the final well-trained models might have several local minima. In this technique several deep learning models are trained independently and then their results are used for prediction. In [24], the paper used Adaboost algorithm to create a strong classifier by combining multiple weak classifiers. In [21] the same strategy was used to design the ensemble discriminative model for final results, the weight parameter is assigned to each discriminator using a weighted-based fusion approach. In [5] five shallow multi-scale CNNs were ensemble and then majority voting was done to make predictions. Also [22, 23, 28] used ensembling of well known algorithms like Resnet and inception V3, Dense Net169, Dense Net201, Xception, InceptionResnetV2 obtaining better results than independent algorithms.

4.4 Performance Measures

There are a variety of performance metrics that can be used to assess the classification performance of DL algorithms.

4.4.1 Sensitivity

It's the proportion of aberrant photos that are classified as such [28].

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \quad (1)$$

4.4.2 Specificity

It's the proportion of normal photos that are categorized as such [28].

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \quad (2)$$

4.4.3 Area Under Curve

AUC is a graph that shows the relationship between sensitivity and specificity. AUC of 0.5 shows that the classifier is random guessing. The capacity to evaluate grading performance for unbalanced data sets is AUC's key advantage [28].

4.4.4 Accuracy

It is the percentage of images that are classified correctly. The following is the equations of each measurement [28].

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TN} + \text{TP} + \text{FN} + \text{FP}) \quad (3)$$

4.4.5 F1 Score

It is the harmonic average of precision and recall. An F1_score close to 1 indicates good performance [28].

$$\text{F1 Score} = 2\text{TP}/(2\text{TP} + \text{FP} + \text{FN}) \quad (4)$$

The number of disease images recognized as a disease is known as true positive (TP). The number of normal images labeled as normal is known as true negative (TN), while the number of normal images labeled as a disease is known as false positive (FP). The number of illness images labeled as normal is known as false negative (FN).

4.4.6 Kappa Score

It is the degree of agreement between the human-assigned and network-predicted ratings. The Kappa score ranges from -1 to 1 , with -1 denoting extreme disagreement and 1 denoting full agreement. Three specific matrices are used to calculate the Kappa score. The confusion matrix O is built first. After that, a weight matrix W is generated, with elements determined using a formula [33]:

$$w_{i,j} = \frac{(i - j)^2}{(N - 1)^2} \quad (5)$$

where: N —number of instances.

Next, the matrix E of expected ratings is calculated. Finally, given matrices are used to calculate Kappa score:

$$k = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \quad (6)$$

4.4.7 Receiver Operating Characteristic Curve [18]

It's a graphical representation of a binary classifier system's diagnostic capability. A probability curve is referred to as a ROC curve.

In Fig. 10 the bars show the number of papers that used the discussed measures. It can be seen that specificity, sensitivity, and accuracy are the most often used performance measures.

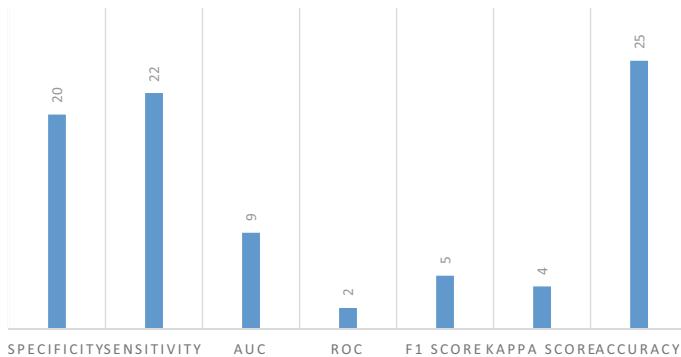


Fig. 10 Distribution of studied papers based on Performance Measures

4.5 Classification Methods

This section summarizes the way DR classification can be done whether binary classification or multistage detection and also tries to bring the light to a better method.

4.5.1 Binary Classification

This classification divides images into two categories: referable DR (refer to a moderate stage or higher) and non-referable DR (refer to a low stage or lower) (No DR or mild stage). In [4] a CNN with Siamese network architecture is proposed to detect diabetic retinopathy by classifying color retinal fundus photographs into two grades. Also in [7, 9, 11, 12, 18, 21, 22, 25, 26, 32] and [24, 28, 29, 33] the authors classified the DR data into referable or non-referable DR.

However, in majority of the studies, the researchers did not grade the DR along with the diagnoses. The DR stages are important in determining the exact stage of DR in order to treat the retina with the proper procedure and avoid deterioration and blindness.

4.5.2 Multi-class Classification

This section reviews the papers in which the DR grading defined as: No DR, Mild DR, Moderate DR, Severe DR and Proliferative DR of DR dataset was done. The researchers in [13] introduced a CNN method for multi-class classification of DR in fundus images in Kaggle database. Several other papers, [4, 5, 8, 10, 13–17, 19–21, 23, 25, 27, 30] did multi-level classification.

Also, it can be seen that many researchers did both types of classification for better results. In [4] A convolutional neural network model with Siameselike architecture which was trained with transfer learning was given binocular fundus images as input. The method used the correlation between the images to make predictions. Also, the binocular model was trained and evaluated on a 10% validation set for five-class DR detection. In [21] for the 4-category grading of DR (R0, R1, R2, and R3) the proposed method using multichannel-based SSGAN achieved the accuracy of 84.23%. For the binary classification of DR, achieved the accuracy of 96.6%. In [25] the CNN architecture achieved a high sensitivity of 94.3% and specificity of 95.5%(for DR/no-DR classification) and a quadratic weighted kappa score (κ^2) of 0.88 for multi-level; grading (0–4).

Figure 11 shows the distribution of studied papers depicting the type of classification done.

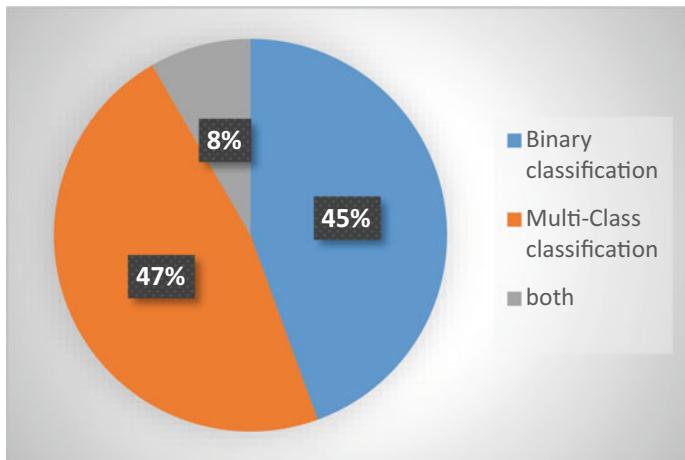


Fig. 11 Distribution of studied papers based on classification types

5 Challenges Faced in DR Detection

Existing researches have been unable to achieve a good accuracy distinguishing all stages of DR as detection of multiple lesions is a challenging task. The colored lesion photographs are used in the current works to diagnose DR. Manual interpretation can only be performed by highly qualified domain specialists and is, therefore, time and expense intensive. The impacts of various hyperparameter tuning (meta-learning) and its ramifications are not examined in the current CNN architecture-based ensemble model for DR detection. To be able to perform feature selection along with ensemble of CNNs might improve the results. Consideration of Multiple datasets for testing the effectiveness of proposed methods is still a challenge.

6 Results and Discussion

The objective is to find a process in which the stages of DR detection and classification are the ones that give the most promising and accurate results. The combination of

Table 4 Most broadly used

Methodology	Name
Dataset	Kaggle dataset
Preprocessing	CLAHE
Learning technique	Ensemble method
Classification	Multi-class classification

these stages gives a better performance when studied. Table 4 is based on review of these 30 papers.

To study the effects of contrast enhancement on the images a sample of 10 images were processed using HE and CLAHE on python. The images contained lesions which were unnoticeable to human eyes, this can result in severe DR as the result of any detection method completely rest on the excellence of the images. Therefore contrast enhancement was needed to suppress the noise. Because CLAHE boosts local brightness, the contrast ratio in every region of the image can be improved. Whereas HE is a global enhancement method, it limits the contrast ratio in specific parts of the image. This results in significant contrast losses in small regions of the image, especially in the background. It was found that CLAHE is better than HE. Figures 12 and 13 shows the left and right image of one ID in its original, and after HE and CLAHE. It can be clearly seen that lesions were more specified after CLAHE.

From Table 5 it can be inferred that multi-class classification i.e. DR level detection along with CLAHE in ensemble model has achieved an accuracy of 96% which is better than any other results obtained on Kaggle dataset with any other method. Also in binary classification, CLAHE as a preprocessing step and the DL model hyper tuned using transfer learning gave an accuracy of 98% better than others. Although a lot of work is being done in the level detection of DR still there is a lot of scope for improvement.

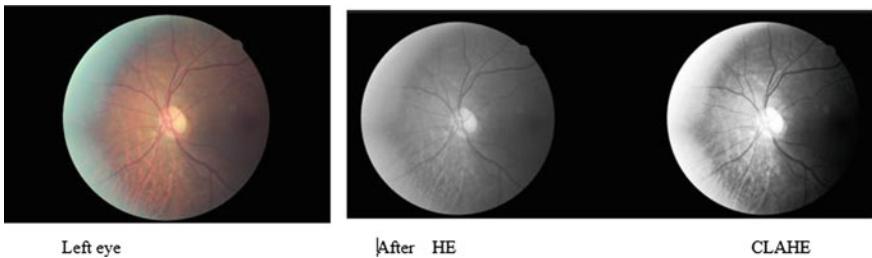


Fig. 12 Left eye of a subject from KAGGLE dataset before and after HE and CLAHE

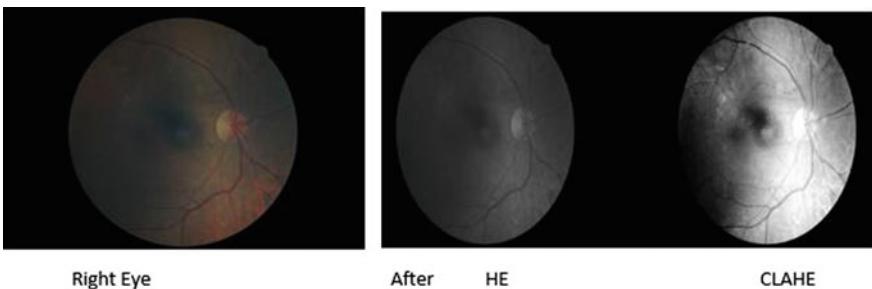


Fig. 13 Right eye of a subject from KAGGLE dataset before and after HE and CLAHE

Table 5 Result comparison for KAGGLE dataset

learning method	Dataset	Preprocessing technique	Classification	Result
Transfer learning [25]	Kaggle	Others	Multi-level	Specificity: 0.95 Sensitivity: 0.943 Kappa Score: 0.88
Random initialization [13]	Kaggle	Others	Multi-level	Specificity: 0.95 Sensitivity: 0.3 Accuracy: 75%
Transfer learning [4]	Kaggle	Others	Multi-level	Specificity: 0.707 Sensitivity: 0.82 Kappa Score: 0.10 Accuracy: 95%
Transfer leaning [26]	Kaggle	CLAHE	BINARY	Accuracy: 98%
Ensemble [21]	Kaggle	CLAHE	Multi-level	Specificity: 0.96 Sensitivity: 0.97 Accuracy: 96.6%
Ensemble [5]	Kaggle	CLAHE	Multi-level	Accuracy: 86%
Random initialization [8]	Kaggle	Others	Multi-level	Kappa Score: 0.80 Accuracy: 93%
Random initialization [31]	Kaggle	Others	Multi-level	Sensitivity: 0.94
Selective data sampling [32]	Kaggle	Others	BINARY	Kappa Score: 0.88 Accuracy: 87.2%
Ensemble [22]	Kaggle	CLAHE	BINARY	Specificity: 0.89 Sensitivity: 0.88

7 Conclusion and Future Scope

Automated technology has suggestively lessened the time required to detect DR thus saving struggle and expenses for ophthalmologists. This in return also helps in the appropriate treatment of patients. This article has reviewed about application of deep learning methodologies in diabetic retinopathy detection. In retinal pictures, deep learning techniques are quite beneficial and effective. Also various techniques which are used for preprocessing of DR images are discussed and based on studied papers image normalization and CLAHE are mostly used preprocessing techniques. To find the effects of CLAHE, images from Kaggle dataset were processed and the results were very promising. The processed images were noise-free and lesions were clearly visible to human eyes. There are various public and private datasets which are used for training but KAGGLE is the favorite of the researchers. This review shows that the ensemble leaning is a growing trend in the present years for training the Deep learning algorithms. Various performance measures of the studied papers were discussed with specificity and sensitivity mostly focused on. In the end,

the classification types were also discussed, where it could be seen that multi-level classification is equally important as DR detection since it helps in identifying the risks at the particular stage. This review in this domain emphasized on detailed reviews of different state-of-the-art deep learning algorithms, learning techniques, datasets available, various preprocessing methods and classification types. Though the paper tries to review the latest researches by choosing the papers from the most recent years but there is a scope of more learning. The future work seeks to implement feature selection techniques like Grey wolf optimization on ensemble of CNN models to detect the level severity of the images and to perform DR detection. Also, an effort will be done to test the proposed model on several databases to check the accuracy of the prosed work.

References

1. American academy of ophthalmology-what is diabetic retinopathy? [Online]. Available <https://www.aao.org/eye-health/diseases/what-is-diabetic-retinopathy>
2. Sengupta S, Singh A, Leopold H A, Gulati T, Lakshminarayanan V (2019) Ophthalmic diagnosis using deep learning with fundus images—A critical review. Artificial Intelligence in Medicine, 101758. <https://doi.org/10.1016/j.artmed.2019.101758>
3. Chakrabarti R, Harper CA, Keeffe JE (2012) Diabetic retinopathy management guidelines. Expert Review of Ophthalmology 7(5):417–439. <https://doi.org/10.1586/eop.12.52>
4. Asiri N, Hussain M, Adel FA, Alzaidi N (2019) Deep Learning based Computer-Aided Diagnosis Systems for Diabetic Retinopathy: A Survey. Artif Intell Med. <https://doi.org/10.1016/j.artmed.2019.07.009>
5. Chen W, Yang B, Li J, Wang J (2020) An Approach to Detecting Diabetic Retinopathy Based on Integrated Shallow Convolutional Neural Networks. IEEE Access 8:178552–178562. <https://doi.org/10.1109/ACCESS.2020.3027794>
6. Mobeen-ur-Rehman Khan SH, Abbas Z, Danish Rizvi SM (2019) Classification of diabetic retinopathy images based on customised CNN architecture. Amity International Conference on Artificial Intelligence (AICAI). <https://doi.org/10.1109/aicai.2019.8701231>
7. Pour AM, Seyedarabi H, Jahromi SHA, Javadzadeh A (2020) Automatic detection and monitoring of diabetic retinopathy using efficient convolutional neural networks and contrast limited adaptive histogram equalization. IEEE Access, 1. <https://doi.org/10.1109/access.2020.3005044>
8. Lands A, Kottarakkithil AJ, Biju A, Jacob EM, Thomas S (2020) Implementation of deep learning based algorithms for diabetic retinopathy classification from fundus images. 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), pp 1028–1032. <https://doi.org/10.1109/ICOEI48184.2020.9142878>
9. Sun Y (2019) The neural network of one-dimensional convolution—An example of the diagnosis of diabetic retinopathy. IEEE Access, 1. <https://doi.org/10.1109/access.2019.2916922>
10. Dharmana MM (2020) Pre-diagnosis of Diabetic Retinopathy using Blob Detection. 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp 98–101, <https://doi.org/10.1109/ICIRCA48905.2020.9183241>
11. Alfian G, Syafrudin M, Fitriyani NL, Anshari M, Stasa P, Svub J, Rhee J (2020) Deep neural network for predicting diabetic retinopathy from risk factors. Mathematics 8(9):1620. <https://doi.org/10.3390/math8091620>
12. Murugan R, Roy P, Singh U (2020) An abnormality detection of retinal fundus images by deep convolutional neural networks. Multimedia Tools Appl <https://doi.org/10.1007/s11042-020-09217-6>

13. Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y (2016) Convolutional neural networks for diabetic retinopathy. Procedia Computer Science 90:200–205. <https://doi.org/10.1016/j.procs.2016.07.014>
14. Hemanth DJ, Deperlioglu O, Kose U (2019) An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. Neural Comput Appl. <https://doi.org/10.1007/s00521-018-03974-0>
15. Shankar K, Zhang Y, Liu Y, Wu L, Chen CH (2020) Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. IEEE Access, 1. <https://doi.org/10.1109/acc.ess.2020.3005152>
16. Saranya P, Prabakaran S (2020) Automatic detection of non-proliferative diabetic retinopathy in retinal fundus images using convolution neural network. J Ambient Intell Human Comput. <https://doi.org/10.1007/s12652-020-02518-6>
17. Kathiresan S, Sait ARW, Gupta D, Lakshmanaprabu SK, Khanna A, Pandey HM (2020) Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model. Pattern Recogn Lett. <https://doi.org/10.1016/j.patrec.2020.02.026>
18. Chowdhury MS, Taimy FR, Nahid A-A, Ali MY, bin Ali F (2019) Retinal fundus identification utilizing supervised and unsupervised nature of deep neural network. 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). <https://doi.org/10.1109/icasert.2019.8934640>
19. Shanthi T, Sabeenian RS (2019) Modified Alexnet architecture for classification of diabetic retinopathy images. Comput Electr Eng 76:56–64. <https://doi.org/10.1016/j.compeleceng.2019.03.004>
20. Muhammad M, Wen J, Nasrullah, D, Song S, Huang Z (2018) Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry 11:1. <https://doi.org/10.3390/sym11010001>
21. Wang S, Wang X, Hu Y, Shen Y, Yang Z, Gan M, Lei B (2020) Diabetic retinopathy diagnosis using multichannel generative adversarial network with semisupervision. IEEE Transactions on Automation Science and Engineering, 1–12. <https://doi.org/10.1109/tase.2020.2981637>
22. Saxena G, Verma D, Paraye A, Rajan A, Rawat A (2020) Improved and robust deep learning agent for preliminary detection of diabetic retinopathy using public datasets. Intelligence-Based Med 3–4:100022. <https://doi.org/10.1016/j.ibmed.2020.100022>
23. Qummar S et al (2019) A deep learning ensemble approach for diabetic retinopathy detection. IEEE Access 7:150530–150539. <https://doi.org/10.1109/ACCESS.2019.2947484>
24. Jiang H, Yang K, Gao M, Zhang D, Ma H, Qian W (2019) An interpretable ensemble deep learning model for diabetic retinopathy disease classification. 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). <https://doi.org/10.1109/embc.2019.8857160>
25. Hervella ÁS, Ramos L, Rouco J, Novo J, Ortega M (2020) Multi-modal self-supervised pre-training for joint optic disc and cup segmentation in eye fundus images. ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 961–965. <https://doi.org/10.1109/ICASSP40776.2020.9053551>
26. Pires R, Avila S, Wainer J, Valle E, Abramoff MD, Rocha A (2019) A data-driven approach to referable diabetic retinopathy detection. Artificial Intelligence in Medicine. <https://doi.org/10.1016/j.artmed.2019.03.009>
27. Wan S, Liang Y, Zhang Y (2018) Deep convolutional neural networks for diabetic retinopathy detection by image classification. Comput Electr Eng 72:274–282. <https://doi.org/10.1016/j.compeleceng.2018.07.042>
28. Zhang W, Zhong J, Yang S, Gao Z, Hu J, Chen Y, Yi Z (2019) Automated identification and grading system of diabetic retinopathy using deep neural networks. Knowledge-Based Systems 175. <https://doi.org/10.1016/j.knosys.2019.03.016>
29. Gadekallu TR, Khare N, Bhattacharya S, Singh S, Maddikunta PKR, Srivastava G (2020) Deep neural networks to predict diabetic retinopathy. J Ambient Intell Humaniz Comput. <https://doi.org/10.1007/s12652-020-01963-7>

30. Jiang H, et al (2002) A multi-label deep learning model with interpretable Grad-CAM for diabetic retinopathy classification. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp 1560–1563. <https://doi.org/10.1109/EMBC44109.2020.9175884>.
31. Chen Q, Sun X, Zhang N, Cao Y, Liu B (2019) Mini lesions detection on diabetic retinopathy images via large scale CNN features. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp 348–352. <https://doi.org/10.1109/ICTAI.2019.00056>
32. Ni J, Chen Q, Liu C, Wang H, Cao Y, Liu B (2019) An effective CNN approach for diabetic retinopathy stage classification with dual inputs and selective data sampling. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pp 1578–1584. <https://doi.org/10.1109/ICMLA.2019.00260>
33. Kwasigroch A, Jarzembski B, Grochowski M (2018) Deep CNN based decision support system for detection and assessing the stage of diabetic retinopathy. International Interdisciplinary PhD Workshop (IIPHDW) 2018:111–116. <https://doi.org/10.1109/IIPHDW.2018.8388337>

A Study of Recommendation System on OTT Platform and Determining Similarity and Likeliness Among Users for Recommendation of Movies



Mohd Saquib, Aqeel Khalique, and Imran Hussain

Abstract In this article, we studied the recommendation system and proposed a recommendation system for OTT platforms that defines a relationship between users by calculating likeliness between users to create better recommendations. We implement our proposed model to calculate likeliness between users with the help of similarity between users. Results show that defining a relationship between users will certainly improve recommendations and also contribute to better user experience on OTT platforms.

Keywords Recommendation system · OTT platform recommendations · Likeliness · Similarity · Movie recommendation

1 Introduction

During the most recent couple of years, with the increase in the usage of OTT Platforms and various different internet amenities, use of recommendation systems has increased in our lives. From Internet services such as e-commerce (recommending articles that buyers may be interested in) to online promotions (recommending users the right content to match their preferences), recommendation systems are now inevitable in our daily online tours. In General, a recommendation system refers to a system that is competent in forecasting the long-term preferences of a group of things for a user and suggests better options. Generally, recommender systems are methods directed towards proposing appropriate articles to consumers (articles are movies, novels, products purchased according to the industry). Recommender systems are very important in certain industries because they will generate a lot of revenue when implemented [1].

M. Saquib (✉) · A. Khalique (✉) · I. Hussain
Department of CSE, SEST, Jamia Hamdard, New Delhi, India

I. Hussain
e-mail: ihussain@jamiahAMDARD.ac.in

Recommendation System has become an essential component of our lifestyle where people depend upon knowledge for deciding about their personal interests. Recommendation system is subclass of data filtering to predict preferences to the things utilized by or for users. Although there are many approaches developed in past search still goes on thanks to its tremendous use in many applications [2].

1.1 Need of Recommendation System

Due to the rise of Internet people have too many options to choose from nowadays, which is one key reason why we need a recommender system in our modern society. Before the rise of the internet, people used to shop from offline stores, in which the items available were limited. For instance, the amount of shoes that can be placed in a showroom used to depend on the size of that showroom. But, nowadays, the web allows people to acquire ample resources online. Netflix, for instance, has a huge cluster of movies. Although the quantity of accessible data increased, a new problem came to light when people started facing problems while choosing the product they want to watch or buy [3–5].

1.2 Working Principle

To be more effective in predicting suggestions for the user, recommendation systems need to know about the user's preferences. So, gathering and combining the user's data is an important part of the development. In data combination, direct interactions like user's reviews, ratings and other information such as age, interests, or gender combine with indirect interactions such as location, dates, device used, and clicks on a link [6].

User interaction information is the base for building a recommendation system. The more data we have about the user the more efficient our recommendation system will be. The information must contain the users' past practices, customer's connection with other consumers, and affinity between numerous items. Online shopping giant "Amazon" uses a recommendation system to recommend products to users on the basis of items they bought in the past. It also recommends the most popular and similar products. Recommendation systems work on the principle of finding patterns in users' past practices. Information collection is one of the main phases in the success of a recommendation system. Information about the user can be collected directly or indirectly [4]. For instance, Netflix already recommends us the movie that we would love to watch, it gives us a list of choices based on our watching history or our previous online excursions. This saves a lot of user's precious time and gives the user a better experience. This function helped Netflix a lot in lowering cancelation rates [6]. There are so many applications of recommendation systems. Figure 1 shows the application areas of recommendation systems.

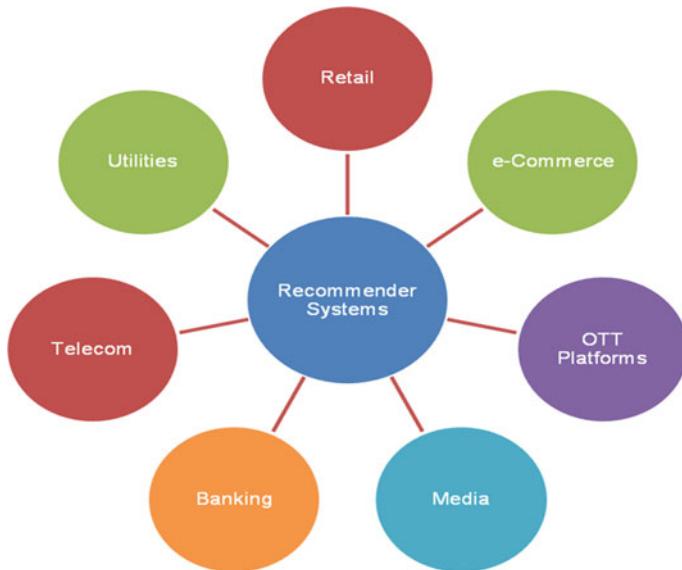


Fig. 1 Application area of recommendation systems

1.3 *Approaches*

There are three basic approaches that we can use to develop a recommendation system, namely Collaborative filtering, Content-Based filtering, and Hybrid approaches.

Collaborative filtering

This method is deployed by assembling and examining data on user's activity, their online practices or liking, and predicting what they will like on the basis of their similarity with other users. The main advantage of collaborative filtering is that it does not depend on machine analyzed content and that is why it is able to recommend something so complex like a movie without even requiring an understanding of that item. This type of filtering is established on the assumption that people who liked a certain kind of items in the past will also like a similar type of items in future also. For instance, if user A likes article (i, ii, iii, iv) and user B likes article (ii, iii, iv, v) then we can say that they have similar preferences and there is a possibility that user A would like item v.

Content-based filtering

This filtering method is established on the illustration of an article and single user's exchange of information and fondness. Data collected from user's browsing history and interactions are used in recommending items to the user. In a content-based recommendation method, exceptional keywords are used to describe the articles and

a consumer profile is also generated to state the type of articles this consumer is interested in. Recommendation systems that are based on content-based filtering techniques use methods that suggest those articles to the consumer which are similar to the ones in which consumer has shown interest in the past. In this approach accuracy is directly proportional to the information, the more information we have about the user, the higher the accuracy.

Hybrid Recommendation Systems

Hybrid technique is the amalgamation of collaborative filtering and content-based technique. Nowadays, it is evident that due to ample amount of research in this area combining content-based and collaborative technique is more effective as compared to using them separately. We can use them by implementing them separately or combining them. We can also combine content-based techniques to a collaborative method or the other way around, or by combining the methods as one blueprint. It is observed that hybrid model performs better. Netflix also uses hybrid type recommendation system. To overcome the usual problems such as cold start problem Hybrid methods are used.

2 Related Work

This segment highlights the existing modern approaches to the suggestion frameworks that define a social bond between users. Social Bond based suggestion frameworks have been broadly considered because social trust presents a different perspective of user likings apart from item ratings. Integrating social trust can upgrade the recommendation system. Nowadays social networking sites have started using the information present in social networks to forecast user behavior. These approaches prove the existence of a social network among users can be used to predict and recommend better items to the user. Table 1 shows the related work.

3 Proposed Solution

Good personalized recommendations can add another dimension to the user experience. On OTT platforms like Netflix, Amazon Prime, and others, there is no parameter with the help of which a user would be able to know that who among the users on the OTT platform is more similar to him/her. Users do not know the value of likeliness (social bond) between them.

We propose a deterministic and scalable model for calculating likeliness among viewers based on user similarity and recommendations. We do not restrict our implementation only on OTT platforms; however, our model can be applicable in all fields of recommendation systems. Our model also utilizes the social relation and determines likeliness among socially connected users. This section discusses framework

Table 1 Related work on recommendation systems

Paper reference	Techniques	Key findings	Observation
[7]	Collaborative filtering	Includes profile similarity and trust-based recommendations	The results show that along with general similarity, there is also a relationship between trust and the massive unique variance in ratings
[8]	KNN	Includes a proposal that combines customer ratings and general trust for suggesting better recommendations	The results depict that for rating prediction distrust information is important and with the help of information propagation we can improve the performance. As compared to other trust-based recommendation systems, their proposal uses distrust links and look into their propagation effects
[9]	KNN	Includes multigraph ranking model to generate recommendations	The results showed discrete user connections in multiple graphs and advanced the multigraph classification prototype to identify and suggest the nearest neighbors of a particular user in a challenging environment
[10]	Collaborative filtering	Proposed a website that unites a social network based on the Linguistic System and implies faith to generate movie recommendations	In this website, trust is used in social networks to customize the customer experience. Faith based recommender system to create better recommendations
[11]	Probabilistic factor analysis	Proposed model determines whether the news is important news by calculating the joint probability of the news	The results show that their method can productively extract important news from raw news data gathered from various online portals

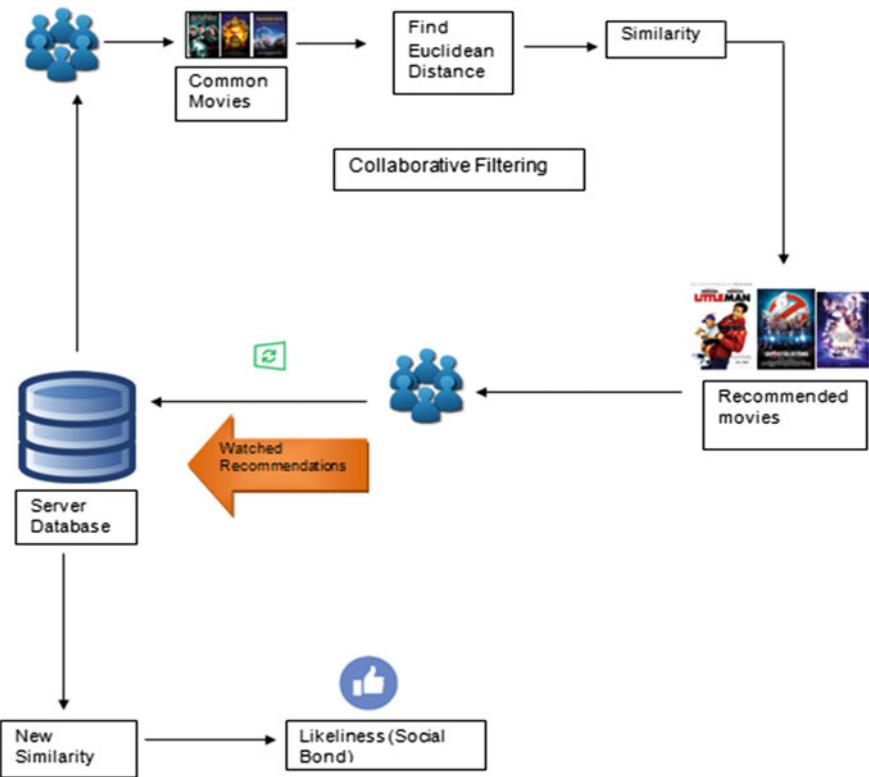


Fig. 2 Flow model of the proposed recommendation system

of our model and algorithms to determine the different objectives of our research. We list these objectives below and further describe taking an implementation scenario. The algorithms for each objective are also mentioned in next section.

- Determining similarity between user ‘a’ and ‘b’ using movie ratings provided by these users.
- Determining likeliness between two users (Fig. 2).

4 Proposed Model and Results

In our proposed model, we have designed and implemented modular algorithms for achieving the objectives mentioned earlier. These algorithms are presented below:

Algorithm 1: Determining similarity between users ‘ u_a ’ and ‘ u_b ’ using movie ratings provided by these users.

INPUT: Name of user from user set $U\{u_1, \dots, u_n\}$ for whom we want to find the similarity

OUTPUT: Similarity of user from user set $U\{u_1, \dots, u_n\}$ with all other users on the system

PROCEDURE:

Step 1: Determine common movies between users, u_a and u_b from user set, $U\{u_1, \dots, u_n\}$ by comparing the watching history of both the users using a “for” loop, where u_a and $u_b \in$ set of users U.

Step 2: Determine Euclidean distance between ratings of common movies.

$$d(p, q) = \sqrt{(q^1 - p^1)^2 + (q^2 - p^2)^2} \quad (1)$$

where p^i and q^i are movie ratings and $d(p, q)$ is Euclidean distance between ratings of common movies.

Step 3: Determine similarity between users, u_a and u_b , using the following Eq. (2).

$$\text{Similarity } S = \frac{1}{2 + d(p, q)} \quad (2)$$

Algorithm 2: Determining likeliness between two users.

INPUT: Name of user u_a .

OUTPUT: Likeliness between user u_a and all other users in user set $U\{u_1, \dots, u_n\}$.

PROCEDURE:

Step 1: DetermineSimilarity S_1 of user u_a with user u_b before user u_a have watched the recommended movies using Algorithm 1.

Step 2: Determine Similarity S_2 of user u_a with other user u_b after user u_a have watched the recommended movies using Algorithm 1.

Step 3: Determine likeliness among users' u_a and u_b using Eq. (3)

$$\text{Likeliness } (u_a, u_b) = |S_1 - S_2| \quad (3)$$

After implementing our model, we have obtained all the results fulfilling our objectives and as per our algorithms. Our model is novel as it shows relationship between users on OTT platforms and also uses that relationship to improve recommendations. The biggest strength of our proposed model is that it can tell if there is increase or decrease in likeliness between two users, after the user has watched the recommended movies. Using this we can determine if two users like each other or not.

These results are presented here as screenshot of the executing browser window. Figure 3 shows similarity, S_1 of user named ‘mohd saquib’ with other users on the system.

Figure 4 shows recommendation of movies suggested by the model for user named ‘mohd saquib’ based on similarity S_1 .

Figure 5 shows similarity S_2 after user named ‘mohd saquib’ has watched the recommended movies.

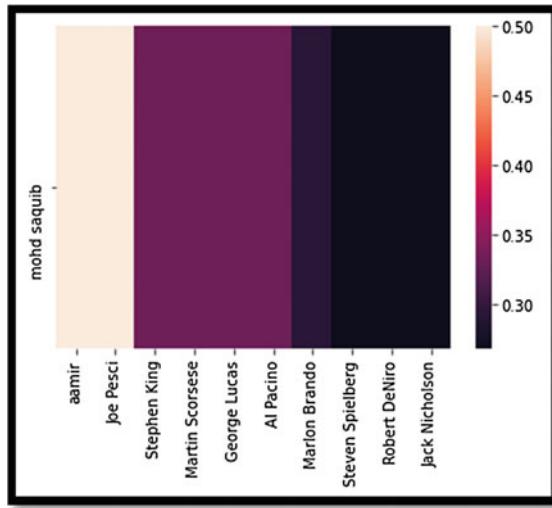


Fig. 3 Similarity, S_1 between selected user 'mohd saquib' and other users



Fig. 4 Movies recommended by the system to the user based on similarity

Fig. 5 Similarity S_2 between users after they have watched the recommended movies

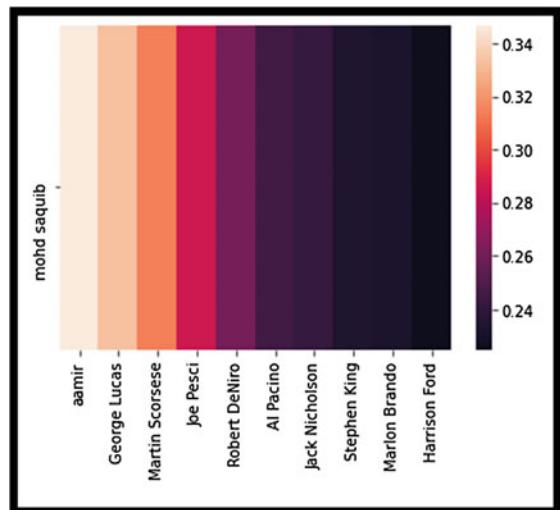
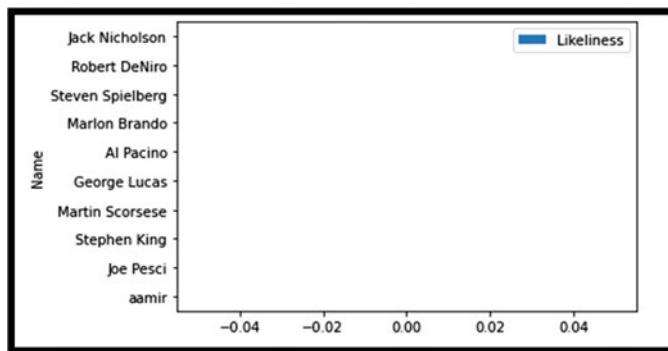
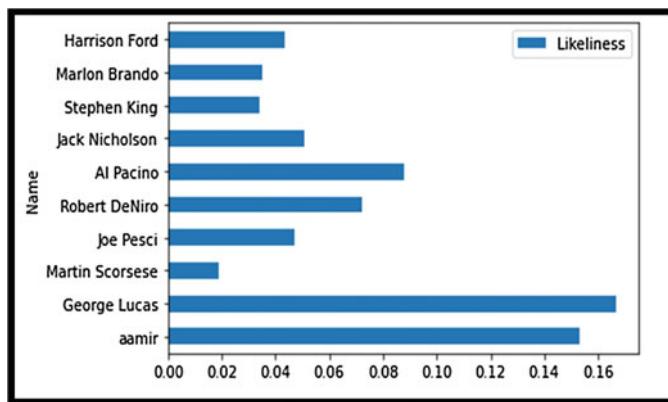


Figure 6a show likeliness between user named '*mohd saquib*' and all other users before the user named '*mohd saquib*' watched the recommended movies and Fig. 6b shows likeliness after selected user named '*mohd saquib*' have watched the recommended movies shown in Fig. 4. By analyzing the above figures (Fig. 6a, b) we can deduce that there is an increase in likeliness between the selected user named '*mohd saquib*' and all the other users on the system just after the selected user watches the movies that were recommended to him by the system.



(a). Likeliness between selected user and other users before the user watched the recommended movies



(b). Likeliness between selected user('mohdsaquib) and other users after the user('mohdsaquib') have watched the recommended movies

Fig. 6 a Likeliness between selected user and other users before the user watched the recommended movies **b** Likeliness between selected user('mohdsaquib) and other users after the user('mohdsaquib') have watched the recommended movies

5 Conclusion

Generally, Resemblance-based techniques are admired because they are automatic, convenient, and easy to include new ratings compared to other methods. Nonetheless, these techniques cause some issues that may influence the precision of the suggested outcomes. We propose social networking among users on OTT platforms to tame the problems and it has been observed that likeliness (social bond) can be helpful in the betterment of recommendation systems. We propose a hybrid approach that can adapt to factors such as user preferences and likeliness (social bond) to make better recommendations. In our proposed recommendation system when a user on the system watches a movie suggested by the system then likeliness (social bond) between the users on the system decreases or increases accordingly. Implementing our model in any OTT platform would upgrade the recommendations and also improve the user experience. With the help of our framework, OTT platforms would be able to improve user experience and also reduce friction between OTT Platform and the target audience.

References

1. Rocca B (2019) Introduction to recommender systems—Towards data science. Medium, June 12. <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>
2. Bhatt B, Patel PJ, Gaudani H (2014) A review paper on machine learning based recommendation system. Int J Engineering Development Res (IJEDR) 2(4):3955–3961
3. Isinkaye F, Folajimi Y, Ojokoh B (2015) Recommendation systems: Principles, methods and evaluation. Egyptian Informatics J 16(3):261–273. <https://doi.org/10.1016/j.eij.2015.06.005>
4. What is a recommendation engine and how does it work. (2020). Express analytics, November 20 <https://expressanalytics.com/blog/what-is-a-recommendation-engine-and-how-does-it-work/>
5. Mall R (2019) Recommender system—Towards data science. Medium, January 10. <https://towardsdatascience.com/recommender-system-a1e4595fc0f0>
6. Chua R (2019). A simple way to explain the Recommendation Engine in AI. Medium, June 29. <https://medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-d1a609f59d97>
7. Golbeck J (2009) Trust and nuanced profile similarity in online social networks. Trans Web 3(4):1–33. <https://doi.org/10.1145/1594173.1594174>
8. Lee WP, Ma CY (2016) Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks. Knowl-Based Syst 106:125–134. <https://doi.org/10.1016/j.knosys.2016.05.037>
9. Mao M, Lu J, Zhang G, Zhang J (2017) Multirelational social recommendations via multigraph ranking. IEEE Trans Cybernetics 47(12):4049–4061. <https://doi.org/10.1109/tcyb.2016.2595620>
10. Golbeck J, Hendler J (2006) FilmTrust: Movie recommendations using trust in web-based social networks. CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference. <https://doi.org/10.1109/ccnc.2006.1593032>
11. Xia Z, Xu S, Liu N, Zhao Z (2014) Hot News Recommendation system from heterogeneous websites based on Bayesian model. Scientific World J 2014:1–8. <https://doi.org/10.1155/2014/734351>

Plant Leaf Disease Identification and Prescription Suggestion Using Deep Learning



P. Y. V. N. Dileep Kumar, Purnima Singh, Sagar Pande, and Aditya Khamparia

Abstract Diseases are quite common for any crops there are lots of diseases occurs for plants according to different seasons. And the farmers are applying different pesticides to their plants without knowing what disease it is. Every disease is dangerous based on its intensity on the crop so early identification of those diseases would be helpful to save the crop. This research paper is destined to perform efficiently and effectively with the help of computer assistance. We are using Deep learning techniques to classify images according to their diseases and built a front-end application to perform the operations directly by giving an image to identify the disease and pesticide prescriptions. In this research paper, we used two different datasets belongs to two different crops as inputs for the deep learning algorithms. For this process, we used popular deep learning algorithms like CNN and ANN to classify the images of the diseased leaves, by changing layers and filters sizes for the algorithms to check their efficiency. And we developed a user interface to predict diseases, the proposed method has shown improved and stable results.

Keywords Convolutional neural networks · Artificial neural networks (ANN) · Leaf disease · Django · Image processing

1 Introduction

Several Countries in this world are depending on agriculture as the main source of their economy, but due to some natural disasters and the fungal diseases to the plants happening regularly there was a loss for both the country's economy and the farmers too. Continents like Asia, Africa are leading their economy with agriculture as backbone to their economies and several other countries are also topped in the rankings of

P. Y. V. N. D. Kumar · P. Singh · S. Pande (✉)

School of Computer Science Engineering, Lovely Professional University, Phagwara, Punjab, India

A. Khamparia

Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India

world's largest agriculture product producers like China, America, Russia, France, etc., so countries like these need to take care of their agriculture developing unless the future generations need to suffer. So, we need to take care of agriculture [1].

In this paper, we proposed a system which uses artificial neural networks as the main neural network architecture to classify the diseased and healthy leaves of the plants and identify the correct disease we tried convolution neural networks also to classify the images of both the plants as 3-layered CNN and 4-layered CNN but for both the crops ANN gave a slight edge over both the CNN layers the main agenda behind this project is to prove that ANN can give better accuracy than CNN in many cases, but we have to build ANN by changing the number of filters in each layer and number of output layers by taking the data overfitting and data under fitting [2]. So, we created an ANN architecture here by taking all these considerations into mind and didn't use any of the existing ANN architectures same for the CNN's to build the architecture by changing the filters and number of hidden layers in between the input and output layers. At starting we were getting just 45% accuracy for ANN and 58% accuracy for CNN but once I modified those neural network architectures, I got the accuracy as 100% for ANN on rice image whereas 87% for corn images, and for CNN I got nearly 85% for rice and 84% for corn. So, we got better accuracy for ANN in both the cases by just changing some parameters of the neural network architectures, as we performed comparisons with the LSTM (RNN) also but it didn't give better accuracy even after changing different parameters so we didn't use it. And finally, we created a front-end portal to access the services of disease predictions with the help of Django framework where you have to select which particular plant you're going to predict the disease and have to choose a file then upload it by clicking the assigned button for prediction then in the backend with the help of already generated ANN models, the prediction of image will be performed and you'll get to know the disease and what are pesticides you need to apply to save your plants from that disease in the images of the pesticides and the text format also shown to you in the web page by routing to the prediction page then you can note down those prescriptions to buy them from your pesticide seller.

2 Related Work

Sammy v. militante (2019) has published a research paper on various plant disease detection using CNN only [3]. As per the paper, he got good Accuracies for both testing and training about 95% accuracy he got combined. But for each and every disease his model accuracy varies. Melike sardogan [4] has published a research paper on tomato plant leaf disease detection using CNN and Linear vector quantization (LVQ) [4]. As per the paper, he got the Accuracies different for each category, the accuracy varies from 80 to 90%. Arhit strikew [5] has published a research paper on grape plant leaf disease detection and diagnosing using ARTMAP and Co-occurrence matrix he got 97% accuracy [5]. But for different gray levels, his model accuracy also varies continuously. Puskara sharma [6] has published a research paper on plant

leaf disease identification but in the paper, they done it for multi plant classification and used knn, logistic regression, svm, cnn, where he got 98% for cnn and remaining all given less than 60% accuracy [6]. Xulan guang [7] has published a research paper on plant leaf disease detection by combining four cnn models on 10 plant species where his model achieved an accuracy of 87%. he used cnn models like Inception, Resnet, Inception Resnet, and Densenet [7].

2.1 Artificial Neural Network

Artificial Neural Network is one of the neural networks that are available to perform image classification. Ann's perform well on non-linear and complex data as most of the problems in the world are non-linear and complex. Ann works similar to a human brain it is developed to perform like how brain to takes inputs and processes the data give outputs. The typical Artificial Neural Network architecture will look like same as other neural network architectures but the processing of data is different from other architectures as shown in Fig. 1 [8]:

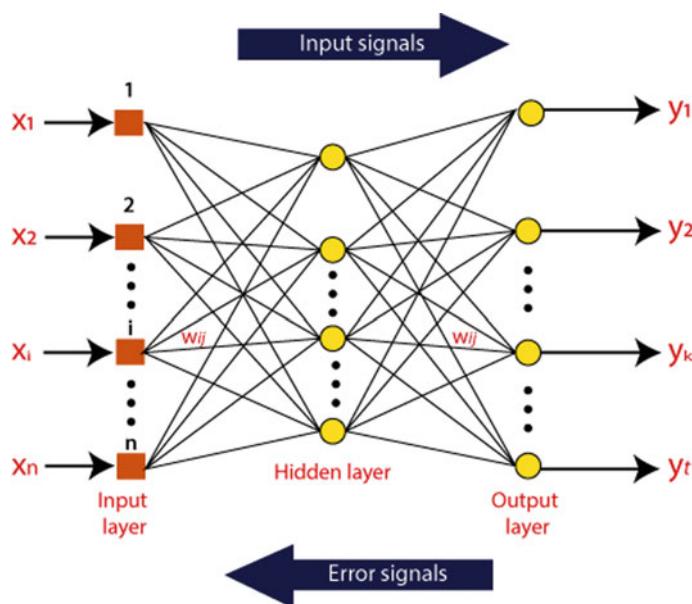


Fig. 1 Artificial neural network. *Source* javatpoint

2.2 Convolutional Neural Network

Convolutional Neural Network is a kind of Artificial Neural Network designed to classify the images based on their pixel processing. CNN's consists of three layers, Convolutional Layers, Pooling layers, fully connected layer. CNN learns the important features of the given images automatically with the help of the layers stacked in its architecture without human support [9]. The typical CNN architecture looks like as Fig. 2.

2.2.1 Convolutional Layers

Convolutional layers are used to store the data of the previous kernel output of the given matrix shape of the kernel which of pixel data processing of an image. Feature mapping of an image will be done in this layer. The process inside a Convolutional layer is depicted in Fig. 3.

2.2.2 Pooling Layers

The pooling layers are used to customize the feature maps generated by the convolutional layers to boost the computational performance. Pooling layers operates on each feature map independently [10]. There are different pooling techniques like Max pooling, Min pooling, Average pooling, but mostly used one is max pooling the pooling layer which is depicted in Fig. 4.

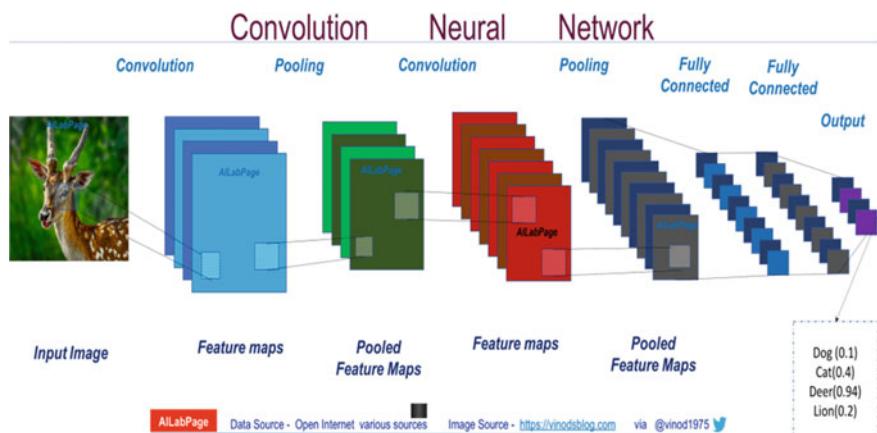


Fig. 2 Convolutional neural network. *Source* vinodsblog.com

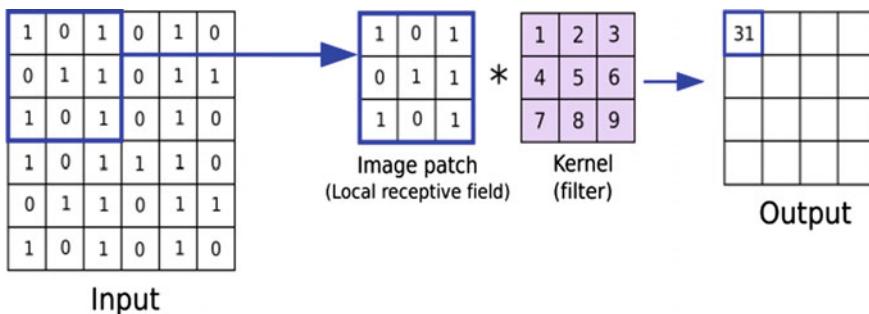


Fig. 3 Convolutional layer inside a CNN. *Source* anhreynolds.com



Fig. 4 Pooling layer operations. *Source* Geeks for geeks

2.2.3 Activation Layer

The most commonly used activation layer is ReLU (rectified linear unit) which is used in each convolutional layer. ReLU don't activate all the neurons at a time and speeds up the training process and the computation of ReLU is also very easy as it converts the negative values to 0 [11].

2.2.4 Fully Connected Layer

Fully connected layers are used to execute the data extracted from all the previous layers to form the final output. We are using two fully connected layers here in our CNN architecture [12].

3 Methodology

3.1 Image Acquisition

The dataset has been collected from Kaggle data repository which consists of nearly 4500 images of two plants Rice and Corn. Rice consists of 3 features whereas corn consists of 4-features.

3.2 Data Preprocessing

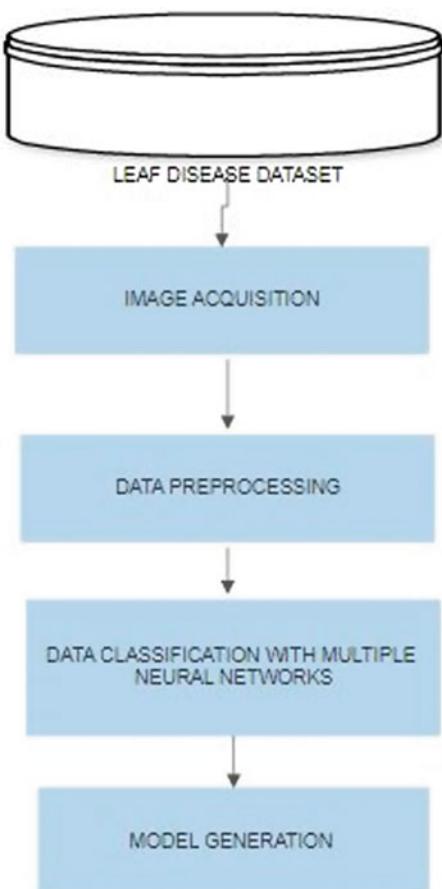
In the data preprocessing all the images in the dataset are transformed to a shape of 100*100(width and height) for a balanced data shape.

3.3 Data Classification with Multiple Neural Networks

Here in this step the classification of images has been done with the help of multiple neural networks which are ANN, 3-layered CNN, and 4-layered CNN to compare which neural network is giving better accuracy with slight parameters change in each neural network like filters in each layer. Figures 5 and 6 narrates the Backend Process & Front-End Process respectively.

3.4 Model Generation

After analyzing all the metrics for each neural network, a model will be generated for the best accuracy neural network this will help us not to run entire backend part again and again. So once the local server started running on the main screen, we will get options to upload images then after successfully uploading image then the model gets invoked to find the leaf disease, and after finding the disease it will suggest some pesticides for the crop to get rid of that disease. So, one can loop it like this as many times as the user wants. And the front-end application which is developed to execute the backend models has been shown below in Figs. 7, 8 and 9. As we earlier mentioned in the proposed system, we used three different neural network architectures modified by our own to get the maximum accuracy. As the whole process is divided into two parts the backend and the front end. Where data preprocessing and date feeding to different neural networks and generating models for each neural network to measure their performance, all this work will be done in the backend process. Model Loading and image choosing predicting the disease and getting the pesticide suggestions on the website will be done in the front-end part.

Fig. 5 Backend process

- Image processing
- Data feeding to neural networks
- Model generation
- Prediction and analysis
- Feeding Models to user interface
- Disease prediction and prescription suggestion

4 Experiment and Result Analysis

In this work, a dataset of totally 4500 images for total training and testing process has been used. The dataset is split according to 8:2 ratio of training and testing split has been done to fit into neural network. This dataset consists of 2 types of plant leaf diseases and of 7 features totally, i.e., 4 different diseases of corn and 3 different

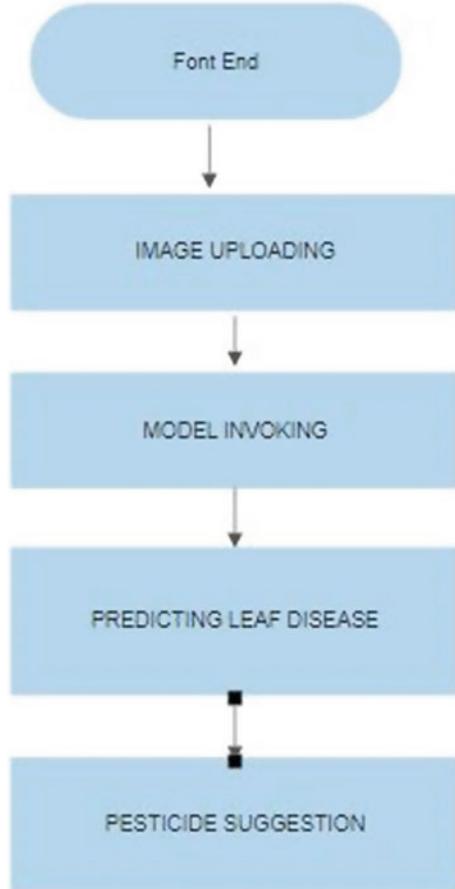


Fig. 6 Front-end process

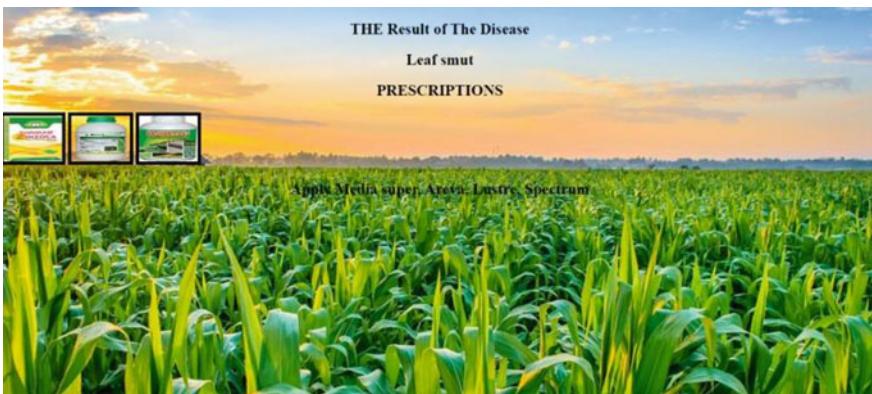


Fig. 7 Front-end application



Fig. 8 Rice crop disease prediction and pesticide suggestion

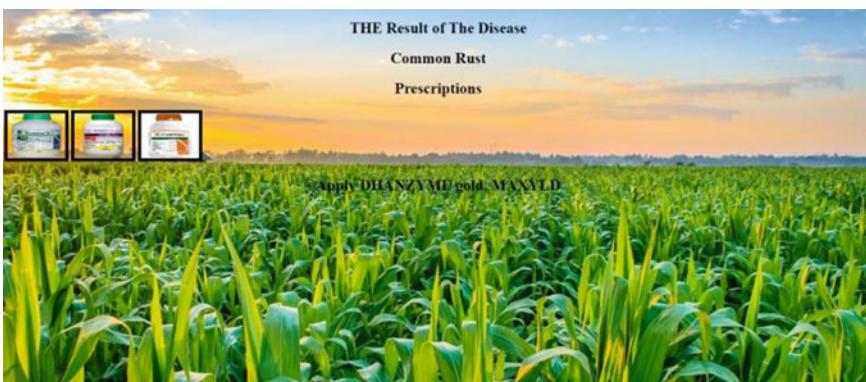


Fig. 9 Corn crop disease prediction and pesticide suggestion

diseases of rice. All the neural network architecture code has been developed in Python programming. And grabbed some images from field to test but the accuracy was quite good, and we trained all the neural networks with 100 epochs. Later model generation has been done and saved to system and developed a front-end application using Django to deploy the deep learning model developed to make predictions and the pesticide prescriptions for the farmers. The backend part has been executed in Google Colab and front end has been developed and run on the HP-pavilion G4 Pentium model laptop of I5 processor and 4gb ram.

A 100% accuracy has been achieved at 100 epochs on rice leaf diseases using Ann and 85% accuracy for corn leaf diseases which is highest among the three neural networks. Down here a series of six graphs has been attached which shows the testing and training accuracy and loss of a neural network on the particular crop as we trained and tested the dataset on three different neural networks, we can see we will get six different graphs as each neural network has two graphs so those graphs

have been shown in Figs. 10, 11, 12, 13, 14, and 15. In each and every crop we can clearly see the bars of accuracy and loss of the particular models referenced with their colors to easily find what their identity is In Table 1, accuracy of the neural networks according to the crops is provided.

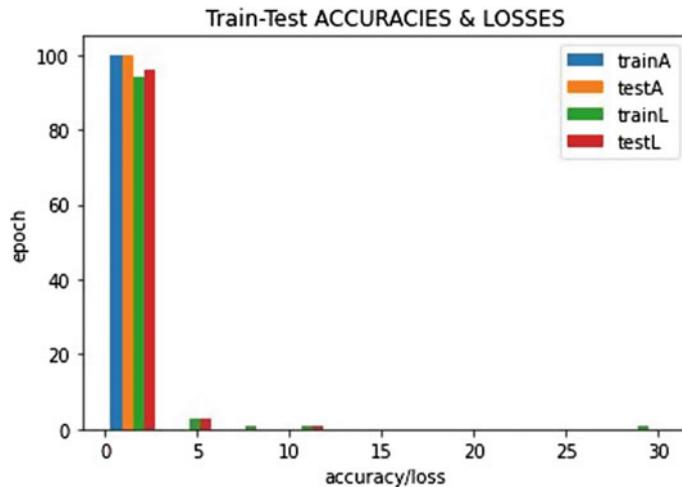


Fig. 10 Train and test accuracy/loss of ANN rice

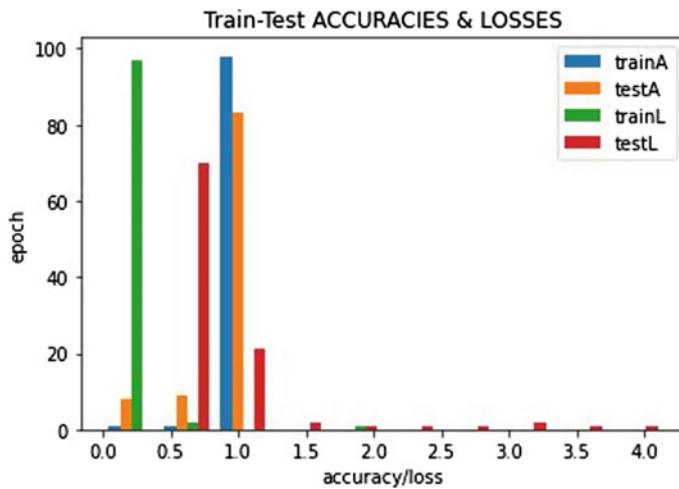


Fig. 11 Train and test accuracy/loss of 3-layered CNN rice

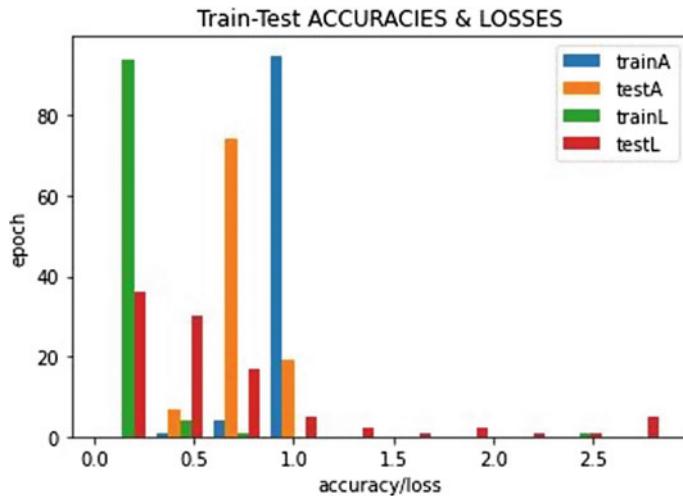


Fig. 12 Train and test accuracy/loss of 4-layered CNN rice

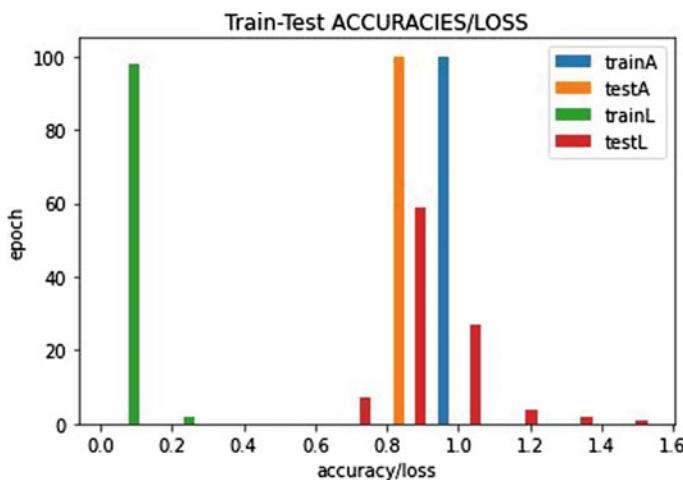


Fig. 13 Train and test accuracy/loss of ANN corn

5 Conclusion

Not today, not tomorrow even after so many years humans and living beings have to rely on food to survive and the food comes from agriculture, so we have to give utmost priority to it in order to save humans. For this, we need to protect our plants from variety of diseases that affects them and we have to increase the fertility of our lands to give good yield. This paper has achieved its goal by getting 100%

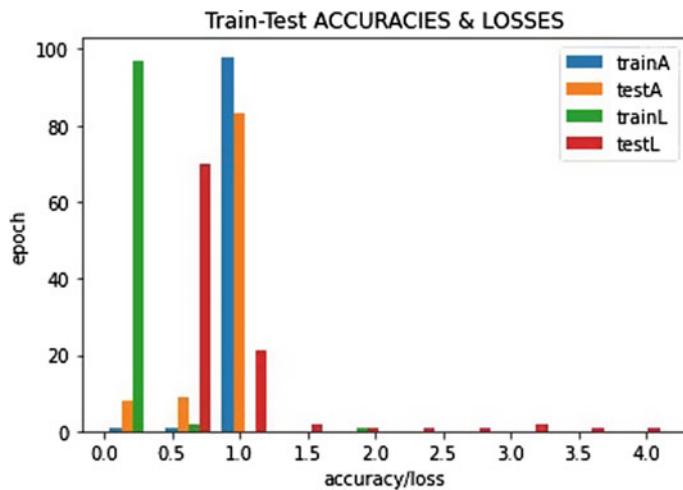


Fig. 14 Train and test accuracy/loss of 3-layered CNN corn

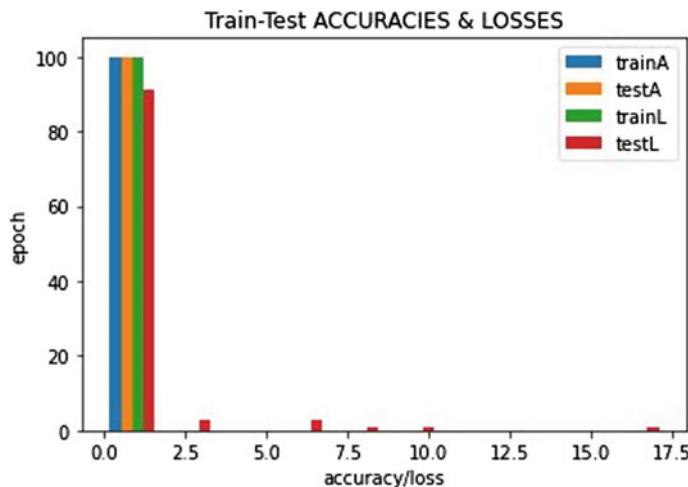


Fig. 15 Train and test accuracy/loss of 4-layered CNN corn

Table 1 Accuracy of each neural network according to crop

Neural networks and their accuracies

S. no	Neural network	Rice (%)	Corn (%)
1	CNN 3-layered	83.33	84.34
2	CNN 4-layered	83.33	83.73
3	ANN 3-layered	100	84.94

accuracy for one plant and 85% accuracy for another plant disease using Artificial neural networks rather than convolutional neural networks. We can use real time images in our plants also to predict the disease with better accuracy. For future work, several other plant disease identification will be included, and sending prescriptions directly to user's whatsapp for ease of using the provided user interface. Getting 100% accuracy means it will be really benefiting for the farmers to predict diseases in real time also effectively. In the future, we are going to develop the front end of the project most effectively like sending the prescriptions directly to the user's mail/whatsapp and adding more plants data to explore wide range of diseases by the farmers/users. And applying multiple techniques and multiple neural networks to make it more effective.

References

1. Militante SV, Gerardo BD, Dionisio NV (2019) Plant leaf detection and disease recognition using deep learning. In 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp 579–582. <https://doi.org/10.1109/ECICE47484.2019.8942686>
2. Marzougui F, Elleuch M, Kherallah M (2020) A deep CNN approach for plant disease detection. 21st International Arab Conference on Information Technology (ACIT), pp 1–6, <https://doi.org/10.1109/ACIT50332.2020.9300072>
3. Sardogan M, Tuncer A, Ozen Y (2018) Plant leaf disease detection and classification based on CNN with LVQ algorithm. 2018 3rd International Conference on Computer Science and Engineering (UBMK), pp 382–385. <https://doi.org/10.1109/UBMK.2018.8566635>
4. Akhilesh SDM, Kumar SA, et al (2019) Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight. 2019 International Conference on Communication and Signal Processing (ICCSP), pp 0645–0649. <https://doi.org/10.1109/ICCSP.2019.8698007>
5. Mugithe PK, Mudunuri RV, Rajasekar B, Karthikeyan S (2020) Image processing technique for automatic detection of plant diseases and alerting system in agricultural farms. Int Conf Communication Signal Processing (ICCSP) 2020:1603–1607. <https://doi.org/10.1109/ICCSP4856.2020.9182065>
6. Deepa RN, Shetty C (2021) A machine learning technique for identification of plant diseases in leaves. 2021 6th International Conference on Inventive Computation Technologies (ICICT), pp 481–484. <https://doi.org/10.1109/ICICT50816.2021.9358797>
7. Sharma P, Hans P, Gupta SC (2020) Classification of plant leaf diseases using machine learning and image preprocessing techniques. 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp 480–484. <https://doi.org/10.1109/Confluence47617.2020.9057899>
8. Vaishnnavi MP, Devi KS, Srinivasan P, Jothi GAP (2019) Detection and classification of Groundnut leaf diseases using KNN classifier. 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), pp 1–5. <https://doi.org/10.1109/ICSCAN.2019.8878733>
9. Classification of plant leaf diseases: A deep learning method (2019) International J Innovative Technology Exploring Engineering 9(1):5195–5197. <https://doi.org/10.35940/ijitee.a9230.119119>
10. Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D (2016) Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intell Neurosci 2016:1–11. <https://doi.org/10.1155/2016/3289801>

11. Liu J, Wang X (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17(1). <https://doi.org/10.1186/s13007-021-00722-9>
12. Bapat A, Sabut S, Vizhi K (2020) Plant leaf disease detection using deep learning. *Int J Adv Science Technology* 29(6):3599–3605. Retrieved from <http://sersc.org/journals/index.php/IJAST/article/view/14162>

Machine Learning Based Data Quality Model for COVID-19 Related Big Data



**Pranav Vigneshwar Kumar, Ankush Chandrashekhar,
and K. Chandrasekaran**

Abstract Big Data is being used in various aspects of technology. The quality of the data being used is essential and needs to be accurate, reliable, and free of defects. The difficulty in improving the quality of big data can be overcome by leveraging computing resources and advanced techniques. In this paper, we propose a solution that utilizes a machine learning (ML) model combined with a data quality model to improve the quality of data. An auto encoder neural network that detects the anomalies in the data is used as the Machine Learning model. This is followed by using the data quality model to ensure the data meets appropriate data quality characteristics. The results obtained from our solution show that the quality of data can be improved efficiently and effortlessly which in turn aids researchers to achieve better results.

Keywords Anomaly detection · COVID-19 · Data encoders · Data quality · Machine learning

1 Introduction

The main aim of this paper is to create a simple yet effective approach to improve data quality for COVID-19 big data. A simple approach is an important goal as it is necessary for users with minimal knowledge to be able to effectively use this system. The solution proposed leverages the power of machine learning. This will help improve data quality by detecting outliers and anomalies in the data which will

P. V. Kumar (✉) · A. Chandrashekhar · K. Chandrasekaran

Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, India

e-mail: pranavnkps.181co239@nitk.edu.in

A. Chandrashekhar

e-mail: ankushchandra.181co206@nitk.edu.in

K. Chandrasekaran

e-mail: kchnitk@ieee.org

help in our goal of achieving good quality data. This data outputted by the machine learning component is then passed to a data quality model to further improve the quality.

The size and amount of data used in various fields have been increasing rapidly. As long as this data used is not properly prepared and processed, its use is limited and can hinder the development of solutions that are, no matter the size of the data used, we cannot ignore the aspect of data quality. Performing this data quality check of detecting and removing anomalous behaviors cannot be done manually by operators as it is a very time-consuming task. With the help of Machine Learning, we can automate this process reducing its complexity and allowing humans to do more useful non-repetitive work.

Types and properties of anomalies can vary from data to data and thus different strategies and anomaly detection requirements need to be used. Thus, anomaly detectors need to be flexible with the anomaly detection logic and work with varying data. To this end, we propose a deep learning [1] model which will be trained with pre-existing COVID-19 data taking different factors as its parameters. The model which we will be using is a neural network which is the best for data quality and anomaly detection. This allows us to take advantage of the flexibility of neural networks.

Following that, we establish a Data Quality model which we will be a modification of the 3As Data Quality Model [2]. The aim of the modification is to establish data quality conditions and metrics specific to COVID-19 data and thus help improve the quality of the data.

The remainder of the paper is organized as follows. Section 2 contains an overview of relevant research papers. Section 3 concentrates on the proposed model and its detailed description. It includes details about the different stages of the data quality model. Sections 4 and 5 contain all the details and results of the experimental work carried out. Finally, Sect. 6 includes the conclusions drawn and the scope for future work.

2 Literature Review

The study by Merino et al. [2] focuses on a new model called “3As Data Quality Model” which proposes three Data Quality characteristics for assessing the levels of Data Quality-in-Use in Big Data projects: Contextual Adequacy, Operational Adequacy, and Temporal Adequacy. The model can be integrated into any sort of Big Data project, as it is independent of any pre-conditions or technologies. The paper shows the way to use the model with a working example. The model accomplishes every challenge related to Data Quality program aimed at Big Data.

Dai et al. [3] proposed using both deep learning models and statistical models to achieve better data quality by eliminating outliers in the data. This paper uses Arkansas State in the US and its employee salary data to test the results and demonstrate outlier detection. They explore multiple deep learning models and come to the conclusion that the Back Propagation Model is best suited for this purpose of

data quality. On the Statistical model side, they use probabilistic models for outlier detection and quality control is done with the help of statistical process control.

The study by Valerie et al. [4] shows that data quality is an important aspect of machine learning tools. This study also tests some preliminary methods which incorporate data quality assessments, thus creating more robust and useful algorithms. This study uses the fundamentals of Bayesian networks and shows how quickly its complexity grows as more nodes and dependencies are added. To achieve data quality in this paper they focus on accuracy, completeness, and believability, and to determine what quality factors are most important when assessing the quality of our data.

3 Proposed Model

3.1 Approach

The Machine Learning based Data Quality Model for COVID-19 Big Data is intended to aid research efforts and provide a comprehensive way to measure Data Quality. This project intends to establish a working and efficient Machine Learning/ Deep Learning model and also modify the “3As Data Quality Model” [2] according to the needs of COVID-19 related Big Data. This Machine Learning model is then integrated to work in tandem with the modified data quality model to improve quality and get better results on the COVID-19 data.

The system intends to make it easy for researchers with minimal data science knowledge to use the tool through a well-documented and guided approach. The Machine Learning model will serve as the first step of the data quality check followed by a Data Quality Model.

The system essentially consists of two parts. The first part is the Machine Learning model. This ML model will be the software/implementation component of this project. With the help of the ML model, we intend to perform defect/anomaly detection on the input data. The input data will contain information about patients such as their age, gender, and other characteristics. Based on this data provided, the classifier will essentially pick out the anomalies from the data. These anomalies can then be removed from the dataset to ensure better and more consistent data quality. Another option is to give researchers the opportunity to study the anomalies and gain new and useful insights. These new insights can give researchers the ability to understand the constantly evolving virus.

The second part involves defining a data quality model based on the “3As Data Quality Model” proposed by Merino [2]. This part on its own does not include any software implementations and is more theoretical. Our intention here is to propose a model which takes into account the appropriate data quality metrics and ensures that the data satisfies these metrics.

The quality of data used in various experiments influences the results to a large extent [4]. In recent times, the importance of data quality has been increasing and this fact is especially evident during the pandemic. The quality of data can be the difference between getting valuable results and not. Keeping this in mind, we picked this topic in the hope that we could contribute to it.

3.2 Method

The aim and working for this project are based on two important things which are the anomaly/outlier detection and the data quality rules that need to be adhered to attain high data quality. The support for the above functionalities will be provided as separate sections which work together to achieve the required result. In the following subsections, we look at each of these stages in detail.

Data Preparation. The preparation composite will act as the input for the entire anomaly detection service. All data inputted will first pass through this composite where certain pre-processing is performed. The raw data received is prepared with relatively minimal input from the user. The components present in this section, as shown in Fig. 1, include:

- The Data Modification component primarily works on modifying the input data by re-analyzing it. It makes the data into same input format as the model. Further, the categorical columns are converted into multiple one-hot encoded columns. This step can be performed in many stages side by side to improve efficiency and performance.
- The Merger component takes as input the data from the *Data Modification* component and merges all the received data and removes redundant data to then pass

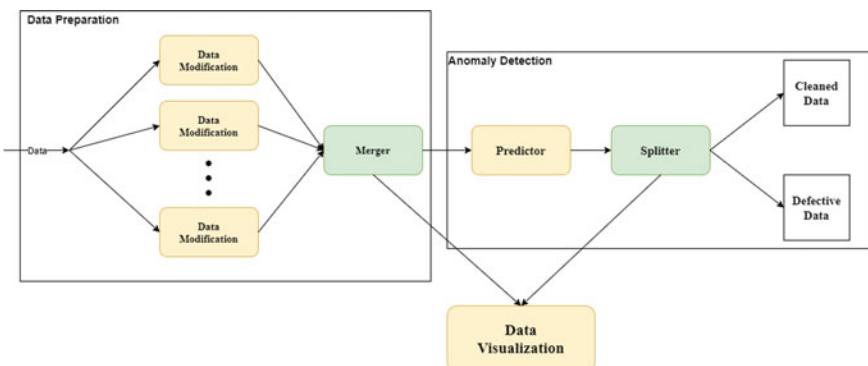


Fig. 1 Working model overview

to the Machine Learning model. Apart from passing the output of this component to the Machine Learning model, it is also passed to the *Data Visualization* Composite for better user understanding.

Anomaly Detection. The Anomaly Detection component will act as the core part of the proposed data quality solution. It is implemented as a machine learning model which is trained to remove the outliers and anomalies from the data. The training of the model undergoes the standard steps such as tweaking the parameters and hyper parameters, fitting the model to the train dataset, checking for the accuracy of the predictions, and then repeating the process until we achieve satisfied results. The structure of this stage is shown in Fig. 1 and each of the stages is explained below:

- The first component would be the Predictor component. Here, the data inputted by the user is passed to the model which has been trained before. The concept of autoencoders is used in the identification of anomalous/defective data points. Autoencoders are a type of artificial neural network used to learn efficient data encodings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise” [5]. Once this is finished, it is passed to the next component in the pipeline.
- The second component would be the Splitter component. Here, using the results of the previous component, we eliminate the anomalous data from the original data thus cleaning the data and improving data quality. This improved data can then be used for further study. We also add the data removed to a separate dataset to allow users to study and gain insights into the anomalies.

Data Visualization. This component of the model is responsible for displaying and helping users interpret the results given by the Machine Learning model. This can help users understand which properties of the data are leading to anomalies without having to delve into the details of the data. Gaining these insights can help in better data gathering as well. Plotting the data before the data is passed to the predictor model can help the user gain an understanding of the data being used.

Data Quality Model. After the data has been passed through the machine learning model and anomalies have been removed, we use the 3A's Data Quality Model [2] to further analyze the data quality. This stage does not contain any implementation as such but it is a set of rules/characteristics which is required to be fulfilled to increase the data quality further. The main requirement for data quality for Big Data projects is the Adequacy of data to the purposes of the analysis. According to Merriam-Webster dictionary, Adequacy can be defined as ‘the state or ability of being good enough or satisfactory for some need, purpose or requirement’. The 3A's stand for Temporal Adequacy, Contextual Adequacy, and Operational Adequacy [2]. Each of these has different features such that we can achieve different characteristics of data quality. This model is designed to provide a way to obtain the extent to which the data is sound and appropriate from the quality point of view for the intended use.

Contextual Adequacy can be defined as the ability for datasets to be used within the same domain of interest of the analysis independently of any format [2]. In terms of contextual adequacy, the following need to be checked:

- Relevancy and Completeness: It needs to be ensured that the data being used is appropriate for the task at hand. Further, it is also necessary to ensure that the data is complete and the amount of data used is sufficient.
- Accuracy: The data being used should represent useful and practical entities in the context of the research being performed.
- Credibility: Experts in the field of research need to ensure the credibility of the data is good enough.
- Compliant: The data needs to satisfy the required regulations and conditions.

Satisfying the above properties ensures that the data fulfills contextual adequacy which is one of the most important properties required to be checked off.

Temporal Adequacy can be defined as the property of data to be within an appropriate time slot of the time of analysis of the data. This means that the data must not be outdated with respect to the time the analysis [2]. In terms of temporal adequacy, the following need to be checked:

- Currentness: It must be ensured that the data have required levels of currentness that is, the data must be similar in age to that of the analysis being performed. This is especially essential in the case of COVID-19 as the virus is evolving constantly.
- Updation: Timely updation of the data must be guaranteed so that the age of the data is relevant.
- Time Consistency: The data must be checked for any unintelligibility related to the represented time.

Satisfying the above properties ensures that the data fulfills temporal adequacy and shows that the data is current, not outdated, and fit to use for analysis.

Operational Adequacy can be defined as the property of the data that ensures the necessary and sufficient resources are available to perform the analysis. It refers to “the extent to which data can be processed in the intended analysis by an adequate set of technologies without leaving any piece of data outside the analysis” [2]. In terms of operational adequacy, the following need to be checked:

- Availability, Recoverability, and Accessibility: The data being used needs to be easily available, recoverable, and accessible to ensure cost effectiveness.
- Authorization: The data being used needs to be authorized for analysis by the required authorities.
- Efficiency: The data should have an efficient representation to ensure minimum wastage of resources.
- Portability: The data should have certain levels of portability to allow for use on different systems.

Satisfying the above properties ensures that the data fulfills operational adequacy.

Using the data quality model proposed improves the quality of the data which in turn should result in better solutions.

4 Experimental Work

Data related to COVID-19 is generally extremely large and hence the proposed solution takes a deep learning approach to take advantage of its power. The deep learning concept of autoencoders is used as the basis for anomaly detection.

Autoencoders are extremely useful and play an important role in unsupervised learning. They are essentially learning models that attempt to transform the input given to the model to outputs with the minimum distortion possible [5]. The understanding behind using autoencoders is that our data will be encoded into a subspace. This is then followed by decoding the features back. By doing this, we expect the autoencoder to model after and learn the features of the standard data. This standard data will usually have outputs similar to the input when applied. This will not be the case for anomalies as it is abnormal data and hence the outputs will be significantly different from the inputs. This approach also enables the use of unsupervised learning which is important as labeling the anomalies in the training dataset is extremely time consuming, expensive, and requires skilled individuals.

To ensure the best result is obtained, we build and train multiple models and aggregate the scores. This is essentially done to prevent overfitting as unsupervised learning is prone to overfitting. Dropout is also used in the neural network architecture to reduce the effect of overfitting. In our solution, we train three models which differ in the number of layers in the neural network architecture. Detailed description of the architecture is provided later in this section.

4.1 Dataset Used

The dataset used for the training of the model contains patient data of COVID-19 patients which includes patient specific information regarding the patient history and habits. This dataset was released by the Mexican Government (<https://www.gob.mx/salud/documentos/datos-abiertos-152127>). The data is also anonymized to ensure the privacy of the patients. This dataset contains about 566,000 patient records. Features present in this dataset include gender, age, and other health conditions associated with the patients.

The dataset contains a large number of categorical variables. Categorical variables can hinder the performance of the model and it is necessary we transform these variables. We encode these categorical variables using one-hot encoding and creating dummy variables for every category in every categorical feature. Following this, we drop the original columns and keep only the newly created dummy variables.

This section will go through the experimental work performed. The experimental work was carried out on jupyter notebook using Python and multiple python libraries.

4.2 Data Preparation

The first stage involved data preparation. The tasks performed in this stage involve importing the dataset, reshaping the dataset by adding and removing certain columns, etc. Pandas, the python data analysis library, is used primarily during this stage of work. The dataset is imported and stored in the form of a pandas data frame. All further changes are made to the data frame.

The original dataset contains two date features namely, the entry date and date of first symptoms column. To obtain more useful information from these two features, we calculate the number of days taken to admit a patient which is the difference between the entry date and date of first symptoms. Following this, we drop the two original features. On further investigation of the data, it came to light that some of the categorical features had different values for the same category. This is rectified by ensuring each category in those features is represented by one value only.

The last task in this stage is to convert the categorical variables into one-hot encoded features. This is an essential step as categorical features in their original form can hinder performance. Dummy variables are created for the different categories of the categorical feature and the original categorical features are dropped from the data frame.

4.3 Anomaly Detection

This stage deals with the anomaly detection component. The various tasks performed here involve visualizing the data before and after anomaly detection, building and training the neural networks, calculating the cut point, etc. The different libraries used in this stage include scikit-learn, a python library for machine learning, PyOD [6], a comprehensive and scalable python toolkit for detecting outlying objects and matplotlib, a comprehensive python library for creating visualizations.

The first task is to visualize the data before performing the anomaly detection. To perform the visualization, we need to reduce the dimensionality of the dataset and this is done by using the technique of Principal Component Analysis (PCA). PCA is used to reduce the data to two dimensions which allows the plotting of these data points. From Fig. 2, it is clear that there are certain anomalies in the dataset.

The next task is to build and train the three neural networks. The input and output layers in all the architectures have 54 neurons. This is because the number of features in the dataset is 54. Dropout is implemented in every layer of the network to reduce overfitting [7]. These autoencoder models are built with the help of the PyOD library

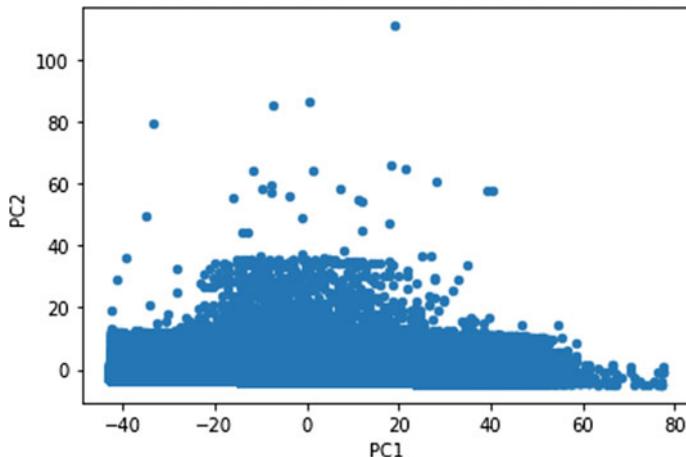


Fig. 2 Data in 2-dimensions

functions with all the hyper parameters except the number of layers and neurons being defined by the library. The three neural network architectures we chose are.

- Model 1: [54, 25, 10, 2, 10, 25, 54]. There are 5 hidden layers with 25, 10, 2, 10 and 25 neurons respectively.
- Model 2: [54, 10, 2, 10, 54]. There are 3 hidden layers with 10, 2 and 10 neurons respectively.
- Model 3: [54, 10, 10, 54]. There are 2 hidden layers, each having 10 neurons.

After building the models, they are fit to the data, this is followed by applying the trained models to predict the anomaly score for each instance of the data. The anomaly score is calculated as the distance between the original data point and output predicted by the model. The anomaly scores of the three different models are combined and the average is calculated.

Figure 3 is a histogram which is plotted to count the frequency by the anomaly score which shows higher scores correspond to lower frequencies indicating the existence of outliers. A cut point or threshold is determined from the histogram to distinguish between the normal and anomalous data. In our case, the cut point chosen was 4.0.

Finally, with the help of the identified cut point, we split the original dataset by removing those data points whose anomaly score is above the cut point and storing it in a separate table. Thus, the end product is two different datasets: one containing the cleaned data after the removal of anomalies and the other containing the anomalies detected. The dataset containing the anomalies can be used to gain valuable insights into the abnormal data which could prove detrimental in research while the cleaned dataset can be used for solutions which require data free from anomalies.

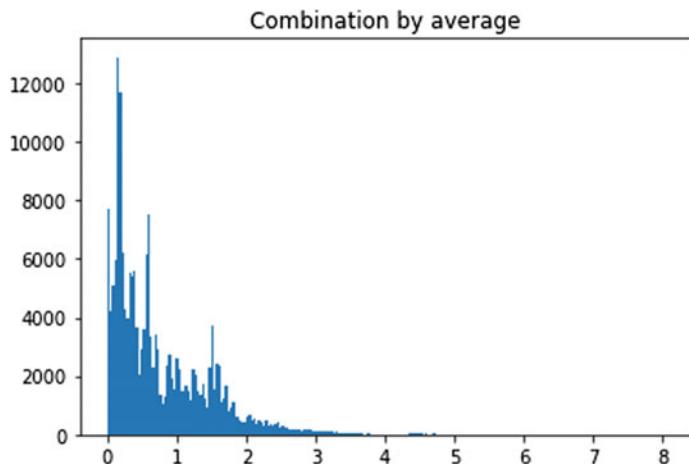


Fig. 3 Frequency distribution of anomaly scores

5 Results

The results of the solution can be seen visually. Since the data is high dimensional, visualizing it is hard. Therefore, for visualization purposes, we reduce the dimensionality of the data to 20 by once again using the technique of Principal Component Analysis (PCA). We assign labels, 0 for non-anomalies and 1 for anomalies, to every data point in the dataset. This is done to plot the anomalies and non-anomalies in different colors. We choose different pairs of features from the dataset with reduced dimensionality to plot. From Fig. 4, we can see that the model is able to detect anomalies (in yellow) comprehensively and thus helps improve the quality of the data.

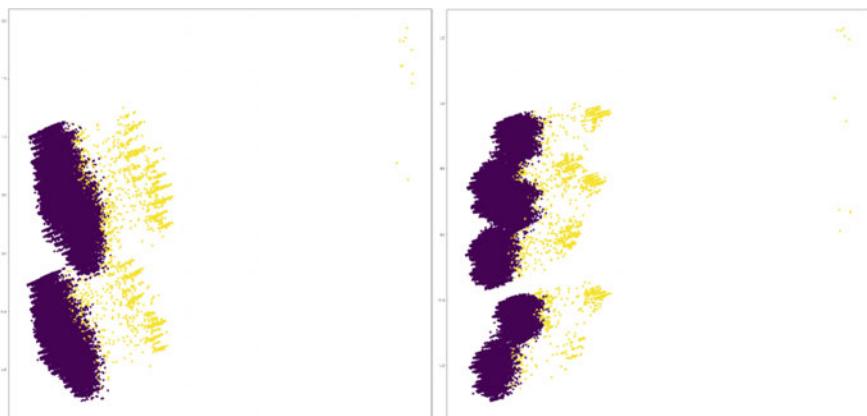


Fig. 4 Plot of anomalies (in yellow) and non-anomalies (in purple)

6 Future Work

In the context of the solution proposed above and data quality, there is much to be done with the machine learning model and the hyper parameters associated with it. These parameters can be tweaked to get better outlier detection results and include other features like checking of record completeness, etc. This solution can also be improved through certain approaches like allowing the model to train even after deployment or allowing real-time integration. In terms of the data quality model, we could include other rules to make the data more specific to the required task and thus more characteristics can be fulfilled with it.

7 Conclusion

The solution discussed in this paper presents a simple yet powerful architecture which takes advantage of the flexibility of machine learning. The results obtained from the experimental work show that the machine learning model, with relatively low human intervention, is effective in detecting anomalies and thus helps improve the data quality. Further, the anomaly detection logic can be modified relatively easily by just re-training the machine learning model until we achieve satisfied results. Thus, the machine learning model in combination with the data quality model is a scalable, flexible, and effective solution to aid researchers in their work.

References

1. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
2. Merino J, Caballero I, Garcia R, Bibiano, Serrano M, Piattini M (2015) A data quality in use model for big data. *Future Generation Comp Sys* 63. <https://doi.org/10.1016/j.future.2015.11.024>
3. Dai W, Yoshigoe K, Parsley W (2018) Improving data quality through deep learning and statistical models. https://doi.org/10.1007/978-3-319-54978-1_66.
4. Sessions V, Valtorta M (2006) The effects of data quality on machine learning algorithms, 485–498
5. Baldi P (2011) Autoencoders, unsupervised learning and deep architectures. *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop*, volume 27 (UTLW'11). JMLR.org, 37–50
6. Zhao Y, Zain N, Zheng L (2019) PyOD: A Python toolbox for scalable outlier detection
7. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958

Paradigm of Handling Data Linked to Cloud Database Impacting Cloud Computing: A Case Study Based on Simulation



Zdzislaw Polkowski and Sambit Kumar Mishra

Abstract It has been observed the significant improvement of implementation of database applications related to information technology. Somehow these applications have now been replaced with specified remote applications linked with committed and appropriate servers along with storage allocation. However, the cloud databases are more appropriate as well as having more commonalities that are provisioned with scalabilities and flexibilities. Actually, the databases in all respect should support the specified functionalities, i.e., atomicity, consistency, isolation as well as consistency. In such situation, it is also necessary to focus on minimization of resources and obtain optimality. The cloud computing concept in real sense helps in enhancement of multiple accessibilities without the prior knowledge of any requisite software or operating system. It is probably an enhanced technology based on internet distributed computing. Its main intention is to develop efficient and powerful computing capabilities with minimal expenditure. In fact, as it is originated from the conventional large-scale distributed computing technology, its focus is towards minimization of processing issues and enhances the services through the service providers. Now considering applications linked to cloud databases, it should be noted that in these cases also the accumulation of relevant data structures as well as query language in structured pattern can be prioritized along with all the components associated with database management system. In true sense, the overall efficiency of handling data can be enhanced. In this paper, it is primarily intended on the evaluation mechanisms of cloud databases and its impact on cloud computing based on simulation approach.

Keywords Query term · Virtualization · Entity integrity · Referential integrity · Response time

Z. Polkowski (✉)
Jan Wyzykowski University, Polkowice, Poland
e-mail: z.polkowski@ujw.pl

S. K. Mishra
Gandhi Institute for Education and Technology, Biju Patnaik University of Technology,
Baniatangi, BhubaneswarRourkela, Odisha, India
e-mail: sambitmishra@gietsbsr.com

1 Introduction

In general, managing data is very essential provisioned with suitable mechanisms on data towards storage and retrieval. In such situation, somehow scheduling mechanisms can be adopted. But considering the application in cloud some challenges may be faced while handling the data retrieved from the application. The reason behind the same may be due to preemptive scheduling and pre-emption of the process for the high priority tasks. As it is known that virtualization, as well as encapsulation, is quite familiar concept in cloud having linked with virtual machine monitors, the non-preemptive scheduling criteria should be adopted to overcome the difficulties. Of course, the cloud applications prove to be efficient and cheap and can be used on computational intensive applications. While prioritizing the integrity constraint of data, it is required to intently focus on the database of all the applications prioritizing more on entity integrity than referential integrity. Though practically it is difficult to maintain consistency on each portion of the databases, but it must be ensured towards maintaining data integrity while associated with cloud databases. Of course, these are encapsulated with distinct servers, not centralized allocated with distinct hubs have complete accessibility over databases. By observing the incremental growth of the databases, it is needed to scale up the databases as per their application and to implement in the cloud with synchronization. The database application with cloud facilitations in true sense is implemented efficiently connected with virtual machines.

The concept of virtualization is the prime feature cloud computing where numerous heterogeneous servers are accumulated in the form of cloud data centers. It is more influenced due to hosting of more virtual machines. In other words, it is more capable of dynamic accumulation and allocation of numerous cores along with processing elements and memory. The allocation of individual virtual machines can be linked with provisioning to handle the requests along with their placement in current execution time. Also monitoring the virtual machines, the minimum usage of physical machines can be accorded consolidating the virtual machines which can further obtain the optimality and focus on minimum energy consumption. In many cases, it has been observed that to enhance the performance of the data centers, virtual machines with structured constraints mechanisms are adopted. Again in the resource management in cloud, somehow specified architectural concept is implemented on generic decision making layer as well as application specific routines which support constraint programming towards better optimization.

Basically, in cloud system, the mechanisms associated with the services can be easily integrated by applying the scientific routines. Therefore it is necessary to maintain the accuracy of data in all respect during deployment of virtual servers. Of course in some situations, the instances of virtual machines are available for the individual virtual server. Accordingly, to manage the instances as well as to obtain the optimality, the adoption of simulation mechanism is one of the approaches. In this paper, the application of particle swarm optimization has been prioritized to schedule the basic criteria as well as to monitor the response time during deployment of virtual servers. It is known that the elasticity property linked to resources is associated with

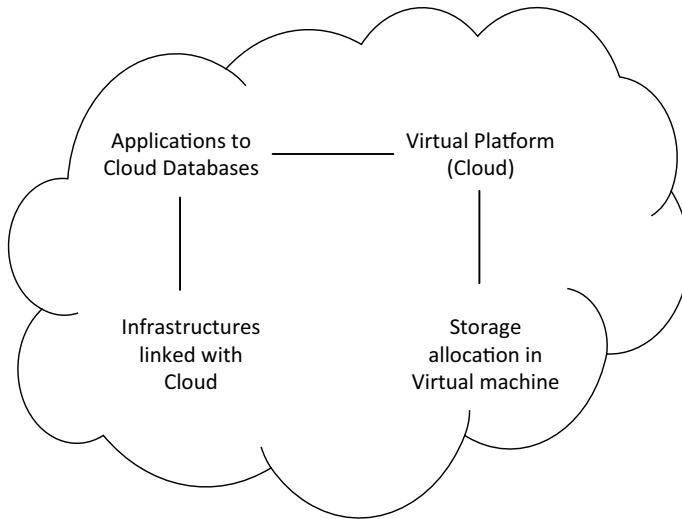


Fig. 1 Application and allocation of data in cloud

cloud computing by which the based on the requirements the scaling can be done as shown in Fig. 1. Considering the cloud data into consideration which is directly associated with the size, category as well as specified ranges, there should be the provision to proper scale the data based on the load in the processing elements. In fact, the search mechanism on data linked to cloud servers can be applied to update individual processing elements along with the cache as shown in flowchart in Fig. 2.

However, as shown in Fig. 3, each deployed sensor can be provisioned with its own clock to maintain and monitor timing signals. Some commonalities linked to the timing scales within the deployed sensors may then be needed to observe any redundancy among data. In fact, each sensor can be operated independently with its own provisioned clock. Therefore, synchronization related to timings is very essential towards enhancement of accessibilities of deployed sensors.

2 Review of Literature

Gutierrez et al. [1] in their work have prioritized IoT technology and towards generation of a large amount of heterogeneous data. In fact, they focused on the storage of these data along with processing abilities in cloud servers. Sometimes, these are required to be queried and updated on-demand. They have also realized the challenges accorded in storage and management of large amounts of IoT data.

Dizdarević et al. [2] during their research observed that the mechanisms associated with cloud computing perhaps enables access to a strong pool of configurable computing provisioned with very marginal management or interaction with service

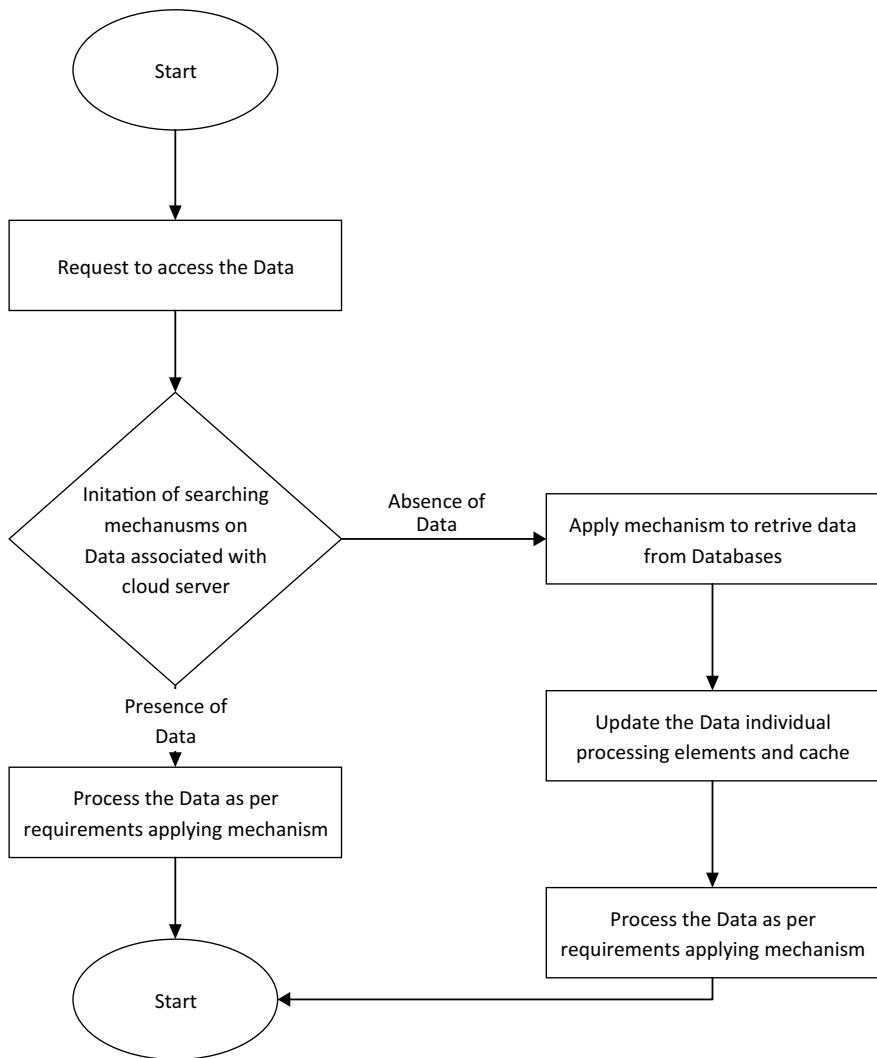


Fig. 2 Flowchart to access and process data linked to virtual server

provider. In fact, the services linked with cloud storage service are the vital portion of the cloud infrastructure which can easily deal with a large amount of data.

Liu et al. [3] in their work focused on the performance requirements related to data size, latency, data rate, reliability along availability. In fact, they observed the main intention IoT technology is to process efficiently the data. In such situation, the servers and data centers are quite responsible towards manipulation as well as storage of the data systematically in the databases.

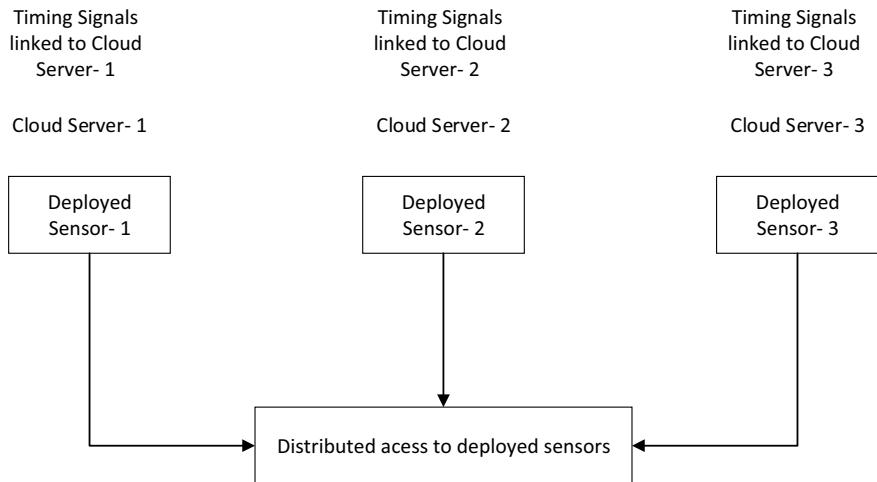


Fig. 3 Timing signals linked to deployed sensors in cloud

Bicevska et al. [4] in their work prioritized the performance of both relational as well as non-relational databases implemented through small scaled as well as large-scaled data. In fact, the performance was linked towards the time of queries along with the size of data.

Rasheed et al. [5] in their work also focused on relational and non-relational databases experimented on data linked to internet of things. They prioritized the experimentation on MYSQL, PostgreSQL, etc., and observed that the performance was better than other structured queries.

Moon et al. [6] in their work observed the dependency of IoT sensor data on the temporal status along with the variables linked to the spatial characteristics differentiated from other homogeneous inputs.

Asiminidis et al. [7] in their work prioritized on the open source MySQL because of its distributive properties like other relational systems. It also supports updates as well as changes in schema migrating the procedures linked to the database and focuses on minimization of application performance.

Tarashev et al. [8] in their work prioritized different strategies associated with the development of specific regions like Kurgan. In fact, the strategies discussed in this application require to support the region. Also, the results linked to this application permit assessing the real level of development of the region.

Cristescu and Vasilev [9] in their work focused on determination of replacement of specialized security mobile applications to open source applications. In such situation, the components are required to be accumulated based on the completeness determining indicator. The parameters of the indicator can be compared with the values already accumulated for the.NET CF3 framework. Also based on the similar mechanisms, the components can be derived also can be further added ensuring the quality and observing the complexities.

Cristescu et al. [10] in their work prioritized on implementation of the Capability Maturity Model Integration in modeling processes. They observed that any type of variations or dissimilarities associated with Capability Maturity Model Integration can be implemented assessing the process maturity which of course based on the scope of application.

Alzubi et al. [11] in their work prioritized on robust cryptosystem which is quite efficient on its applications on internet of things. During their studies, they have also focused on algebraic geometric curves which is in fact linked with their cryptosystem. Their main intention is to protect the information resources and to make it more consistent in all respect provisioning security measures.

Alzubi et al. [12] in their work have focused on the mechanism linked to cost optimized deep learning to observe the performance and security measures related to transmission of industrial IoT data through cloud. In fact, in their work they prioritized on generation of public keys, and using the same they tried to enhance the execution time during transmission of data through secured channels and to minimize the computational cost.

3 Application of Particle Swarm Optimization

It is understood that particle swarm optimization usually optimizes the specified problem to obtain better candidate solutions. In fact, it tries to obtain the solutions of the problem accumulating the population linked with candidate solutions associated with particles. The particles usually move in the search space and generally are linked with local best known positions. In the next iterations, these can be updated towards better positions in the search space to obtain other particles which may be the expectation of the swarms towards optimality. The primary constraint associated with the cloud service provider is the criteria of elasticity linked to cloud as each and every assigned task are required to be permitted which may make a question on the performance of cloud. Of course, the resource allocation mechanisms may not obtain the optimal solution. In such situation, particle swarm optimization technique can help to resolve this type of criticality linked with cloud service providers at cloud data center. Basically, it is used on allocation of resources along with tasks towards minimization of cost of task execution and to gain better efficiency. Generally, each particle can be treated as path of solution and in the initial situation; the particles can be positioned randomly as well as can be updated on the regular interval. During this process, each particle can learn from each feasible pair and the position implied approach can ensure the reasonability of the positions.

3.1 Basic Scheduling Criteria

1. Initialize the particle (Query term) based on specified algorithm
2. Concentrate on fitness parameters linked to each particle.
3. Update position and velocity at each level
4. Obtain the fitness value
5. If (fitness value == Null) then again focus on the fitness parameters

3.2 Steps Towards Focusing on Particles

Step 1: for each particle (represented as query terms) focus and initiate the localized positions along with requisite velocities

- Step 2: while((!eof) || nonattainment of criteria)
 - for each particle, evaluate the fitness parameter
 - Step 3: if the fitness parameter > = initial parameterized_value then set it as p_best_currvalue
 - Step 4: for each particle, obtain the velocity of the particle
 - Step 5: Update the position of the particle

Algorithm

```

Step 1 : Generate initial population
  for i= 1 to size of population
    particle[i].best= current_position
    particle[i].bestfitness_value=current_fitness
Step 2 : Obtain the global best(gbest) parametric value
  gbest= particle.best with fitness parameter(initial)
Step 3 : Determine the velocity based on applied functions with constriction
parameters
  for j= 1 to Max_iteration
    for i= 1 to size of population
       $v_i^{j+1} = c_0 v_i^j + c_1 f_1 X(q_i^j - x_i^j) + c_2 f_2 X(q_i^j - x_i^j)$  /*  $c_0, c_1, c_2$  are the constriction parameters,
       $f_1, f_2$  are the applied functions on the query terms  $q_i$  and parametric constraints,  $x_i$ .
Step 4: Determine the fitness values and set the position of particles
  if current_fitness_value < particle[i].bestfitness_value then
    particle[i].best= current_position
    particle[i].bestfitness_value= current_fitness
    gbest= particle[i].best with lowest fitness

```

In this manuscript, the primary intention is to apply particle swarm optimization techniques to record the response time of queries during the deployment of virtual servers. In such situation, based on the basic scheduling criteria, it is required to concentrate on the fitness parameter of each query term designated as particle. After updating the position of each particle, it is essential to determine the velocity of each particle along with constriction parameters. Accordingly, based on the current fitness

parametric values, the global best parametric values can be obtained which in turn will help to monitor the response time of each query term during the deployment of virtual servers.

4 Experiment Analysis

The deployed servers in cloud in general process the datasets in parallel based on the response time. In fact, the cloud computing exhibits better performance even if in distributed environments. Each and every data center provisioned with multiple processing elements helps during storage allocation as well as the linked computational applications. So to enhance the performance in cloud, it is required to prioritize the scheduling mechanisms and presence of cloud resources. To improve cloud performance, it is extremely important to identify the performance factors having significant impact on cloud performance. Keeping this to consideration, the particle swarm optimization technique is applied in this case to record the response time based on query along with query execution time prioritizing the fitness parameters as well as the position of the particles. The parametric evaluation of query terms in such situation has been processed through MATLAB 13B for further analysis and result generation (Table 1).

Somehow the mechanisms linked to virtualization are pretty essential to enhance the computational approach. Also, performance linked to the associated processes should be measured according to the deployed virtual servers. Also, the adoption of virtualization techniques offers flexibility as well as reliability to the system during implementation. Along with the deployed servers, the operations can be efficiently performed based on the response time on the requisite systems as shown in Fig. 4. In fact, keen observation of the virtualized infrastructure should be kept to obtain better solution. In some situations, the performance linked to the applications on virtual machines based on important criteria, i.e., the virtual machines are responsible towards sharing the hosts along with the resources. But as the application of the virtual machines lies on similar hosts, it is definitely required to monitor the actual utilization stages and the appropriate situation of the system (Table 2).

In many situations, it is observed that the mechanism associated with cloud computing is based on computing linked to utility. Also having the distributed computing capabilities, the tasks are also required to be scheduled efficiently within

Table 1 Number of deployed virtual servers with response time

S. No.	Number of deployed Virtual servers	Response time based on query (ms)
1	19	0.92
2	29	0.53
3	37	0.55
4	40	0.71

Fig. 4 Number of deployed virtual servers with response time

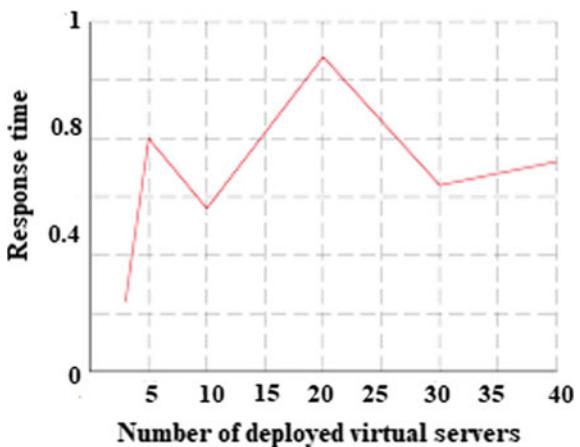
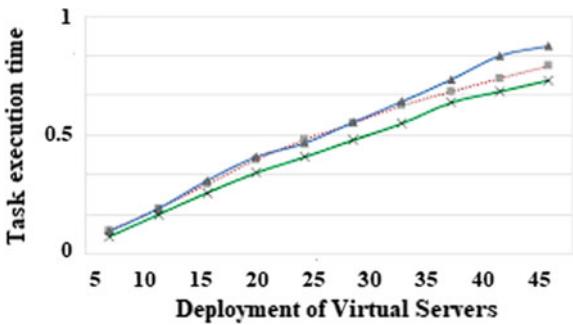


Table 2 Deployment of virtual servers with task execution

S. No.	Deployment of virtual server	Query execution time (ms)
1	19	0.47
2	29	0.56
3	37	0.61
4	40	0.83

the virtual machines. As soon as the tasks are scheduled in the deployed virtual servers, the allocation of the tasks is being scheduled with the corresponding virtual servers prioritizing the execution time as shown in Fig. 5.

Fig. 5 Deployment of virtual servers with task execution



5 Discussion and Future Direction

As the tasks are scheduled in the virtual machines, somehow these are required to be reallocated based on the response on the virtual machines. Of course, the virtualization mechanisms linked with the system should have improved confined parameters along with enhanced rating at every stage of virtual machines. In fact, to achieve better service quality in cloud, the virtualization mechanism has a vital role. Also, the service quality should depend on parameterized estimation. The execution time of tasks can also be used as the evaluation metric at every scheduling level. To observe the optimality, the simulation mechanism can be used for performance evaluation. The simulation along with the user request can be handled towards the actual deployment of cloud tasks. In this application, the particle swarm optimization technique has been applied towards monitoring the response time of each and every query term during the deployment of virtual servers. To do so, fitness parameters of each query term designated as particles have been determined. Accordingly, prioritizing the current fitness parametric values, the global best parametric values are obtained which in turn are helpful to monitor the response time of each query term during the deployment of virtual servers.

6 Conclusion

In general, virtual machines are provisioned with linked computational resources, though the processing elements are different in their execution as well as in task scheduling. In such situation, the response time is more vital during the association with large-scale applications in the virtual platforms. Of course, the data centers have equal responsibilities to maintain dynamism during the allocation of tasks. It is also required to monitor the response time as well as processing time to manage the operations. As measuring performance in cloud is not so easy task, it is therefore required to make prioritization the execution mechanism of each and every data center and its dependency on cloud. The performance of the virtual servers is also required to be measured.

In this work, while estimating the performance of the virtual servers, it is observed that the implementation mechanism either in large-scaled data or the heterogeneous data linked with cloud must be confined towards response as well as task execution time. In fact, there are many instances with specified capabilities towards comparison of performance of the linked databases. Also based on these approaches, there can be a clear distinction on specified workloads on databases, capabilities of resources, and other requirements. Moreover, it is somehow required to prioritize latency as well as the size of the database.

References

1. Gutierrez-Madronal L, La Blunda L, Wagner MF, Medina-Bulo I (2019) Test event generation for a fall-detection IoT system. *IEEE Internet Things J* 6(4):6642–6651
2. Dizdarević J, Carpio F, Jukan A, Masip-Bruin X (2019) A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration. *ACM Comput Surv* 51(6):1–29
3. Liu Y, Hassan KA, Karlsson M, Pang Z, Gong S (2019) A datacentric Internet of Things framework based on azure cloud. *IEEE Access* 7:53839–53858
4. Bicevska Z, Oditis I (Jan. 2017) Towards NoSQL-based data warehouse solutions. *Procedia Comput Sci* 104:104–111
5. Rasheed Y, Qutqut M, Almasalha F (2019) Overview of the current status of NoSQL database. *Int J Comput Sci Netw Secur* 19(4):47–53
6. Moon J, Kum S, Lee S (2019) A heterogeneous IoT data analysis framework with collaboration of edge-cloud computing: focusing on indoor PM10 and PM2.5 status prediction. *Sensors* 19(14):3038
7. Asiminidis C, Kokkinis G, Kontogiannis S (2019) Managing IoT data using relational schema and JSON fields, a comparative study. *IOSR J Comput Eng* 20(6):46–52
8. Tarasyev AM, Vasilev J, Turygina VF, Panchenko AD (2020) Analysis of data on sources of official statistics, development strategy of the Kurgan region. In: Conference: thermophysical basis of energy technologies (TBET 2020)
9. Cristescu MP, Vasilev J (2021) Specialized applications used in the mobile application security implementation process. Organizations and performance in a complex world. https://doi.org/10.1007/978-3-030-50676-6_4
10. Cristescu MP, Vasilev J, Stoyanova M, Stancu A-MR (2019) Capability and maturity. Characteristics used in software reliability engineering modeling. Project: IT Solutions for Business Process Management, Lab: Marian Pompiliu Cristescu's Lab, Land Forces Academy Review 24(4):332–341
11. Alzubi OA, Alzubi JA, Dorgham O, Alsayyed M (2020) Cryptosystem design based on Hermitian curves for IoT security. *J Supercomputing*. <https://doi.org/10.1007/s11227-020-03144-x>
12. Alzubi JA, Manikandan R, Alzubi OA, Qiqieh I, Rahim R, Gupta D, Khanna A (2020) Hashed Needham schroeder industrial IoT based cost optimized deep secured data transmission in cloud. *Measurement*. <https://doi.org/10.1016/j.measurement.2019.107077>.

Provision and Allocation of Large Scaled Data in Virtual Environment: A Case Study with Simulation Approach



Sambit Kumar Mishra and Zdzislaw Polkowski

Abstract The mechanism of processing large scaled data especially in virtual environment is something linked with sharing of resources towards obtaining optimality and maximizing the effectiveness of the shared resources. Perhaps, the dynamical allocation of resources in the present situation is most required towards preservation of data security. Therefore, it is highly essential to prioritize on provision of auto esteemed service on demand, parameterization of the service or application, proper scheduling of resources as well as uninterrupted high speed internet services. Of course, all the facilities should be available while accessing the mechanisms through heterogeneous virtual platforms which are flexible as well as scalable. The system associated with the large scaled data in such situation also can control the mechanism possessing similar descriptions associated with meta-data. The subset of these data can also be confined to specific domains prioritizing the application and may not be embedded. The term large scaled data is basically provisioned with large volume having the ability of high range of processing the data as compared with the response time and of course heterogeneity in nature. Somehow, it can be unstructured or semi structured. But, the more important thing is to analyze the data as well as to extract the information towards enhancement of making decisions. The minimization of complexities linked to the system can be obtained provisioning with scalability as well as reliability. This paper is mainly associated with observing the response time by allocating the databases in the virtual machines and cost of query terms by applying specific ant colony optimization techniques.

Keywords Parameterization · Scalability · Heterogeneity · Pheromones · Query terms

S. K. Mishra (✉)

Gandhi Institute for Education and Technology, Baniatangi, Bhubaneswar, affiliated to Biju Patnaik University of Technology, Rourkela, Odisha, India
e-mail: sambitmishra@gietsr.com

Z. Polkowski

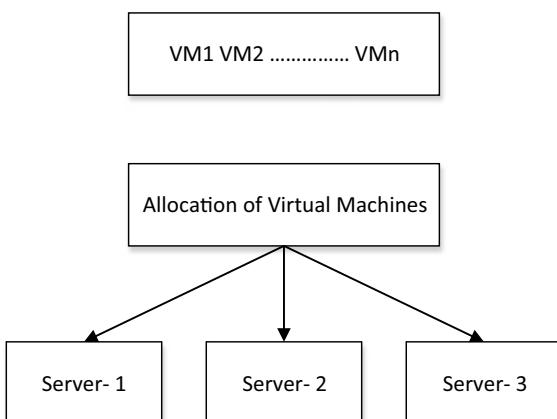
Jan Wyzykowski University, Polkowice, Poland
e-mail: z.polkowski@ujw.pl

1 Introduction

In general case, managing data is very essential provisioned with implementation mechanism and efficient retrieval mechanism. Somehow, the traditional schedulers face challenges towards handling these applications properly and efficiently. In such situation, it is most important to prioritize data handling applications. The concept of virtualization mechanism in such situation is very much required as it accumulates and possesses control on the resources applying encapsulation and of course, it is its potentiality. In fact, for security measures, the attributes associated with either primary keys or foreign keys should be managed properly based on key management applications. Computation mechanism in virtual platform is the need of the present situation and therefore more efficiency and effectiveness should be preserved towards of large storage allocation. The service providers accordingly should focus on enhancement of capacity of virtual machines. Also, the implementation of computational services, i.e., database as a service can overcome the difficulties and fulfill the standardized operations in general. Based on the consideration, usually, the large scaled data centers are being comprised of heterogeneous servers having provisioned with infrastructure as a service during applications. Every server associated with the data center is accommodated with processors as well as processing elements. In fact, virtual machines can be classified and characterized through the efficiency of the processors along with the network bandwidth. In every situation, the data center should be able to manage whether it is high-processor instance or small as well as micro instance concentrating on the requests. Again the computing capabilities of every virtual machine should be proportionate as compared with processing capabilities of individual processors.

Specifically, the criteria linked to the virtualization system are based on utilization of resources and their performance evaluation. In fact, the resources allocated in the virtual machines can also be altered dynamically during the operation as shown in Fig. 1. Accordingly, the servers can focus on scheduling mechanisms to enhance both

Fig. 1 Allocation of virtual machines



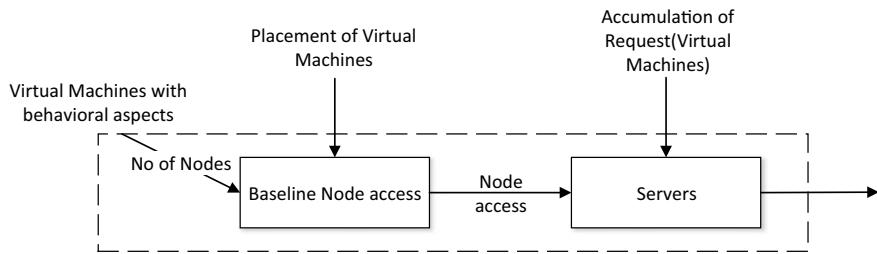


Fig. 2 Accumulation and placement of virtual machine

accessibilities as well as scalability. The scheduler initially checks the dependency among the processing elements and the sharing of the processing elements should also be provisioned with sharable resources. In such situation, it is required to maximize the throughput and minimize the delay response.

In fact, managing resources in the virtual platforms is based on the accumulation of requests and the mechanisms linked with allocation of virtual machines. Accordingly, resource allocation can be done dynamically prioritizing the baseline node access as shown in Fig. 2. Therefore, it is required to make prioritization on allocation of virtual machines to servers and evaluate the performance of the processing elements linked to individual virtual machines (Fig. 3).

2 Review of Literature

Kitchen et al. [1] in their work prioritized the security parameters in cloud computing. In fact, in their work, they discussed the issues of cloud computing as well as enhancement of security and response time.

Baldini et al. [2] during their studies have focused on mechanisms towards detection of malware, as provisioning cloud computing facilities in such situations can assist towards keeping the data more safe and confidential.

Dempsey et al. [3] in their work observed the workability of internet service providers towards enhancement of internet based services. In fact, the services associated with virtual platforms can enhance the global storage allocation with all facilities.

Varghese et al. [4] in their work have focused on implementation of cloud computing services. Also, they observed that the private cloud networks are having more priority than the general service providers.

Baldini et al. [2] in their work focused on analysis of data different companies. They observed that the cloud computing paradigms are more required towards accession of analytical information at the time of requirement.

Kundu et al. [5] in their work focused on complexities while adopting cloud computing. In fact, they focused on governance as well as development of hardware systems. Also, they focused on specific approach towards focusing on the problem

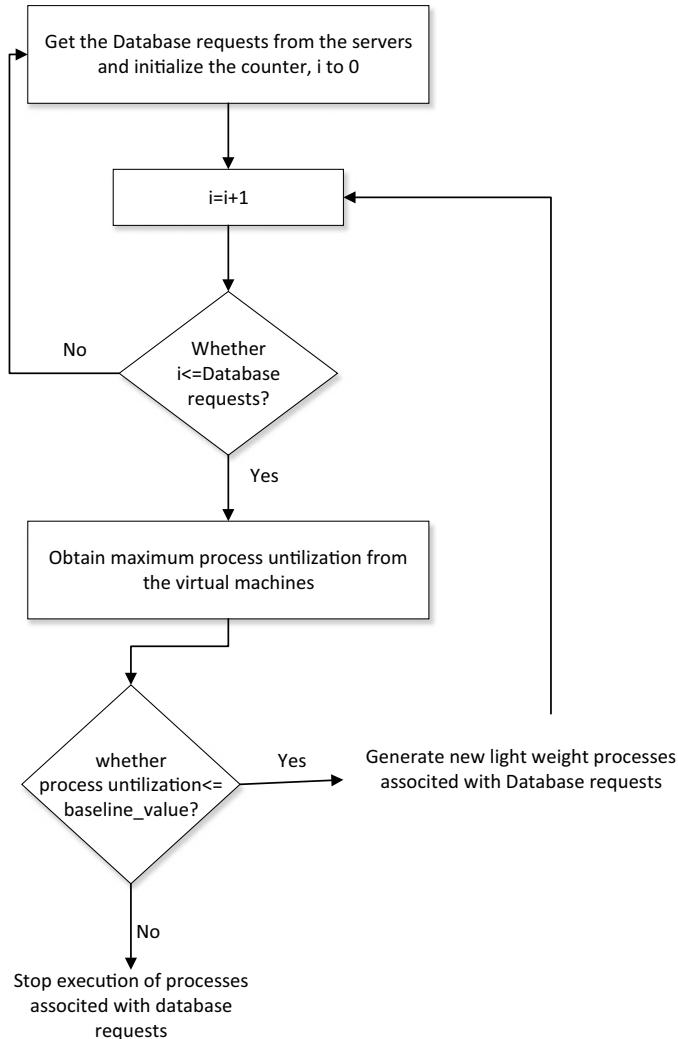


Fig. 3 Flowchart of process utilization from virtual machines

linked to hardware development as well as verification from a black-box driven model to a white-box model to enhance the reliability along with efficiency.

Cao et al. [6] in their work have proposed the blockchain technology towards protection of the data being tampered by the cloud provider. Specifically, in their technique, the data can be stored in the public blockchain and also can be modified with proper authenticity. In fact, the analysis of security measures can be more prioritized and more efficiency should be attained.

Stoyanova et al. [7] in their discussion regarding transformation of Big data including property management. Considering the potentiality of the applications of

big data they have also focused on the challenges. Their main intention is to inspect the capabilities of the big and to identify the opportunities associated with them.

Tarashev et al. [8] in their work focused on extraction mechanisms as well as prediction mining associated with the large scaled data while exploring the reserves and forest resources in the mining region of Russia. In their research, they have also implemented the related forecasting mechanisms based on comparative studies.

Vasilev et al. [9] in their application prioritized the provisioned web based access towards extension of functionality of the enterprise resource planning system along with the designing mechanism and implementation. In fact, they focused on specific web based software solution mechanisms on the large scaled data in the databases along with enhancement of functionality approaches.

3 Application of Ant Colony Optimization

Generally, while associated with the mechanisms linked with retrieval of data, it is very much required to manage the data in structured manner implementing suitable indexing and querying mechanism. Also, the representation of queries should be focused on formalism and similarities which can also be standardized during retrieval mechanism. In such situation, the application of ant colony technique on the retrieval mechanism proves the better one obtaining the optimality. Basically, this technique is inspired considering the behavior of real ants along with the pheromone trails. In most situations, the pheromone trails are updated after accumulation of complete route and somehow it is linked with problem specific local heuristics. In such situation, the ants that are artificial in nature can proceed and iterate implementing the artificial pheromone on each state of the search space to achieve the optimal or near optimal solution. In fact, just like the real ants, in this situation, the shortest route can be chosen to reach near the solution. While initiating the search, each query term associated with the data can be linked with pheromone representing its uniqueness with the preceding steps while obtaining the minimum near optimal solution.

3.1 Steps to Manage the Level of Pheromones

- Step 1: Initialize the level of pheromones and focus on the population of ants
- Step 2: While(!eof (end of file) || termination condition not satisfied) do
 - Step 3: Develop candidate solutions
 - for counter = 1 to maximum iteration, do
 - Choose randomly the selected path of the ant
 - Step 4: Initiate and continue local search (calculate the cost and solution for the ants applying functional parameters)
 - Step 5: Update the pheromone values.

4 Algorithm: Application of ACO Accessing Query Terms Linked to Pheromones

- Step 1: Initialize population of ant and associated parameters.
- Step 2: Estimate the fitness parameters of each query terms
 - apply the cost parametric values and update the position
- Step 3: while(!eof || !total_iterations) do
 - for ant_q = 1 to total_iterations
 - apply the cost parametric values and update the position
- Step 4: for ant_q = 1 to total_ants
 - Update pheromone locally and globally
- Step 5: Obtain the optimal cost parameters of ants.

In this work, it is primarily aimed to observe the response time by allocating the databases in the virtual machines and to determine the cost of query terms by applying ant colony optimization techniques. So, each query term associated with the data is linked with pheromone representing its uniqueness with the preceding steps while obtaining the minimum near optimal solution. Initially, the level of pheromones is to be maintained and it is essential to focus on the population of ants. After the fitness parameters of each query term is to be determined. Finally, the optimal cost parameters of ants can be evaluated by updating the pheromone level at local and global states. It is observed that to optimize the performance in the virtual servers, the size of databases in the virtual servers is required to be maintained consistently.

5 Experimental Analysis

Somehow the criteria linked to query optimization in the virtual platform are less concerned as many times it can be accommodated with the default parameters. Therefore these are required to be properly taken care during the allocation of resources in the virtual platform. Sometimes the instances of a database can be shifted from one location to another or can also be replaced by some other instance which is in turn required to be monitored. To some extent, the query optimization criteria can be able to solve this issue but may not be always due to virtual deployment. In fact, it can only focus on prediction of query evaluation process and enhanced performance. Therefore, to optimize the performance in the virtual servers, the size of databases in the virtual servers is required to be maintained consistently. Also, the response time should be based on the allocation of databases within the server. The parametric evaluation of each query term in such situation has been processed through MATLAB 13B for further analysis and result generation (Table 1).

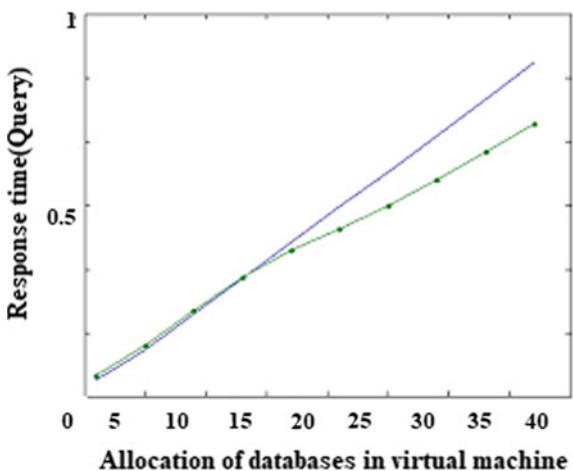
It is understood that computing paradigm associated with cloud is primarily based on virtualization mechanism. Also, the very essential application in such situation tends towards management with databases. In such situation, the control mechanisms

Table 1 Allocation of databases in VM with response time

S. No	Size of database in VM	Response time(ms)
1	19	0.47
2	29	0.56
3	34	0.67
4	40	0.74

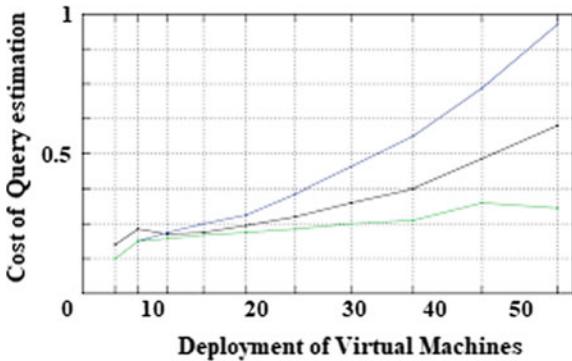
should be adhered prioritizing the virtualization. Therefore it is required to focus on obtaining feasible solutions optimizing the performance of the instances of the databases associated with the virtual machines based on response time as shown in Fig. 4. Of course, the access linked to applications in virtual platforms has provision of accumulation of virtual machines towards construction of high end applications. But somehow, in some situations, it can be a difficult task towards prediction of response time of virtual machines due to dynamic scalability. Therefore, it is required to monitor and prioritize the response time of initial processing element along with its linkages with other processing elements linked to the virtual machines (Table 2).

While deploying any specified applications in virtual machines, it is essential to measure the capacity of the virtual machines as well as to focus on the deployment process. In fact provision of deployment of virtual machines is based on several

Fig. 4 Allocation of databases in VM with response time**Table 2** Deployment of virtual machines with cost of query terms

S. No	Size of virtual machines	Cost of query terms(ms)
1	20	0.34
2	30	0.47
3	40	0.56
4	50	0.74

Fig. 5 Deployment of virtual machines with cost of query terms



factors linked to the current situations, i.e., virtual processing units, storage allocation as well as utilization of networks. In some specific virtualized platforms, there is provision of flexibility in computation in optimum. In such situation, accessibility to the virtual machines should be gained towards deployment of databases through the virtual machines. As shown in Fig. 5, the cost of execution of query terms within the databases depends on the size of the deployed virtual machines.

6 Discussion and Future Direction

Indeed, the application associated with cloud in the presentation situation is most acceptable due to the linked elasticity properties along with efficient deployed database applications. In fact, this characteristic can enhance the performance of databases and help to obtain highly optimized parametric values. In addition to the consideration of deployment of virtual machines, it may be also required to focus on the linked clusters provisioning the facilities to minimize the operating costs of associated data centers.

7 Conclusion

The virtual platforms in true sense permit the accessibility towards global storage and instant access of data at instance. In fact, the service providers have the major roles to enhance the workability along with the allied services. But in every respect, it is desirous and should be more focused on data security. In the real sense, there should be more provision on sharing and accessing data in virtual platforms prioritizing the entity integrity as well as referential integrity. Of course, the computation in virtual environment is quite competent towards hosting the large scaled data. But prioritization of these data in the virtual platform sometimes becomes challengeable in the real

application. The mechanism of allocation of these data in different forms in virtual platforms can also be the support in many development sectors. The main merit in virtual computation along with the large scaled data is its integral portion and storage allocation with processing abilities. In fact, it can have access also towards large scale resources and can also support data analytical activities. In return, there may also be the provision of minimum complexity and gaining maximum productivity. Also having provisioned with operational flexibilities it is also needed to enhance the applicability linked to virtual machines. The optimization criteria of the virtual machines can be evaluated based on balanced ratio of the processing elements with storage allocation. The deployment of virtual machines in all stages is based on the optimization criteria of the resources. In fact, it is intended more towards virtual processing elements, storage allocation as well as assigned networks.

References

1. Kitchen K, Reiss M (2018) Ransomware is coming. It'll make you WannaCry. Heritage Foundation. Retrieved from <https://www.heritage.org/technology/commentary/ransomware-coming-itll-make-you-wannacry>
2. Baldini I, Castro P, Chang K, Cheng P, Fink S, Ishakian V, Suter P (2017) Serverless computing: current trends and open problems. In: Chaudhary S, Somani G, Buyya R (eds) Research advances in cloud computing. Springer, Singapore, pp 1–20
3. Dempsey D, Kelliher F (2017) Industry trends in cloud computing: alternative business-to-business revenue models. Springer, Berlin, Germany
4. Varghese B, Buyya R (2018) Next generation cloud computing: new trends and research directions. *Futur Gener Comput Syst* 79:849–861. <https://doi.org/10.1016/j.future.2017.09.020>
5. Kundu A, Sura Z, Sharma U (2018) Collaborative and accountable hardware governance using blockchain. In: 2018 IEEE 4th international conference on collaboration and internet computing. IEEE, Washington, DC, pp 114–121. <https://doi.org/10.1109/CIC.2018.00026>
6. Cao S, Zhang G, Liu P, Zhang X, Neri F (2019) Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain. *Inf Sci* 485:427–440. <https://doi.org/10.1016/j.ins.2019.02.038>
7. Stoyanova M, Vasilev J, Pompiliu Cristescu M (2021) Big data in property management. In: Conference: thermophysical basis of energy technologies (TBET 2020), Lab: Marian Pompiliu Cristescu's Lab, March 2021, AIP Conference Proceedings, vol 2333, issue no (1), pp 070001
8. Tarasyev AM, Vasilev J, Turygina VF, Strelchuk AE (2019) Methods for predicting the production of natural resources. *AIP Conf Proc* 2186(1):050010
9. Vasilev J, Kehayova-Stoycheva M (2019) Sales management by providing mobile access to a desktop enterprise resource planning system. *TEM J* 8(4):1107–1112. ISSN 2217-8309. <https://doi.org/10.18421/TEM84-01>

Implementation of Secure Communication Framework for Wireless Sensor Network



Pankaj Kumar Sharma and U. S. Modani

Abstract Sensor network protection and safety standard required can vary based on particular application specifications, whether sensor networks are installed. So far, most protection strategies for sensor networks have been layered, implying that a specific approach may be extended to a single layer itself. Integrating both of them is also a new task for science. If the network is used for some role sensitive purposes, such as a military battlefield, this issue is more critical. In real-life implementation environments, random node failure is often possible. The article address the issue of maximum-independent set issue using a wireless sensor network as a completely concurrent and distributed hardware architecture. This article is a simulation of a neural network from TOSSIM. The simulation model was developed using TinyOS with the default protocol stack along with nesC. Mote counting simulations of 10, 50, 100, and 182 were carried out; the expense of messages complexity, memory, and the period of simulation were calculated. Results showed that the neural optimization algorithm could measure the solutions to the MIS problem as the most relevant finding.

Keywords WSN · Security · Dos attack · SDN

1 Introduction

Wireless sensor network (WSN) is an evolving system that uses a variety of small nodes to detect different phenomena in an accessible environment. Because of the resource restriction of the sensor nodes and their operation in an accessible world, the nodes are more vulnerable to attacks inside and outside [1, 2]. Authentication, anonymity, and honesty may be ensured by the encryption method. However,

P. K. Sharma (✉)

Department of CSE, Government Women Engineering College Ajmer, Ajmer, Rajasthan, India
e-mail: pankaj.cse@gweca.ac.in

U. S. Modani

Department of ECE, Government Engineering College Ajmer, Ajmer, Rajasthan, India
e-mail: drusmodani@ecajmer.ac.in

researchers suggest a confidence-based scheme to address external Black holes, sink pit, and Denial-of-Service assaults (DOS). Administration of trust in other network areas such as social networks, ad hoc networks, and P2P has shown successful performance. Confidence templates and approaches for other networks do not explicitly refer to the network of wireless sensors, since WSN is a network of resource restrictions [3]. The Fig. 1 presents the model of a wireless sensor network. Confidence in WSN may be seen as a faith in connectivity and data confidence, or trust in the node, pathway and operation. In the wireless sensor network, the following concept of confidence offers general reasons. The faith is characteristically arbitrary, non-transitive, thoughtful, and asymmetrical. If the node does any operation in accordance with clear networking laws, the trust between nodes increases. If a node breaks the network laws, nodes must be marked as compromised nodes and excluded from further network contact. The node will develop trust by analysing and advising it directly. The suggestion, faith means that confidence values align more easily on the basis of neighbouring recommendations. However, as the topology of nodes varies dynamically, such suggestion systems are more useful. The stable sharing of recommendations raises correspondence costs. In order to measure the trust of a node on the network, it is easier to use direct observatory-bases trust. Belief is the chance of a node determining the confidence degree. For e.g., 0 refers to full mistrust and 1 refers to full confidence. Esteem and real possibility are said to be reliable. The predicted likelihood of conviction. The mismanagement of the confidence and confidence differences would enable space in relation to weaknesses to be misestimated [4, 5]. Trust alone in all activities is in certain situations not enough. However, before they are included in the confidence estimates the risk, standard of service, and trust must be discussed separately. Specifically, device designers are conscious of the data to be protected for what kind of protection service in a different area of sensor network applications. Two typical WSN implementations can be taken as instances, for example, agriculture agriculture farming, and the military surveillance framework, although for agriculture, data integration is only feasible (HASH); however, security resources such as encryption, authentication, and strong resistance to node breach attacks are required by military surveillance. In order to validate the safety guarantees and raise the application developers' understanding about what elements are stable and risky, the protection system of an application must always be thoroughly evaluated to prevent a false sense of security [6, 7]. For a fair security assessment, we have inserted another logical aspect in the sensor node framework, namely ISA, which addresses a specific sensor network implementation security level needs.

1.1 Literature Review

Kandris [1] with the many advantages offered by their use, Wi-Fi sensor networks are one of the most quickly developing areas of technology. As a consequence, wireless sensor networks have seen an increasing variety of applications from their first introduction up to now. The purpose of this article is to present both conventional

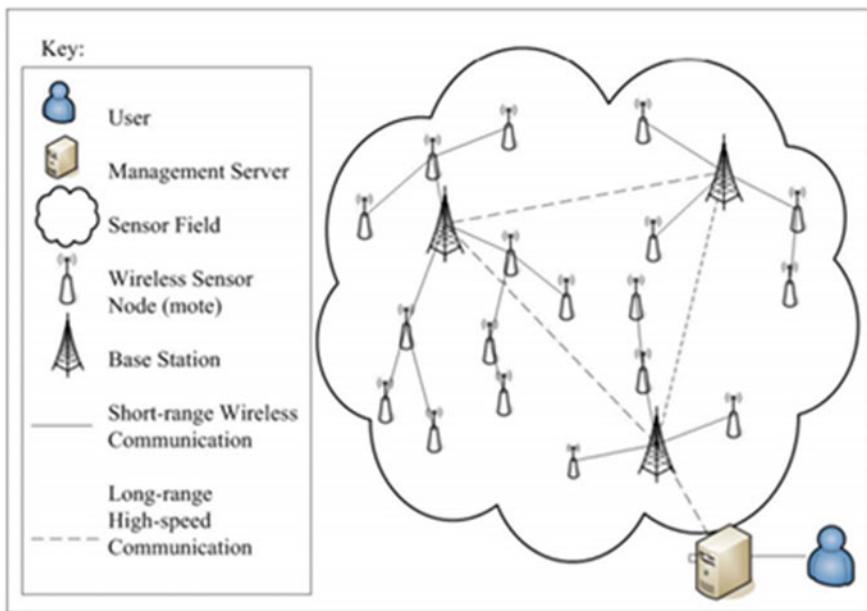


Fig. 1 Wireless sensor network

and latest wireless sensor networks technologies up to date and I hope not only to allow the scientific field to be understandable but also to enable new applications to be perceived. The key applications of wireless sensor networks are defined in order to accomplish this aim, and characteristic examples are analysed. They describe their features and show their benefits and drawbacks. Next, there is a summary of certain considerations relevant to both of these groups. Lastly, the final statements are made.

Jyothi [8] wireless sensor network (WSN) small capacity, poor computing functions, and electricity restrictions. WSN is effective in cases of connectivity without technology assistance, in terms of these restrictions. The major security issue of WSN is the possibility of attacking sensor nodes and hacking details. WSN's protection should be able to guarantee, and not changed during a transmission, that the message obtained was transmitted from a specific node sent. For WSN applications, lightweight and powerfully authenticated data collection mechanism from unprivileged users is essential. The best approach for stopping unwanted and unbroken communications in wireless sensors networks is authentication. Several researchers have established methods to facilitate authenticated connectivity. The popular protection techniques are encryption and decryption. This are built on publicly available cryptosystems or symmetric key schemes. Many of the current technologies have connectivity and computer expertise drawbacks. There is often little power and scalability of the network given by existing structures. A polynomial approach has been implemented recently to solve these problems. Key delivery in the key control of WSNs is an important factor. The fastest way to deliver the key is by hand in couriers

days. Nowadays, the majority of keys is immediately distributed. In networks where two parties are expected to transfer their security keys in the same medium, the automatic delivery of key is critical and convenient. This paper suggests a different form of mechanism for main exchanges. The proposed authentication mechanism between sensor nodes is promising according to the results of the simulation. The unknown nodes set up a single, yet arbitrary, key for the symmetric key cryptosystem.

Alves [9] the software-defined networking (SDN) model can provide adaptive routing and can accommodate the various wireless sensors network connectivity patterns (WSN). However, it is not easy to adapt this model to resource-constrained networks, particularly where protection services are a must. Over the span of time, current SDN-based methods to WSN established to meet resource restricted needs. However, defence systems are not integrated into its architecture and execution. A reliable SDN-based architecture for wireless sensors networks is the key contribution to this work. The core resources the system must have are called safe entry to nodes and end-to-end key delivery to facilitate secure communication. We define the specification, configuration, execution and tests of the system and protocol. The findings show that we have accomplished these targets with sufficient overheads to medium-sized networks.

Usama [10] there is a strong requirement for the introduction of protection measures for protected wireless communication due to the wide size and rapid growth of wireless sensor networks (WSNs). The WSNs have several implementations in which efficiency and cost reduction play an important role. The constrained, confined design of the sensors, and the potentially dynamic actions of WSNs require that a protected communication system be correctly applied so as to deter an intruder from manipulating or modifying the transmission illegally. We also suggested a WSN architecture that includes four modules, i.e., a redundancy checker, a message prioritisation system, malicious node checking, and malicious data verification. For comparison and assessment with the full deployment with NS2 network simulator, comprehensive protection and performance review results have been obtained. The proposed architecture has various interesting features to show that malicious nodes, and data in WSNs are performing acceptably.

Geetha [11] a new system that is used to feel and track the world is the wireless sensor network. Since the nodes are used in an accessible world, protection is one of the key considerations. The strategies of encryption will guarantee secrecy and honesty. The wireless sensor network must also cope with attackers both internally and externally. Many researchers in the field of the wireless network sensors propose a confidence mechanism for dealing with external attackers, assaults by hack or malicious nodes. The trust management system can be used in various security management applications, including secure information aggregation, cluster head set, trusted routing, access control, etc. For these safe systems, several researchers have various types of solutions focussed on confidence management. However, it is essential to plan and develop a confidence management scheme that takes into consideration the different facets and implementations of the wireless system sensors on a single sensor node on the network. In this document, we suggest a stable communication architecture and a method of confidence management for wireless

sensor networks focussed on the parameter and Factor of confidence. Our main contribution is to recognise different parameters and trust factors that influence Wi-Fi confidence and have a framework for trust management, which is dependent on different parameters and trust. MATLAB simulation studies for secure communication, data aggregation, and identification of intrusion in wireless sensor networks illustrate the functioning of the proposed model.

2 Scheme Design

The weakness of WSN protection schemes which rely exclusively on symmetrical encryption is the attacker's vulnerability to access keys on a sensor's mote. In addition, due to memory constraints and other security considerations, only a minimum number of keys can be pre-distributed in WSN circumstances. As a result, an intruder may still have access to large sections of transmitted information from a small number of compromised motes. Tamper-resistant hardware can help reduce this issue, but usually it is too costly.

An inherent safety issue of WSNs that rely solely on symmetrical encryption is its susceptibility to attackers who gather sensor mote access to keys access. Furthermore, only a small number of keys can be pre-distributed in WSN situations owing to memory limitations and other reliability factors [12]. As a result, an intruder may now obtain access to significant sections of information via a low number of compromised motes. Manipulative hardware can alleviate this issue, but it is usually too costly.

However, the above solution is less costly in terms of the expense of computing such that the resources of motes must be saved. Additional precautions must be taken to guarantee that attackers cannot replay intercepted mote packets. In the handshake, used to bind two motes, all parties must make sure that their communications partner lives and does not only replay older packets, thus missing the handshake's portion. As previously stated, the symmetric key should be produced with both motes. If one of the motes only replays packets, the negotiated key cannot be properly determined. Packets are authenticated through HMACs that use the handshake key to ensure that certain cases are detected. Because a block cypher is used for symmetrical encryption in counter mode, security from replay attacks is easily realised by throwing packets with less counter value than the previously agreed packet and then inserting an HMAC via the encrypted message and counter.

3 Platform Security Features

As previously noted, node sensors are subjected to caught attacks that subsequently reveal sensory node information and adversary credentials. The sensor node protection features should be improved to mitigate the impact of this threat. This is supported by the suggested sensor node platform:

1. Private key and other confidential credentials secured memory for safely managed sensory node.
2. Confidence zone involves a division into stable and non safe mode and area of execution mode and memory region.
3. Secure boot mechanism that measures boot images integrity and selected peripherals on a platform sensor node and ultimately.
4. The unique identification of the sensor node is generated depending on the unique value of the platform components chosen.

The plan used node identities and node management value that were created only after the first boot-up node is configured for preventing unauthorised nodes in the network. In the authentication protocol IBE-Trust is used the values along with identity-based cryptography (IBC) algorithms.

The Procedure structure is submitted. In the first installation node, IBE-Trust offers a secure platform that provides valid nodes for reporting their management value at the base station. Safe contact between nodes of the network is essential for wireless as a means of communications in the WSN world. IBE-Trust offers basic safety features:

1. **Privacy:** clarify the communications submitted to all recipients that are unreadable, like eavesdroppers. Secrecy Generated by encrypting the message with the public key on the base station.
2. **Authenticity:** to check the correct sender for the packets or messages obtained. Generated with the sender identity authentication.
3. **Integrity:** checking the validity of the submitted packets. Generated by the algorithm Message Authentication Code (MAC).

The suggested framework's summarised method flows as seen. In order to validate integrity of the running codes, it occurred first at the sensor node. The result of a so-called measuring method is a particular attribute that represents a single object of a platform. This phase also creates a specific platform identification, which is used in the protocol for identity encryption. When the sensor node boots successfully, the measurement value will be reported to the basis station. The value is sent by the IBE-Trust protocol encrypted to the base station (BS). This procedure ensures trust, accuracy, and completeness of the letter. When BS receives the value, sender authentication will be carried out accompanied by calculating value verification, which will validate the efficiency of the node entering the network.

4 Security Analysis

A TinyOS implementation has been published and reviewed to check the feasibility of the scheme. The implementation has been first evaluated on a small-scale IRIS motors test bed in TOSSIM, the simulator used on the TinyOS [13, 14]. TinyCEC was selected for the implementation to include ECC and some functions for cryptographic utilities as they are a security-sufficient implementation in the real world,

and a Quite fast tempo of AVR platform deployment. XTEA Block Implementation cypher in nesC were written from scratch, a random cryptographic generator number and a counterblock cypher mode. Determine Random Byte Generator Counter mode is chosen, since the block encoder must also be documented. The system was built around TinyOS' AMSend and Receive interfaces and used a limited range of additional commands and events in a new CryptoLayer interface. This new gui allows users to search and directly build and drop links with details on currently linked and known motes.

First a couple of ECC keys are created to be used as keystones of a central authority while preparing to blink the motes. The private key is used to enter the mutual keys and mote addresses and to send the public key. In compilation time, cryptographic keys are then generated for motes. The central authority will then sign the public key and address of the mote. The signature is often directly compiled into the mote: CA public key and the random number generator input for an initial entropy. Table 1 shows the scale of the different packet types of the software.

Connect any extra overhead to the headers of the format underlying the AMPacket, which varies on the platform, but is typically 5–12 bytes. This execution was checked frequently with TOSSIM in the course of creation. TOSSIM enables TinyOS-based motes to be simulated utilising the TinyOS programme directly. A script Python 2.5, such as the number of motes used, boot times and signal strength of the motes, is used for defining the action of a simulated network. For the safety checking in TOSSIM, the following situation has been implemented: 100 motes are uniformly used on a grid of 25 m to 50 m in different simulation runs. As seen in Fig. 2 the signal intensity between the two motors is proportionate to the Euclidean driver benefit of 0 dBm at a minimum -112 dBm at a maximum grid distance. The Fig. 2 presents the 25-mote TOSSIM for simulation run.

There are decreased to 10 the number of simultaneous encounters with each mote to imitate limited stories on the actual mote. As these motes shape the connections, the server user data packets containing the “user data packet” sequence send the pairs periodically. Any mote is used with a debugging function, such as receiving and sending parquets, signature checks, etc., to provide precise information about the current state of its activities. The performance is documented in the log file and analysed to assess the virtual WSN's interconnect at various times. A few motes are

Table 1 The various package forms used in this implementation

Packet	Size (in bytes)
Certificate	62 B
Key exchange offer	78 B
Accept	58 B
Finish	21 B
User data	> 30 B
Error	52 B

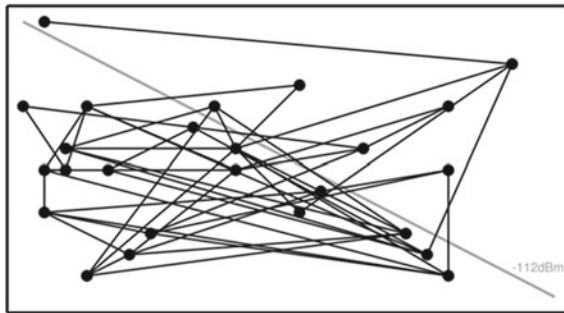


Fig. 2 A 25-mote TOSSIM simulation run

Table 2 The average number of links formed over many simulation runs, until a certain connectivity is reached

Connectivity	Avg. total connections	SD
> 50 % of motes	200	11
> 95 % of motes	723	48
= 100 % of motes	887	27

Table 3 Value settings for Hop-field neural network limit weight parameters

Mote count weight parameters	10	50	100	182
g_a	0.2	0.4	0.7	0.6
g_b	1.2	1.2	1.2	1.2

said to be connected to the simulation when the first user data packet is transmitted between them. The network's connectivity is the number of motes in the largest connected motor chart. Further, the Table 2 demonstrates the average number of links formed over many simulation run.

Simulations were performed for WSNs with mote counts of 10, 50, 100 and 182. Case of any mote count has been computed ten times and an allocation has been used to classify data statistically. Table 3 presents the setting of the two limit weight parameters along with the binding result of the second paper in the three-paper sequence. Three performances measurements have been measured: the TOSSIM memory footprint of the WindowsTM OS, the amount of neural computation packets shared and the overall TOSSIM simulation period as given by the TOSSIM. Includes all messages obtained by each mote with a local counter for monitoring the messages sent by the total number of packets.

Table 4 t-distribution model parameter values for message complexity measure

	10	50	100	182
Mean (packets)	22734.8	41132	527469	3842367
Deviation	23773.45	43752.52	42365.60	2701705.01

Tables 3 and 4 display simulation effects as plots of box and whisker. The WSN would not alter considerably its memory footprint: the distance between the WSN processes with a speed of 10MB (19.5MB) and 182mote (21.2MB). However, for 10 mote instances, the message complexity dictated by the total number of network sensor packages is increasing dramatically from tens of thousands to millions for the 182-mote case. This also influence the length of the simulation: the 10-mote sensor network simulation takes approximately 20 s and the 182-mote network case almost 3 min. For all three measurements in complexity, the t-distribution values are provided in Table 4.

5 Conclusion

This paper presented the findings of the simulation with the aid of TOSSIM, which was used as the hardware framework to perform a fully parallel neural optimization algorithm and wireless network sensor distributed mode. The suggested parallel and distributed neural optimisation problem analysis was shown to be feasible by the simulation findings. The study's complexity has shown that the costs of messages and extended simulation periods for vast sensor networks must be overcome by implementing more complex TinyOS MAC and routing protocols. We have tackled the problem of testing and analysis of sensor network applications and have shown that scalable, highly reliable simulations of complete sensor network applications can be generated. The number of errors and faults in key TinyOS services shows that simulation can be an essential stage in application growth and that it is not prohibitive to move easily between implementations and simulation. Our hope is that the sensor network group can be used by TOSSIM to allow research in this new and large area.

References

1. Kandris D, Nakas C, Vomvas D, Koulouras G (2020) Applications of wireless sensor networks: an up-to-date survey. *Appl Syst Innov* 3(1):14
2. Sharma K, Ghose M et al Complete security framework for wireless sensor networks, arXiv preprint [arXiv:0908.0122](https://arxiv.org/abs/0908.0122)
3. Kumar V, Jain A, Barwal P et al (2014) Wireless sensor networks: security issues, challenges and solutions. *Int J Inf Comput Techno (IJICT)* 4(8):859–868

4. Bok K, Lee Y, Park J, Yoo J (2016) An energy-efficient secure scheme in wireless sensor networks. *J Sens*
5. Kifayat K, Merabti M, Shi Q, Llewellyn-Jones D (2010) Security in wireless sensor networks. In: *Handbook of information and communication security*. Springer, Berlin, pp 513–552
6. Gao Y, Ao H, Feng Z, Zhou W, Hu S, Tang W (2018) Mobile network security and privacy in WSN. *Procedia Comput Sci* 129:324–330
7. Alfandi O, Bochem A, Kellner A, Göge C, Hogrefe D (2015) Secure and authenticated data communication in wireless sensor networks. *Sensors* 15(8):19560–19582
8. Cholli NG et al (2020) An efficient approach for secured communication in wireless sensor networks. *Int J Electr Comput Engi* (2088-8708) 10:(2)
9. Alves RC, Oliveira DA, Pereira GC, Albertini BC, Margi CB (2018) Ws3n: wireless secure sdn-based communication for sensor networks. *Sec Comm Netw* 1–14
10. Usama M, Bin Muhaya FT (2013) Framework for secure wireless communication in wireless sensor networks. *Int J Distrib Sens Netw* 9(12):585491
11. Geetha V, Chandrasekaran K et al (2014) A distributed trust based secure communication framework for wireless sensor network. *Wirel Sens Netw* 6(09):173
12. Serpen G, Li J (2011) Parallel and distributed computations of maximum independent set by a Hopfield neural net embedded into a wireless sensor network. *Procedia Comput Sci* 6:390–395
13. Li J, Serpen G (2011) Tossim simulation of wireless sensor network serving as hardware platform for Hopfield neural net configured for max independent set. *Procedia Comput Sci* 6:408–412
14. Li J, Serpen G (2011) nesC-tTnyOS model for parallel and distributed computation of max independent set by Hopfield network on wireless sensor network. *Procedia Comput Sci* 6:396–401

EMBRACE: Electronic Medical Record Safety, Blockchain to the Rescue



Parth Khandelwal , Rahul Johari , Medha Chugh , and Anmol Goel

Abstract The cutting edge technologies have changed the perspective to an intelligent virtual world where machines will interact with humans and data is the fuel to it. Data carries an infinite amount of potential to change the way of decision making. Healthcare industry has also been focusing on collecting large amounts of data through different types of smart devices, care unit systems, and medical equipment. This big data is stored in cloud for its processing and can be used to create smart and intelligent diagnosis and health management systems with the back support of AI and ML making it easier for the healthcare providers and executives. The major issue faced is the security and the privacy of healthcare data collected from patients and storing it. Blockchain can serve the right purpose to secure the data collection and sharing in the cloud based applications in future. Though, it has some limitations too. This paper focuses on the development and implementation of a plugin for health management system which ensures the storage of all the data entered by the user into the Blockchain. The hashing technique used to store the information is Secure Hash Algorithm (SHA-256) to securely save the data. The application developed can successfully save the data and allows the sharing of it with other providers too. For testing the application, records from liver and chronic kidney disease datasets collected from Kaggle, are stored in the blocks to create the Blockchain network. This can prove to be a major step towards the practical implementation of such applications at an enterprise level.

Keywords Healthcare · Blockchain · Security · Patient's records · Health care and Blockchain · Types of Blockchain · Hashing

Supported by GGSIP University

P. Khandelwal () · R. Johari · M. Chugh · A. Goel
SWINGER (Security, Wireless, IoT Network Group of Engineering and Research) Lab, USICT,
GGSIP University, Sector-16C, Dwarka, Delhi, India

1 Introduction

With the advent of incredible technologies such as Artificial Intelligence (AI), MiCEF (Mist, IoT, Cloud, Edge and Fog Computing), and Big Data tools, each sector of industry has been excelling towards a visionary future and changing the game of business worldwide. However, security, privacy, and trust are still¹ the main concerns faced by every enterprise which slows down the pace of undergoing revolution. To deal with these raised concerns of security and privacy, the industries have found a potential solution in the name of Blockchain Technology.

Healthcare industry has already been on the depriving side of this race of development of digital infrastructure. The traditional ways of Healthcare management systems are undergoing changes to create better systems. The initial steps have been taken by collection of health records using various IoT devices such as wearables like bracelets and fitness bands while working out, patient's health records in hospital management system, data collected from tests like Electrocardiography (ECG), Magnetic Resonance Imaging (MRI), etc., Intensive Care units (ICU) and other methods of diagnosis and treatments and storing them in the cloud system. However, the amount of data collected is large and remains underutilized. For the proper utilization of data, it is needed to ensure that data collected from various resources can be integrated without inconsistencies and redundancy. Also, the privacy of the patients remains at stake and any hacker can easily corrupt the data and misuse it violating the privacy rights of a patient. Therefore, security is to be ensured to avoid such circumstances. Blockchain is a public distributed ledger technology which can not only ensure data privacy and security but can also help to attain interoperability of the data records among health providers. And therefore, blockchain offers it to be a valid option for being the technology behind the enterprise application which are being designed and developed in the healthcare sector. Currently, the health care providers and executives have been investing their time and efforts to develop enterprise based healthcare solutions in order to automate the systems for the data integrity, interoperability of the patient's data records, intelligent diagnosis, and drug supply chain management and blockchain can play an integral role in such systems.

1.1 *Blockchain*

Blockchain is the collection of digital pieces of information called blocks, chained to each other where each block stores various transactions performed and is immutable in nature. These blocks are visible to all the users connected to the network which ensures trust. Adding to that, the need for validation of each new block that is to be added in the chain by each user participating in the network makes it trusted system.

¹ Parth Khandelwal, Rahul Johari, Medha Chughand Anmol Goel.

1.2 *Blockchain in Healthcare*

The sharing of information in traditional healthcare systems doesn't seem viable option in the current scenario as there are several problems associated with it. Sharing information of patients and their health records among different health providers for analysis and decision making is hindered by the lack of trust in the network. Every region in the world has different medical protocols and rules.

EMBRACE: Electronic Medical Record Safety, Blockchain to the Rescue leading to strict restrictions and varying costs per transaction offered to recording and sharing of data. Also, the access to population health data is limited to the privacy rules associated. Varying data standards result in different data formats leads to desynchronization of the patient,s records collected.

2 Literature Survey

In [1], author(s) presents the need for security and privacy of the Electronic Health Records (EHR) stored in cloud in real time when generated using wearable like fitness bands, tests, Health Informatics System (HIS). The author addresses the issues raised on using traditional distributed databases for the storage of patient's health records and then writes about the solutions offered by blockchain. The big data generated can be stored using blockchain technology which ensures the implementation of cryptographic techniques to avoid any changes or thefts of data. With the advantages of transparency, integrity, and security of data stored, it proves to be a cutting edge technology. The author then discusses the underlying challenges of the privacy laws which demand immediate erasure of data when requested by the individual and immutable data hashing.

In [2], author(s) proposes a model which can be used for personal data storage. It allows the user to store and modify his/her personal documents in a highly secure environment. The security can be achieved by blockchain which ensures the authenticity and privacy of the owner of the documents. Also, the document can be shared directly in encrypted form without hampering the security. Also, the confidential data can be stored and scattered across various data custodians and giving it only on request of access.

In [3], author(s) suggests an emotion detection system of the students through the images obtained from the CCTV surveillance system installed on the campus. The emotions of happiness, sorrow, anger are predicted by applying Adaboosting on the data obtained after image processing. The results obtained are used to counsel the students who may have psychological troubles, depression, or anxiety issues. The accuracy obtained on applying the model came to be 83%. The author successfully shows the use of big data in emotional analysis.²

² Parth Khandelwal, Rahul Johari, Medha Chughand Anmol Goel.

In [4], author(s) designs a mobile healthcare application that integrates blockchain to the cloud for data sharing among various healthcare providers and insurance companies. The system includes users, healthcare providers, insurance providers, cloud databases, and blockchain networks as the entities. The users uses IoT devices such as wearables and mobile phones and the sensors collect all the data and health records of the users and save it to an online account. This data is later transferred to the blockchain network in the cloud. Merkle tree, which is a binary tree data structure stores the pair of record hashes at its nodes. The data can later be shared and accessed by insurance companies to decide the pay rate of the users. Also, the doctors use the data to suggest treatment to the patients. The integrity of the healthcare data stored is secured by using proof of integrity and validation method to the blockchain which can be retrieved any time from the cloud database.

In [5], author(s) uses technical and domain specific metrics to evaluate the use of blockchain in healthcare applications. The paper can serve as an initial step towards the development of healthcare uses cases based on the novel technology of blockchain. The metrics for assessment of the application were that the complete workflow implemented should comply to the Health Insurance Portability and Accountability Act (HIPAA) where there should be support for authentication and identification of the user and healthcare provider. Further, the system should be cost effective and can be generalized for a large population, and should be designed taking care of the patients. Structural interoperability should be minimum and it should be able to support Turing operations.

In [6], author(s) talks about the blockchain based remote monitoring system of patients. The system does not focus on storing the personal or health related information of the patient and instead, records the events of visit or treatment at the distributed ledger. The patient in the system whenever is evaluated by the doctor, then the medical devices collect data and that data is sent and compiled by the application in the smart devices. This information is then fed to smart contracts named Oracle using Ethereum protocol. In addition to a new block to ledger, both the doctor and the patient are notified through an alert message on the smart devices. The system proved to be better than the traditional method as it provides efficiency, speed, confidentiality and can be easily available, accessible to all.

In [7], author(s) has implemented a mobile application named Healthcare Data Gateway (HDG) which uses BlockChain to securely store, control, and share patient's data and also ensures the privacy of the patient. The model is developed such that patient can control the adding of any new entry in the database and no other person or party can use it without the patient's permission. The architecture used consists of three layers namely storage layer, data management layer, and data usage layer. The storage layer uses the blockchain cloud which is secured using cryptographic techniques so that no one can corrupt the records. The data management layer consists of private data gateways which handle all the requests and queries made by the users to access or update the records. The data usage layer is where all the actors are present. Healthcare providers, technicians,

EMBRACE: Electronic Medical Record Safety, Blockchain to the Rescue patients, doctors, and insurance companies can request data but the patient has all the authority to mask the data that he/she wants to keep hidden from others, and then no one else will be able to access it.

3 Objective

The primary objective of undertaking the current work was to design and launch state of the art BlockChain that comprise the Patient Liver's Clinical Test Record, secured using SHA 256 Algorithm. The dataset was fetched from Kaggle. The figure shown presents the flowchart of the implementation of the block's creation using the records of the patients.

4 Methodology Adopted

Major steps adopted to solve the problem statement are enumerated as follows³:

1. The major initiative was to design a BlockChain oriented Secure application. The application would be available for use as Plug-In by Hospital Management System. The proposed application was programmed in Java using ECLIPSE IDE. The application received following initial inputs from end user, before sending the patient data for the purpose of verification and validation:

Patient Age: P_a
Patient Gender: P_g
Patient tot_bilirubin: P_{tb}
Patient direct_bilirubin: P_{db}
Patient tot_proteins: P_{tp}
Patient albumin: P_{al}
Patient ag_ratio: P_{ag}
Patient sgot: P_{sgpt}
Patient sgot: P_{sgot}
Patient alkphos: P_{alp}

2. After receiving the Patient Liver's Clinical Test Record, a 32 bit number representing Patient Concatenated Record Information was prepared.
3. If the length of Patient record was less than 32 bits, dummy bits were padded to make Patient record 32 bits.
 - (a) Each block contained following information:-
 - (i) hash Value of previous block

³ Parth Khandelwal, Rahul Johari, Medha Chughand Anmol Goel.



Fig. 1 Hashing of a given input

- (ii) Data
 - (iii) Timestamp
 - (iv) Current block hash value
- (b) hash Value of previous block(Genesis block) always contains a Zero.
4. Hash value of the Current block was calculated by invoking the SHA 256 Algorithm
 5. The Chain of the block was then stored in the ArrayList.
 6. Sample Record taken from liver dataset:

age	gender	tot	bilirubin	direct	bilirubin	tot	proteins	albumin	ag	ratio	sgpt	sgot	alkphos
65	Female		0.7	0.1		187	16			18	6.8	3.3	0.9

The above record when saved in blockchain looks like as:

Block Contents 64 bit hashed block after applying SHA 256 algorithm

Contents of 1st Block

a9733b3051d3331854f81df41a35d0f6e56146e67834c7a93db874b78a479169

Contents of 2nd Block

9144287536b06190435ebdb8295596b6e204429d466ee80a17c2f478a97577eb

Contents of 3rd Block

87a8f6d20909e949e0e56295131de4a32e9cbeec971a66a29c2d5a0ade8b2932

Contents of 4th Block

8f5a81c4c014920a5b51186034826bf867533fd0c191b025a7101aaadf1dbbd8

EMBRACE: Electronic Medical Record Safety, Blockchain to the Rescue 7.

5 Result

The snapshot shown in Fig. 1 shows the patient record application which is designed with Java as the programming language using Eclipse as the Integrated Development Environment (IDE). The output shows the attribute values entered for the storage of the record in the application. The snapshot in Fig. 2 shows the content of different

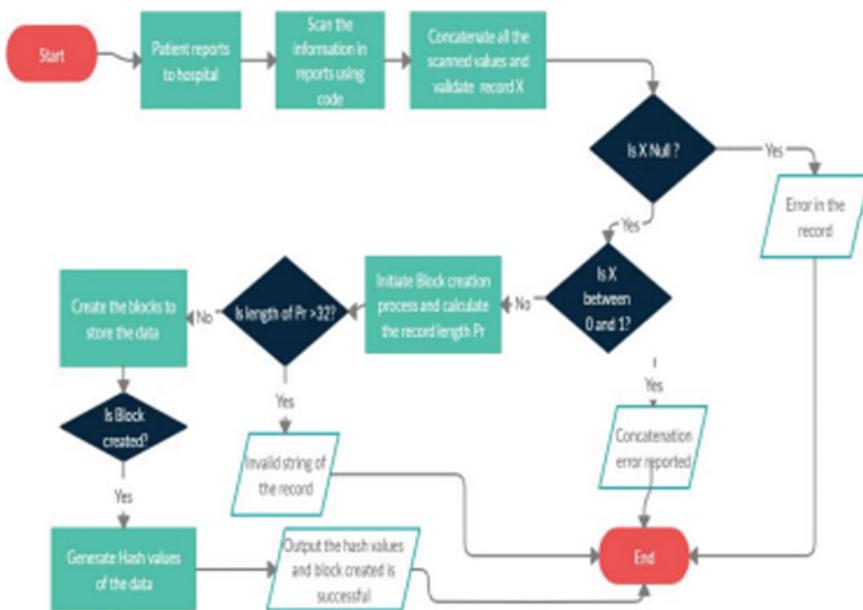


Fig. 2 Flow chart to show the implementation of the blockchain

blocks after applying SHA 256 algorithm on them. One cannot determine the store value using the hash value directly hence the information though, accessible to all is not decoded easily (Fig. 3).

6 Conclusion and Future Work

Blockchain Technology is one of safe, strong, and secure technology. It is a robust and resilient technology with a primary objective to store, fetch, encrypt the data not with symmetric and asymmetric cryptographic techniques but with the help of hash based algorithms such as MD 5 [Message Digest 5] and SHA [Secure Hash Algorithm]. Blocks are all linked and connected together so that any attempt by Hackers and Crackers can be successfully thwarted. In the current research work, the application of BlockChain technology was successfully demonstrated using real time authentic dataset as the record of patient⁴ liver dataset was imported from Kaggle dataset, it was then verified and validated, suitably padded so as to ensure that the block, designed and developed are similar and symmetric. Such blocks were then hashed using SHA 256 algorithm, so as to ensure the EMR [Electronic Medical record] of

⁴ Parth Khandelwal, Rahul Johari, Medha Chughand Anmol Goel.

```

1 package bch;
2
3 import java.security.MessageDigest;
4
5 public class Block {
6
7     public String hash;
8     public String previousHash;
9     private String data;
10    private long timeStamp; //as number
11    private int num1;
12    public String blockdata;
13    //Block Constructor.
14    public Block(String data, String previousHash) {
15        System.out.println("Inside constructor");
16        this.data = data;
17        this.previousHash = previousHash;
18        this.timeStamp = new Date();
19        this.hash = calculateHash();
20    }
21
22    public String calculateHash() {
23        String calculatedhash = "";
24        try {
25            MessageDigest md = MessageDigest.getInstance("SHA-256");
26            md.update(data.getBytes());
27            byte[] hash = md.digest();
28            StringBuilder sb = new StringBuilder();
29            for (byte b : hash) {
30                sb.append(Integer.toString((b & 0xFF) + 48));
31            }
32            calculatedhash = sb.toString();
33        } catch (Exception e) {
34            e.printStackTrace();
35        }
36        return calculatedhash;
37    }
38
39    @Override
40    public String toString() {
41        return "Block{" +
42                "hash=" + hash +
43                ", previousHash=" + previousHash +
44                ", data=" + data +
45                ", timeStamp=" + timeStamp +
46                ", num1=" + num1 +
47                '}';
48    }
49}

```

Fig. 3 Sample snapshot of BlockChain—patient record application programmed in JAVA using ECLIPSE IDE

the Liver record of patient dataset are safely created and archived in database for future reference.

In future, it is proposed to carry out the current research work on more and more EMR dataset and to provide BlockChain Technology as free Java based ECLIPSE IDE oriented plugin so that it can be integrated with the already operational IT based Health Care Management System is used in thousands of Primary care clinics and hospitals worldwide.

References

1. Esposito C, De Santis A, Tortora G, Chang H, Choo KKR (2018) Blockchain: a panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Comput* 5(1):31–37
2. Chowdhury M, Colman A, Kabir A, Han J, Paul S (2018). Blockchain as a notarization service for data sharing with personal data store. 1330–1335. <https://doi.org/10.1109/TrustCom.BigDataASE.2018.00183>
3. Sinha S, Mishra SK, Bilgaiyan S (2020) Emotion analysis to provide counseling to students fighting from depression and anxiety by using CCTV surveillance. In: Machine learning and information processing. Springer, Singapore, pp 81–94
4. Liang X, Zhao J, Shetty S, Liu J, Li D (2017) Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In: 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC). IEEE, pp 1–5
5. Zhang P, Walker MA, White J, Schmidt DC, Lenz G (2017) Metrics for assessing blockchain-based healthcare decentralized apps. In: 2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom). IEEE, pp 1–4

6. Griggs KN, Ossipova O, Kohlios CP, Bac carini AN, Howson EA, Hayajneh T (2018) Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. *J Med Syst* 42(7):130
7. Yue X, Wang H, Jin D, Li M, Jiang W (2016) Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *J Med Syst* 40(10):218
8. <https://www.investopedia.com/terms/b/blockchain.asp> Accessed in March' 2021

Stress Prediction Using Machine Learning and IoT



Vividha, Drishti Agarwal, Paras Gupta, Soham Taneja, Preeti Nagrath, and Bhawna Gupta

Abstract Stress is a mental condition that affects every aspect of life leading to sleep deprivation and various other diseases. Thus, it's necessary to analyse one's vitals to stay updated about one's mental health. This paper presents an effective method for detecting cognitive stress levels using data from a physical activity tracker device. The main goal of this system is to use sensor technology to detect stress using a machine learning approach. Individually, the impact of each stressor is assessed using ML models, followed by the construction of a NN model and assessment using ordinal logistic regression models such as logit, probit and complementary log-log. The paper uses heartbeat rate as one of the features to recognise stress and the Internet of Things (IoT) and Machine Learning (ML) to alert the situation when the person is in real danger. The patient's condition is predicted using machine learning and the patient's acute stress condition is relayed using IoT. Based on the heartbeat, a prediction of whether a person is under stress or not can be made. The paper presents a model that can predict stress levels based entirely on electrocardiogram (ECG) data, which can be measured with consumer-grade heart monitors. The ECG's spectral power components, as well as time and frequency domain features of heart rate variability, are included in the model. The stress detector system takes the real-time data from the IoT device (sensor), then applies a machine learning model on the data to detect

Vividha (✉) · D. Agarwal · P. Gupta · S. Taneja · P. Nagrath
Bharati Vidyapeeth's College of Engineering, New Delhi, India
e-mail: vividha.cse1@bvp.edu.in

D. Agarwal
e-mail: drishtiagarwal.cse1@bvp.edu.in

P. Gupta
e-mail: parasgupta.cse1@bvp.edu.in

S. Taneja
e-mail: sohamtaneja.cse1@bvp.edu.in

P. Nagrath
e-mail: preeti.nagrath@bharatividyapeeth.edu

B. Gupta
Panipat Institute of Engineering and Technology, Samalkha, India

stress levels in an individual and eventually informs/alerts the individual about their stress condition. By collecting data, this system is tested and evaluated in a real-time environment with different machine learning models. Finally, a comprehensive comparative analysis has been depicted amongst the applied models with Random Forest Classifier showing the highest accuracy. The novelty of this work lies in the fact that a stress detection framework should be as unobtrusive to the user as possible.

Keywords Electrocardiography (ECG) · Electromyography (EMG) · Galvanic skin response (GSR) · Heart rate (HR) · Internet of Things (IoT) · k-nearest neighbours (KNN) · Physionet · Random forest · Support vector machine (SVM)

1 Introduction

Stress in the twenty-first century has emerged as an integrated section of our everyday life and is a prominent notion in the healthcare sector. Nowadays, Stress has become an inherent part of everyone's life, generally in today's competitiveness of this generation. In the workplace, every person continuously faces several situations, like excessive work pressure, job insecurity, poor job satisfaction and the pressure of staying updated. The constant stress pressure can cause multiple negative health effects, which can include high blood pressure, sleep deprivation, vulnerability to infections and cardiovascular diseases. All these conditions lead to mental stress, which is the prime cause of many diseases these days. Such unfortunate effects not only affect a person's health and well-being but also result in low productivity and overall profit. Stress can be detected using data and sensors like accelerometer, keystroke dynamics, or blinking.

Generally, an ensemble of several markers is used while compromising with an increased budget and user involvement. New non-contact methods have also been discovered to quantify stress, which includes hyperspectral imaging techniques, human voice, pupil diameter, visible spectrum camera, or using stereo thermal and visible sensors.

The objective of this research are:

- To integrate the ML techniques with the IoT devices along with web deployment on the cloud servers.
- To achieve state-of-the-art performance by the project so that it can be used for stress detection accurately even in real-life scenarios.
- To analyse the results and compare the performance and accuracy of multiple ML classification algorithms.

The paper is composed of V sections. Section I being the introduction followed by Section II, the related works in the field of stress detection. In section III, the research methodology is explained thoroughly and is further divided into subsections like Data collection and preprocessing, Feature Selection, Classification Model and IoT device

architecture. In Section IV, the results are mentioned and analysed. Finally, Section V concludes the paper and discusses future work in the given field.

2 Related Surveys

Owing to its increased importance in contemporary society, human stress has become an occasional subject in research papers (Table 1). There has been extended research work in terms of tackling stress and mental health issues in everyday life with the help of IoT and machine learning applications. [1] uses Theano in which stress is detected using the position of the eyebrow from its mean position. Many studies have been performed to determine the exact facial features associated with depression. A depressed/stressed face has the same features as a sad face that can be identified by the SVM classifier [2].

Variability of the heart rate refers to variations in the beat of the heart rate. It can be predicted, based on heart rate, whether or not a person is stressed. To evaluate the stress level of any person using his/her heart rate, an IoT system called Remote Stress Detector is used in [3]. Stress can also be detected by using sensors to measure body temperature and skin conductance, evaluating the degree of stress [4]. Signals like ECG and breathing rate are calculated with the aid of various wearable sensors [5].

Lakudzode and Rajbhoj [6] introduces a mobile phone emotion recognition device that can be used as a smart keyboard. With ML techniques, the smart keyboard senses the emotional state of an individual. [7] discusses Health Monitoring and Prognosis wearable sensor-based systems; compares different systems to address the flaws of the present biosensor networks, & offers guidance and direction for upcoming unobtrusive ways. In [8], Heart Rate Variability (HRV) trends are examined by normal healthy people during sleep. This is done for stress awareness. It is understood that HRV is an indication of the development of the autonomic nervous system. The modification of the beat-to-beat (RR) intervals represents HRV.

Minguillon et al. [9] proposes a three-level approach (stress, relaxation and neutrality) to detect stress with a few-second resolution and 86% accuracy. Both hardware, as well as the software, were installed in their laboratory, they produce a method called RABio w8 (real-time detection of biosignals, wireless, eight channels).

Three blocks (acquisition block, control block and communication block) form RABio technology. They did a study that included ten volunteer studies to test their approach. Normal relative gamma (RG), heart rate (HR), normal skin conduction (SC) and normal trapezia movement are symptoms of depression used here (TA).

Table 1 Literature survey

Paper title	Theoretical/conceptual framework	Methodology	Analysis and results
Detection of stress using image processing and machine learning techniques [1]	A real-time non-intrusive video is captured, which detects the emotional status of a person by analysing the facial expression	Employs a technique that allows to train a model and analyse differences in predicting the features. Theano is a python framework that aims at improving both the execution time and development time of the linear regression model which is used here as a deep learning algorithm	The experimental results show that the developed system is well on data with the generic model of all ages
Machine learning and IoT for prediction and detection of stress [3]	The developed prototype detects whether a person is under stress using variability in his/her heart rate	The heartbeat readings from a well-calibrated end IoT device are pushed to the server where they are filtered using a user's network id to keep track of readings for a particular individual. The data is applied with various machine learning models for detection	Stress labels are calculated and the median in our data set was stressed and map that to our detector, it is not possible to detect someone's age by their resting heart rate—while some correlations exist, the relationship is not clearly defined. The applied SVM and Logistic Regression show considerable improvement over VF—15 and Naive Bayes
Portable system for real-time detection of stress level [9]	Present and validate a portable system for real-time detection of stress level, based on the RABio w8 (real-time acquisition of biosignals, wireless, eight channels) system	The hardware was made of portable, wireless and low-cost electronics. The software was composed of an application programming interface (API) and a graphical user interface (GUI)	The classification accuracy or probability of success (pain in the detection of stress level (stress, relax and neutral) and the 95% confidence interval were reported

3 Research Methodology

The research methodology consists of various stages including a collection of data and preprocessing, feature selection, classification using stress detection using Random Forest Classifier and the construction of IoT devices (Fig. 1).

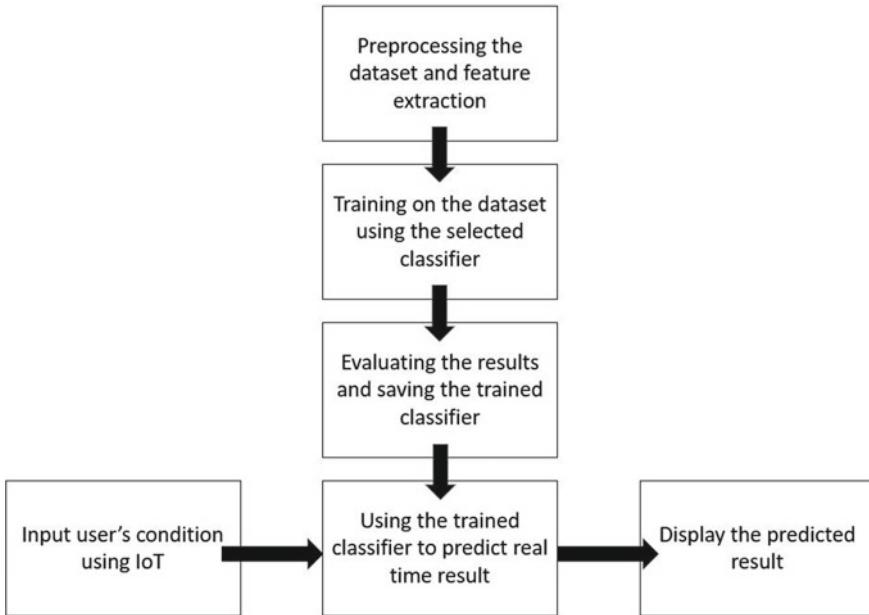


Fig. 1 Project workflow

1. Data collection and preprocessing

The database is in the form of a specific Physionet format which is divided into Eighteen.dat files & Eighteen.hea files having the corresponding metadata [10]. This data was made up of signals from the foot such as ECG, EMG and GSR steps from the hand, HR and Respiration. All variables are floating-point numerics, with a sampling frequency equal to 15.5 per second. To clean the database we use software created by Physionet called WFDB to format the files into comma-separated values with their column names. After that, the heart rate variation is based on the RR gaps by running a custom python programme called the WFDB system.

Heart rate variations can be determined from a 20-s instantaneous window for the closest RR interval, with 10 RR intermediate samples from the preceding immediate window and 5 RR intermediate samples fetched from the succeeding immediate window. This leads to a variable section at a time having a full size of 30 s. and prevents cutting hard using split windows.

This method can insert any events towards the end or at the beginning of windows, to determine multiple RR peaks when calculating HR variables, simultaneously resulting in a database consisting of more samples instead of the case where the windows were, e.g. 30 s without splitting. The cleaned database is stored in a.csv file.

Data enhancement ensures that binary numbers labels are 1 (stressed) or 0 (free) for ease and accuracy in categories. We erase all the lines manually and

then fill in the missing values using the inf value of inf. The heartbeat columns are further cleaned with a central filter, this is done to remove reading errors.

2. Feature Selection

The galvanic skin response features are not considered here which were used to label data, as well as ECG along with EMG data were denied as these details won't be easily and simply available on wearable devices in the consumer market directly. Also, the ultra-low frequency (ULF) fraction is not considered as most values are nill, which is caused due to the short intermittent periods of the signals. Also, the low frequency band (VLF) data is rejected as a study from various researchers found that the lowest frequency band proves to be inefficient in less than 5 min of reading and also that the samples that are in the dataset are inconsistent.

The feature set now contains the Average of all NN intervals (**AVNN**), **Ratio of Low to High (LFHF)**, **Standard Deviation of NN intervals (SDNN)**, **Percentage Difference of NN intervals (pNN50)**, **Total spectral Power (TP)**, **Root Mean Square of standard Deviation of difference between NN intervals (RMSSD)**, **Low Frequency**, **RR intervals**, **Heart Rate**, **High Frequency**. In 4132 samples it was necessary to exclude cardiac variability factors as the RR intervals and heart rate is the obvious values from a 30-s interval.

3. Classification Model

Machine Learning is applied for classification based on the features selected. Several ML classifiers were implemented to get results. Out of all the classifiers, Random Forest Classifier gave the best accuracy. It is applied to detect whether a person is under stress using physical attributes such as skin conductance, EMG and ECG.

Random Forest Classifier

Random Forest Classifier (RF) is a guided machine algorithm used for partitioning, retreating and other operations using decision trees. Random forest planning creates a set of decision trees from a randomly selected set of training sets. A collection of deciduous trees (DT) from a randomly selected training set and Collect votes on various decision trees to determine the final prediction.

The Random Forest Classifier, as its name states, has hundreds of tropical trees that serve as a team. Every tree in a random forest event explains the forecast of the category and the label having the majority of the votes becomes the output of this RF model. The basic concept of a random forest is simple yet powerful—the wisdom of the crowds. In terms of detail, the reason why the informal forest model works distinctively is that the large number of unrelated species (trees) that function as a jury will outweigh any other species. Low integration between models is key. As small-scale investments (such as stocks and bonds) come together to create a portfolio greater than its total assets, unconventional models can produce an increased number of accurate forecasts in contrast to any other prediction. This positive effect is observed because the trees prevent one another from their flaws (as long as they do not deviate in the same way).

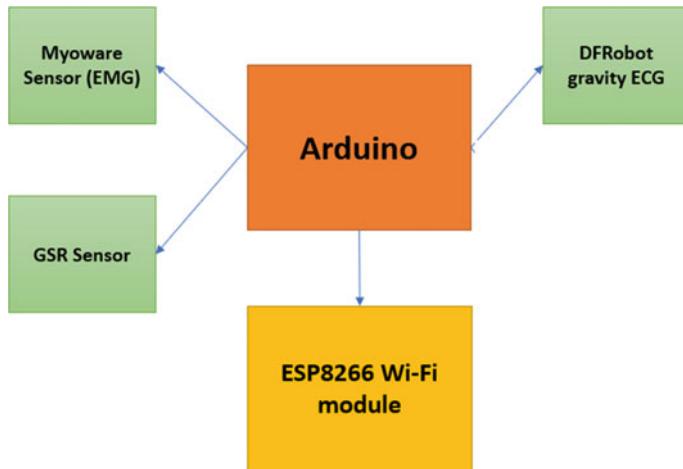


Fig. 2 Device architecture (block diagram)

While some trees may not be good, many other trees will be good, so as a group the trees can move in the right direction. Therefore, the requirements for a random forest to function properly are required to have a certain trait in our features so that models built using those features perform better than random guesses (so mistakes) made by individual trees need to be a fusion of each other (Fig. 2).

3.1 IoT Device Architecture

The device is based on the basic principles of IoT, involving sufficient use of sensors that will measure the required vitals of our users [11] (Fig. 3).

The backbone of any IoT based device is the microcontroller which connects and implements all these sensors together and makes them simultaneously. Arduino is the most effective hardware-software platform to work upon devices involving sensors. Along with this, the data will be sent to a cloud database with the help of the ESP8266 Wi-Fi module which will provide internet access to our Arduino board for data transfer.

4 Results

The Random Forest classifier achieved an accuracy of about 0.735. Figure 4 shows the test results where the algorithm successfully detected the stress, based on the heart rate feature over a particular interval of time which eventually derives the required Heart Rate Variability. As the Heart Rate shows peaks values the algorithm

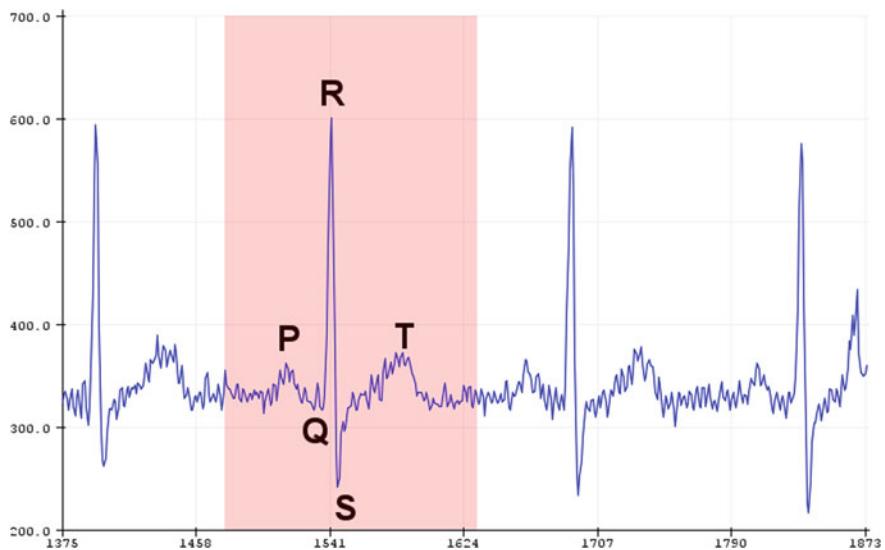


Fig. 3 ECG Generated waveform

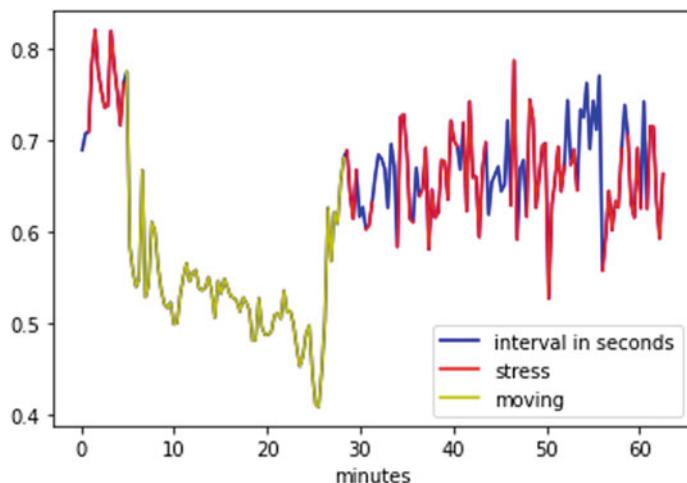


Fig. 4 Graph depicting the relation between HRV and stress of random forest classifier

detects a stressed condition (represented as 1). Thus, establishing a relation between HRV and Stress. The study compared the accuracy of different classifiers which consisted of SVM, KNN and Random Forest Classifier. On comparing the accuracy of Random Forest Classifier was found to be the best with an accuracy of 0.735 (approx.). Figures 5 and 6 show the various Graphs obtained on plotting the HRV and time interval using different classifiers.

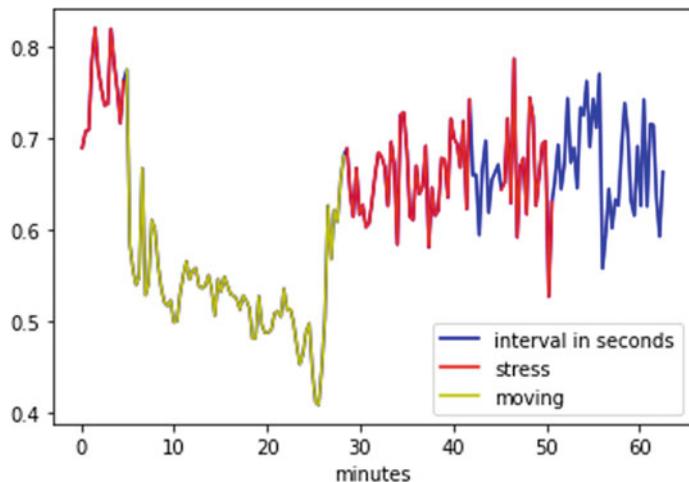


Fig. 5 Graph depicting the relation between HRV and stress of SVM classifier

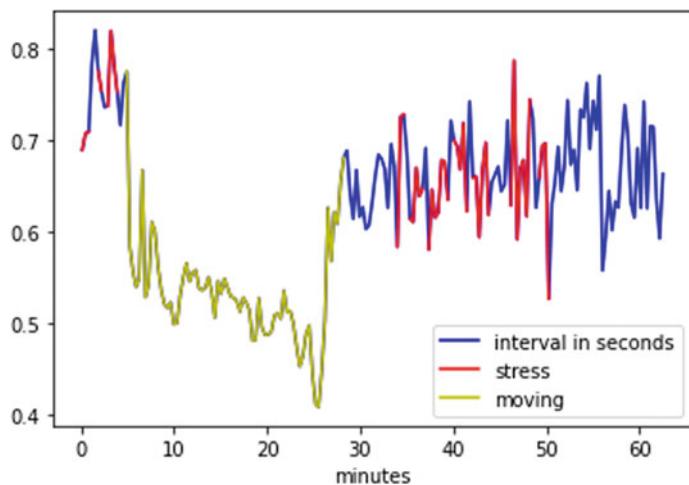


Fig. 6 Graph depicting the relation between HRV and stress of KNN classifier

It can be seen that (Table 2) the RF Classifier outperforms the other ML models with an accuracy of 73.5% as compared to the SVM classifier with an accuracy of 62.2%(approx.) and the KNN Classifier with an accuracy of 73%. Hence, it can be inferred that tree-based ensembling algorithms are suitable for the task of stress prediction.

Table 2 Accuracies of different classifiers used

Models	Accuracy
RF	0.7348668280871671
SVM	0.6222760290556901
KNN	0.7300242130750605

5 Conclusion and Future Scope

In this research study, using IOT device architecture and Machine learning classification algorithms, a device to detect stress was studied and implemented. For the stress detection model, a combined framework using ML and IoT was proposed. The classification of stress using the user's physical attributes such as ECG, heart rate, skin conductance was achieved using a Random Forest classifier. For the physical attributes, IoT device architecture was established with different end-points for different features of the user to detect stress.

For future scope, the research would extend to stress in specific audiences such as maternal stress or stress in students due to different cognitive approaches. Along with this, to ensure greater accuracy in real-world cases, a two-level stress detection approach would be implemented which would take into account the facial and visual aspects of the user as a feature using deep learning-based image classification techniques such as Convolutional Neural Networks (CNNs).

References

1. Raichur N, Lonakadi N, Mural P (2017) Detection of stress using image processing and machine learning techniques. *Int J Eng Technol* 9(3)
2. Venkataraman D, Parameswaran NS (2018) Extraction of facial features for depression detection among students. *Int J Pure Appl Math*
3. Pandey PS (2017) Machine learning and IoT for prediction and detection of stress, IEEE
4. Kumar T, Kumar RS et al (2019) Health monitoring and stress detection system, IRJET, March 2019
5. Hesham, Elkhorazaty Y, Aloul F (2016) Emotion recognition using mobile phones, IEEE
6. Lakudzode SF, Prof. Rajbhoj SM (2016) Review on human stress monitoring system using wearable sensors
7. Pantelopoulos A, Bourbakis NG (2009) A survey on wearable sensor-based systems for health monitoring and prognosis. In: *IEEE transactions on systems, man, and cybernetics, part c (applications and reviews)*; vol 40, issue 1
8. Amir Muaremi; Bert Arnrich; Gerhard Troster; Towards Measuring Stress with Smartphones and Wearable Devices During Workday, Sleep. Zurich, Switzer- land. 2013
9. Minguillon J, Perez E, Lopez-Gordo MA, Pelayo F, Sanchez-Carrion MJ (2018) Portable system for real-time detection of stress level. *Sensors* 18(8):2504
10. Healey J (2005) Wearable and automotive systems for affect recognition from physiology
11. Healey J, Picard W (2000) Wearable and automotive systems for affect recognition from physiology

Mitigation of DDoS Attacks Using Honeypot and Firewall



V. Harikrishnan, H. S. Sanket, K. S. Sahazeer, Siddarth Vinay,
and Prasad B. Honnavalli

Abstract A DDoS attack or distributed denial of service attack is an attempt to make servers unavailable by concentrating the target server with a flood of network traffic. DDoS attacks can cause huge financial loss to businesses. There are different types of DDoS attacks such as SYN flood, Smurf attack, Ping of Death, and Buffer Overflow. Implementing the firewall and honeypot in the network can help us mitigate the effect of the attacks and analyze the attacker's identity and signature. Our paper proposes an architecture which uses a packet generator, a firewall, and a honeypot to mitigate the DDoS attacks when it is happening at the target's endpoint. We demonstrate and discuss these attacks as well as the methods used to block malicious packets. We have simulated multiple attack scenarios and mitigated them with the help of various firewall rules. These blocked packets are later used for analysis in the honeypot.

Keywords DDoS attack · Server · Firewall · Honeypot · Protect · pfSense · Security · Cybersecurity

1 Introduction

A distributed denial of service (DDoS) attack depletes a server of its resources such as computing power, memory, and storage by concentrating a large number of requests to the target server. It can decrease the Web site's performance or even crash the entire server. There has been an exponential growth in the number of DDoS attacks in recent times. There are different types of DDoS attacks such as SYN flood, Smurf attack, Ping of Death, and Buffer Overflow. There are multiple solutions to mitigate DDoS attacks such as distributing requests to different servers using CDN's, using Web application firewalls provided by different companies like Akamai, Zscaler, or blocking traffic with your own firewall using different rules.

A typical production server's resources will be used and accessed by millions of users. A denial of service attack will deny the user the data or resource that they

V. Harikrishnan · H. S. Sanket · K. S. Sahazeer · S. Vinay (✉) · P. B. Honnavalli
PES University, Bengaluru, India

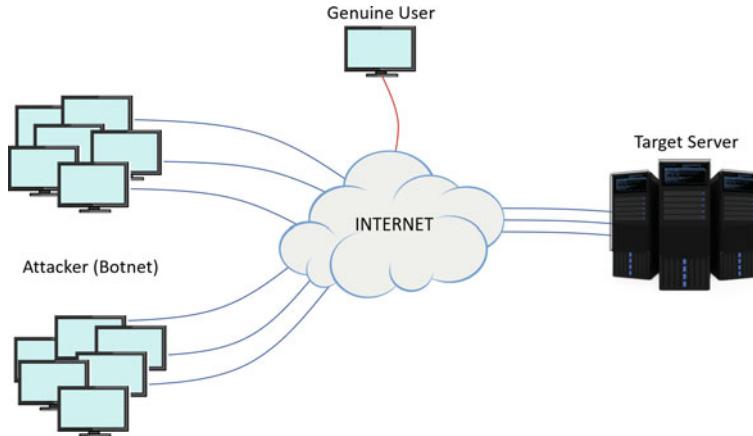


Fig. 1 Illustration of DDoS attack

have requested. Hence, these attacks cause financial and reputation loss to large enterprises. Our solution tries to mitigate DDoS attacks with the help of a firewall and a honeypot when it is happening at the target's endpoint (Fig. 1).

We have simulated various DDoS scenarios using Scapy. The firewall (pfSense) acts as a traffic controller for our network in order to decide whether a request or packet should be permitted in the network based on defined rules. Commonly used rules are based on IP address, port numbers, and protocol used. We use the honeypot to store the malicious packets and try to derive some important insights about the attacker and use it to deduce common attack patterns which are in turn used to develop more secure applications. The flowchart of our proposed method can be seen in Fig. 2.

2 Related Work

The work done by Campbell et al. [1] talks about the embryonic trends with respect to research in the field of honeypots. In this paper, honeypots are perceived as a mechanism for defense by logging any suspicious behavior to understand what is being done (e.g., used for analysis) and to keep track of them for legal purposes later. It gives a brief introduction about the various types of honeypots and how they are used. It gives different perspectives of what people defined honeypots to be. They can be categorized mainly into four types: shadow honeypots, honeynets, honey farm, and honey tokens. They can also be categorized in different ways: based on interaction level, deployment modes, and deployment categories.

The work by Weiler [2] talks about a method to mitigate DDoS attacks. It stresses about honeynets. It mainly stresses about learning how an attacker performs his

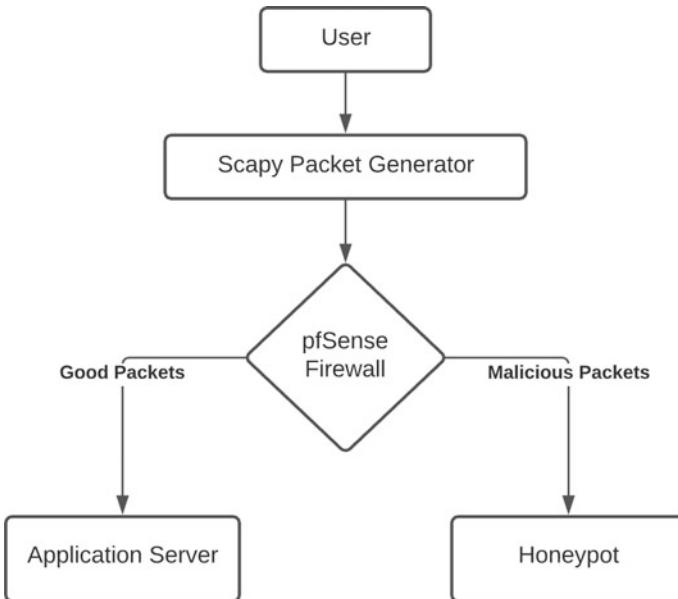


Fig. 2 Flowchart of our method

attack. Honeynet is a research honeypot. It has a different purpose than what a traditional honeypot does. This work describes the effect of a DDoS attack on a honeynet. There are darknet-based approaches as well, which make use of honeyfarms or forensic honeypots for the monitoring of darknets [3]. These approaches can be applied in the production server setting in order to analyze attacks as well as defend against them.

The present networking systems use IP protocol to send packets. The packets generally contain the source and the destination IP addresses of the hosts. But, when the packets arrive at the destination, the correctness of these source IP addresses are not checked for legitimacy. The work by Templeton [4] talks about the various ways which have been figured out to distinguish malicious packets and genuine packets. There are two kinds of high level methods: routing methods and non-routing methods. This work goes in depth into the different routing and non-routing methods available.

Bringer et al. [5] offers an in-depth survey of the various recent developments in honeypot analysis. In order to keep up with the latest developments in the internet, new forms of honeypots have been developed. These developments include the emergence of modern popular network technologies, the widespread popularity of wireless network technology, and variations in consumer demographics. Some of the popular new honeypots introduced are discussed here, including BitSaucer [6], Shark [7], and Shadow [8] honeypots.

The role of firewalls in general network security is well documented in literature [9]. There is also a large quantity of computer network literature which have docu-

mented the effect of denial of service (DoS) attacks. These attacks are undeniably a significant issue on the Internet. The primary goal of a DoS is to interrupt systems by wanting to curtail access to a computer or facility rather than undermining the service itself. This type of attack seeks to make a network incapable of delivering regular service by focusing on the network's bandwidth or connectivity. In the work by Douligeris et al. [10], the authors provide details about the state-of-the-art in this field, in this paper by categorizing DDoS attacks and the security mechanisms that can be used to counteract these attacks.

3 Proposed Methodology

Our project focuses on making the endpoint stronger and less prone to highly impactful DDoS attacks when it is happening at the endpoint. Completely preventing the DDoS attacks can only be done by making a highly accurate anomaly detection system and placing these all over the world. These systems would ensure that when a malicious packet reaches that system in any part of the world, it is dropped, hence making sure that it does not reach the actual endpoint server. There are eight main scenarios which are handled as part of our DDoS mitigation system.

3.1 Geographical IP Based Blocking

Assuming that our services are only offered in certain specific countries, a malicious attacker based in any other country may wish to attack the services and slow it down by performing a DDoS attack on the main server from a location outside the offered countries. This attack would result in a huge loss to the company and result in the attacker being successful.

Since our business offers services only in a limited number of countries, we will consider only packets from these locations as genuine packets or requests. Any other packet from any other location of the world can be dropped in a situation when we are experiencing an unusually high amount of traffic which might indicate a DDoS attempt. But in the scenario where the traffic flow is not unusual, we will allow these packets (Fig. 3).

3.2 TCP Max Connections

We can assume that a malicious attacker wants to cause harm to the company by hacking into multiple client systems (inactive ones who are not currently using the service) and sending excessive amounts of TCP connection packets to the server on different ports to make sure that the server's buffer capacity is filled hence making



Fig. 3 Geographical IP based blocking

genuine clients who actually want to use the service struggle from trying to establish connections with the server. In such a case, we can keep a limit on the number of connections a certain client can have with the server (10 in our scenario), and any other future connection requests from that particular client will be dropped. This ensures that genuine clients will have the opportunity to establish connections to complete their task.

3.3 Invalid TCP Flags

TCP packets are frequently used to carry out DDoS attacks. TCP packets have multiple flags. These flags are SYN, ACK, FIN, URG, PSH, RST, ECE, and CWR. Any invalid combination of these flags may be set in malicious packets.

There are specific flag combinations which are not expected from the client side in a typical client server architecture. A few of them are:

- SYN-ACK—this combination of flags is set only in packets from server to the client.
- SYN-FIN—this combination is not expected as no user is expected to create and end a connection at the same time.
- FIN-URG-PSH—used to perform OS fingerprinting

Any packets with these flag combinations can be blocked at the firewall.

3.4 *Synproxy*

A malicious attacker can perform a SYN flood DDoS attack. The server will take up some of its packet buffer to handle each incoming SYN packet. If there is a flood of these packets, the buffer will be occupied and hence the performance will be affected. The server will not be able to distinguish between the genuine SYN packets and malicious SYN packets, thereby consuming its resources.

A synproxy can be used on the firewall in order to handle the incoming traffic instead of the server. TCP connections are established with the help of a three way handshake: The SYN packet from the client, SYN-ACK packet response from the server and an ACK packet is sent as a response from the client. Usually, the server behind the firewall will handle the connection on its own. However, by enabling the synproxy state the firewall will establish the connection instead. The downside of this method is that it results in reduced network performance due to the extra load on the firewall to manage connections.

3.5 *Rate Limiting*

As we know, DDoS attacks aim to flood the channel with a load of packets and paralyze the performance of the server. Ping flood is one such type of attack. In this, the attacker sends a huge amount of ICMP echo packets as a result of which the target system becomes inaccessible to the normal users. This could result in a lot of inconvenience for an organization.

Rate limiting is the method to control the frequency of packets coming to the interface by putting a restriction on the rate at which they can arrive. This is very useful to stop various types of cyber attacks. Here, it is an effort to stop ping flood attacks. If the rate at which the ICMP packets arrive is limited, the attacker will not be able to negatively affect the server.

3.6 *ICMP Type Based Filtering*

There are cases when a client needs to check the health of the server for diagnostic purposes. For this reason the client should use ICMP echo packets to ping the server. There is no other type of ICMP packet which is required. But during a DDoS attack, any type of packet can be sent to cripple the network. We have to develop a countermeasure for that.

In the event where we find clients sending some other packets like “Router Selection” (type 10), etc., we can drop them. This reduces the need for the server to process the packet and reply to it unnecessarily, hence reducing its effect (Fig. 4).



Fig. 4 ICMP type based filtering in action

3.7 *Bogon IP Addresses*

Bogon addresses are IP addresses that should never be used for global routing. These are IP addresses that are not yet allocated or delegated by the Internet Assigned Numbers Authority (IANA). Malicious users use bogon IP address because they cannot be traced back to an existing host or source.

Routers can never discard packets with bogon IP address because they will only examine the destination IP address of the packet against the routing table, not the source address. The firewall can be used to block all the packets with source IP addresses as bogon IP addresses.

3.8 *Port Blocking*

A few legitimate port numbers belong to the services, such as: SSH(port 22), DNS (port 53), HTTP(80), and HTTPS(port 443). However, there are numerous services such as FTP(port 21) and Telnet(port 23) which suffer from numerous security vulnerabilities through which a malicious user can directly attack our system. We can block all the ports which are not in use or have numerous vulnerabilities associated with it. A firewall can be utilized in order to enable incoming and outgoing traffic on only the ports selected by the organization. The firewall can block all other traffic, thereby mitigating the high impact of DDoS attacks.

4 Implementation

We made use of the network architecture to carry out demonstrations, which looks like Fig. 5. The three main components of our system are the packet generator, the firewall, and the honeypot.

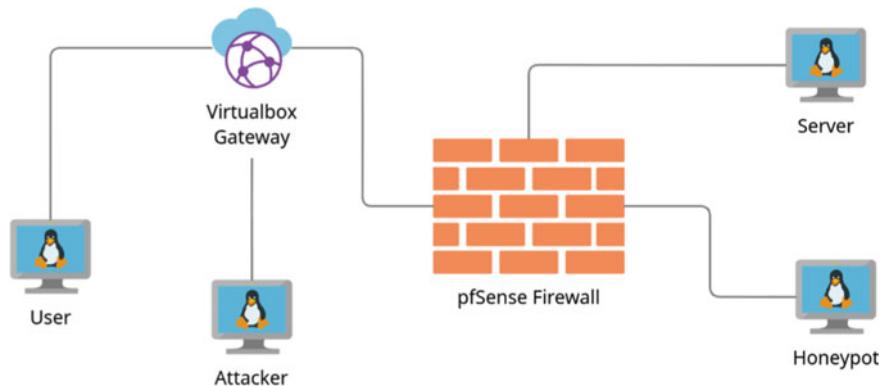


Fig. 5 Network architecture used

4.1 Network Setup

We set up three Ubuntu Virtual Machines and named them as “Attacker,” “Server,” and “Honeypot” and created three virtual networks called “OutsideNetwork,” “ServerNetwork,” and “HoneypotNetwork.” We set up the pfSense firewall and named it as “Firewall.” After that, we configured the network settings for each of the machines as follows:

- (a) Attacker is connected to the OutsideNetwork
- (b) Server is connected to the ServerNetwork
- (c) Honeypot is connected to the HoneypotNetwork
- (d) Firewall has three interfaces, connected to OutsideNetwork, ServerNetwork, and HoneyPotNetwork, respectively.

4.2 Packet Generator

Our packet generator is built to generate packets as per the scenarios described earlier. The packets are generated with the help of the Scapy library in Python [11]. There are 8 different scenarios, and they are UDP flood, GeoIP packets flood, TCP invalid flags based flood, SYN flood, ICMP flood, Invalid ICMP type based flood, bogon packet flood, and malicious port-based flood.

4.3 Firewall

The most important component is the firewall which is responsible for identifying and filtering malicious packets. We set up rules which efficiently distinguish between good packets and malicious packets. The eight scenarios described in the previous section are handled (blocked and logged) with the help of this firewall [12].

4.4 Honeypot

The honeypot converts the raw logs into structured logs (CSV format) using Python libraries like pandas [13] and numpy [14]. The CSV format depends on the type of packet that we have captured. Using this CSV file, we can derive useful insights which can help us make future rules more comprehensive and assist us better in securing our network.

Apart from the components stated above, we have used tools like Netcat [15] for establishing connections and Wireshark [16] for capturing and analyzing the packets.

5 Results and Observations

These are the results obtained for the eight scenarios:

1. GeoIP Packet Filtering: We set up the firewall rules to block traffic from India and allow all other traffic. The generated packets were successfully blocked and visible in the firewall logs.
2. TCP Max Connections: We used Netcat to observe multiple concurrent TCP connections. We set the maximum permitted TCP connections to 2. The third connection via Netcat was unsuccessful.
3. TCP Flag-based Filtering: We created packets with SYN + ACK flags set together. A pfSense rule was set to block this type of traffic. The packets were successfully blocked and visible in the logs.
4. Synproxy: We created packets with only SYN flag set. We enabled the Synproxy setting in the pfSense rules. Using Wireshark, we observed that the SYN packets do not reach the target.
5. Rate limiting: We set up a limiter with a maximum bandwidth of 700 kbps and set it to handle ICMP traffic. Our packet generator sends infinite amount of ICMP echo packets at a rate of 6 M Bits/sec. But we observe that the limiter reduces it to a value lesser than the specified rate. This rule is not completely effective in blocking packets as genuine packets can also get blocked during the rate limiting process.

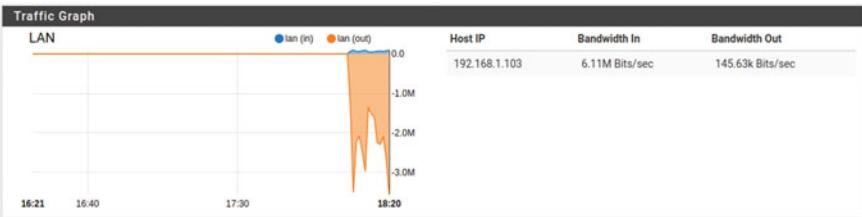


Fig. 6 Traffic graph before rate limiting

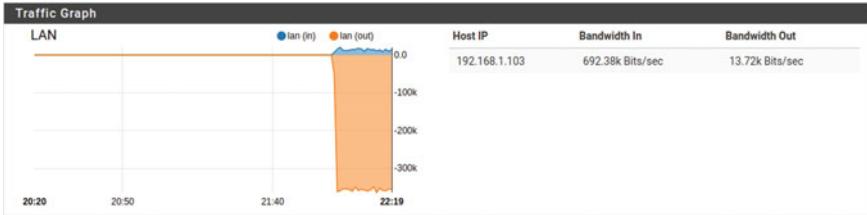


Fig. 7 Traffic graph after rate limiting

6. ICMP Type Filtering: Our packet generator creates ICMP type 11 packets. We set up our rule to only accept ICMP echo packets. We can view the blocked packets in the pfSense logs.
7. Bogon IP Address Filtering: We set up our rule to block packets with bogon IP addresses. We create packets having these addresses. These packets are blocked and can be viewed in the firewall logs.
8. Port blocking: We only allow packets on these ports: 22, 53, 80, and 443. We create packets having different destination ports. The packets are stopped at the firewall and can be viewed in its logs (Figs. 6 and 7).

6 Conclusion and Future Work

DDoS attacks degrade server's performance which severely impacts the target organization. We have proposed an architecture which helps to minimize the effect of these attacks. We simulated various real-world DDoS scenarios and found that our architecture helps in mitigating these attacks. The methods adopted successfully block and log malicious packets which are analyzed in the honeypot.

For further enhancements, we can strengthen the firewall by adding more rules to handle multiple complex attacks. We can also add efficient anomaly detection models to log the packets which seem suspicious. Scaling the systems would help in achieving better performance and providing enhanced security. Additionally, we can provide detailed analysis on the malicious packets logged in the honeypot based on pattern detection, etc., so that it can be used to strengthen our defenses from future attacks.

References

1. Campbell RM, Padayachee K, Masombuka T (2015) A survey of honeypot research: trends and opportunities. In: 2015 10th international conference for internet technology and secured transactions (ICITST). <https://doi.org/10.1109/icist.2015.7412090>
2. Weiler N (n.d.) Honeypots for distributed denial-of-service attacks. In: Proceedings of eleventh IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises (n.d.). <https://doi.org/10.1109/enabl.2002.1029997>
3. Bailey M, Cooke E, Jahanian F, Provos N, Rosaen K, Watson D (2005) Data reduction for the scalable automated analysis of distributed darknet traffic. In: Proceedings of the 5th ACM SIGCOMM conference on internet measurement—IMC ’05. <https://doi.org/10.1145/1330107.1330135>
4. Templeton SJ, Levitt KE (n.d.) Detecting spoofed packets. In: Proceedings DARPA information survivability conference and exposition. <https://doi.org/10.1109/disex.2003.1194882>
5. Bringer ML, Chelmecki CA, Fujinoki H (2012) A survey: recent advances and future trends in honeypot research. Int J Comput Netw Inf Secur 4(10):63–75. <https://doi.org/10.5815/ijcenis.2012.10.07>
6. Yu A, Oyama Y (2009) Malware analysis system using process-level virtualization. In: 2009 IEEE symposium on computers and communications. <https://doi.org/10.1109/iscc.2009.5202313>
7. Alberdi I, Philippe É, Vincent O, Kaaniche NM (2012) Shark: spy honeypot with advanced redirection kit. In: Proceedings of the IEEE workshop on monitoring, attack detection and mitigation. Available <http://spiderman-2.laas.fr/METROSEC/monam.pdf>. Last accessed 17 Aug 2012
8. Anagnostakis KG, Sidiropoulos S, Akrreditis P, Xinidis K, Markatos E, Keromytis AD (2005) Detecting targeted attacks using shadow honeypots. In: Proceedings of the conference on USENIX security symposium, pp 9–23
9. Dandamudi S, Eltaeb T (2015) Firewalls implementation in computer networks and their role in network security. J Multi Eng Sci Technol (JMEST) ISSN: 3159-0040, 2(3), March - 2015
10. Douligeris C, Mitrokotsa A (2004) DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput Netw 44(5):643–666. <https://doi.org/10.1016/j.comnet.2003.10.003>
11. Biondi Philippe (2005) Scapy: explore the net with new eyes. Technical report, EADS Corporate Research Center
12. pfSense Project (2004). <https://docs.netgate.com/pfsense/en/latest>
13. McKinney W (2011) pandas: a foundational Python library for data analysis and statistics. Python High Perform Sci Comput 14(9):1–9
14. Van der Walt S, Colbert SC, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. Comput Sci Eng 13(2):22–30. <https://doi.org/10.1109/mcse.2011.37>
15. Giacobbi G (2014) The GNU Netcat project. <http://netcat.sourceforge.net>
16. Orebaugh Angela, Ramirez Gilbert, Beale Jay (2006) Wireshark and ethereal network protocol analyzer toolkit. Elsevier. <https://doi.org/10.1016/b978-159749073-3/50015-4>

Predicting Student's Performance Using Linear Kernel Principal Component Analysis and Recurrent Neural Network (LKPCA-RNN) Model



Amita Dhankhar and Kamna Solanki

Abstract Technological advancement has led to the generation of a lot of data in all sectors including the educational sector, and it has provided unlimited opportunities to students, teachers, and educational institutions to explore it for their benefits. The exploration and analysis of the educational data to be used to discover and establish significant patterns are called data mining. Educational Data Mining (EDM) is the main concept of data mining used for collecting, examining, and representing the data. The aim of this study is to predict student academic performance from computer courses, using classification and optimization approaches. The data is collected from the OULA (Open University learning analytics) database. For feature extraction (FE), linear kernel PCA is used which extracts the feature vector. Optimization feature selection is performed using the Gini index which checks the distribution of the probability of the specific data point. After the FE process, the training process is done by RNN deep learning classifier which creates an optimized model. The performance is evaluated in terms of sensitivity, specificity, accuracy, and f-measure which perform the predictions for extracting student's performance and displays result in the form of the "pass", "fail". 91.29% accuracy has been achieved by the proposed approach.

Keywords Educational data mining · Feature extraction · Kernel PCA (principal component analysis) with Gini index · RNN (recurrent neural network)

1 Introduction

Digitization is transforming all aspects of our life, and the education sector is no exception [1]. It has brought about drastic changes in teaching and learning techniques. Learner's interaction with online and offline learning techniques have led to generate a lot of data that can be used for analysis purposes [2]. Large-scale attempts

A. Dhankhar (✉) · K. Solanki

Department of Computer Science and Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak, India

have been done for estimating student's achievement for various goals, such as recognizing students with low performance, the chances of student's retaining, and subjects and resources allocation [3].

With the advancement in technology, several educational datasets, and Artificial Intelligence (AI) methods have also increased, and hence, the prediction techniques in EDM gained popularity in 2009 [4]. The purpose of this study is to predict student's academic performance from computer courses, using classification and optimization approaches. The data is collected from the OULA (Open University learning analytics) database. For feature extraction (FE), Linear Kernel PCA is used which extracts the feature vector. The KPCA method is used to reduce the non-linearity in the dimensional space because data is arranged in the multidimensional space and the data points need to map linearly. Optimization feature selection is performed using the Gini index which checks the distribution of the probability of the specific data point. After the FE process, the training process is done by RNN deep learning classifier which creates an optimized model. The performance is evaluated in terms of sensitivity, specificity, accuracy, and f-measure which perform the predictions for extracting student's performance and displays result in the form of the pass, fail. The remaining sections of the paper include related work, methodology, results, conclusion, and future scope.

2 Related Work

This section described several Educational Data Mining (EDM) and Data Mining (DM) methods. The EDM algorithm has predicted academic student performance and evaluates the performance metrics. Various Machine Learning (ML) methods were executed in this area in the past, but it is recent years Deep Learning (DL) got recognition in the academic field. Akçapınar et al., (2019), studied the interconnection information of students in electronic learning for analyzing if the educational achievement of students during the completion of the session can be anticipated in the prior weeks. The outcomes of the analysis showed that the K-nearest neighbor (KNN) technique correctly estimated failed students at the completion of a session with an 89 percent rate [5]. Kumar and Singh, (2019), presented a collaboration of the K-means clustering method with Artificial Neural Network (ANN) as well as the Support Vector Machine (SVM) categorization method to assess the student achievement. The outcomes proved that the accomplishment of ANN as compared to SVM was greater [6]. Kovalev et al., (2020), discussed a novel technique based upon the usage of neural network designs, and hybridization of various techniques [7]. Shrestha and Pokharel (2019) discussed and inspected students' tasks for getting unseen data by utilizing clustering and classification methods. The information was gathered from students registered in a Massive Open Online Course (MOOC) known as C programming provided by Kathmandu University of Nepal. For clustering, the K-means technique was utilized and for classification, an SVM classifier was executed for creating a predictive design that estimated the students' achievement

categorized as low, medium, or high. SVM gave an accuracy of 76.1 percent [8]. Aljohani et al. (2019) have proposed deep LSTM to find out students at-risk by converting the problem into a sequential weekly format. The authors have used the deep LSTM model, SVM, Logistic Regression, ANN for the study. The dataset used for this study was OULA. The proposed model achieved 90% accuracy [9]. Qiu et al. (2018) have predicted dropout by using an integrated framework with feature selection and feature generation. The proposed FSPred Framework used feature selection + logistic regression. The dataset used for the study was XuetangX for KDD CUP 2015 the largest MOOC platform in china. The proposed framework has achieved an F1 score equals to 84.69 [10]. Asif et al. (2017) have predicted the performance of students before the completion of the degree course at Information Technology Engineering University, Pakistan. Analyzed the progress of the students throughout the course and combine them with prediction results. The classification methods used for the evaluations were decision tree, 1-nearest neighbor, Naive Bayes, neural networks, and random forest trees. Out of all these methods, Naive Bayes has achieved the highest accuracy that is 83.6% [11]. Amita et al. (2019) used classification methods to predict the grade of second-semester students by using 10th, 12th, and 1st-semester marks as attributes. The classification methods used were SMOreg, LWL, simple linear regression, random forest, and decision table [12].

3 Research Methodology

This section describes the methodology used in this research paper. Firstly, it describes the dataset that has been used for this research. Then, the pre-processing, feature extraction, feature selection which have been applied to the dataset to prepare it have been explained in detail. After the preparation of the data, the classification method has been applied to predict the student's performance.

3.1 Dataset

In this research work, data is collected from the OULA (Open University learning analytics) database [13]. In this paper, we have created a database, consisting of information from programs demonstrated at the Open University (OU). The reason that the database is distinct is that it consists of demographic information along with a collection of click-stream information of student's activities in the Virtual Learning Environment (VLE). It allows recognizing the responses of the student, as shown by their performance.

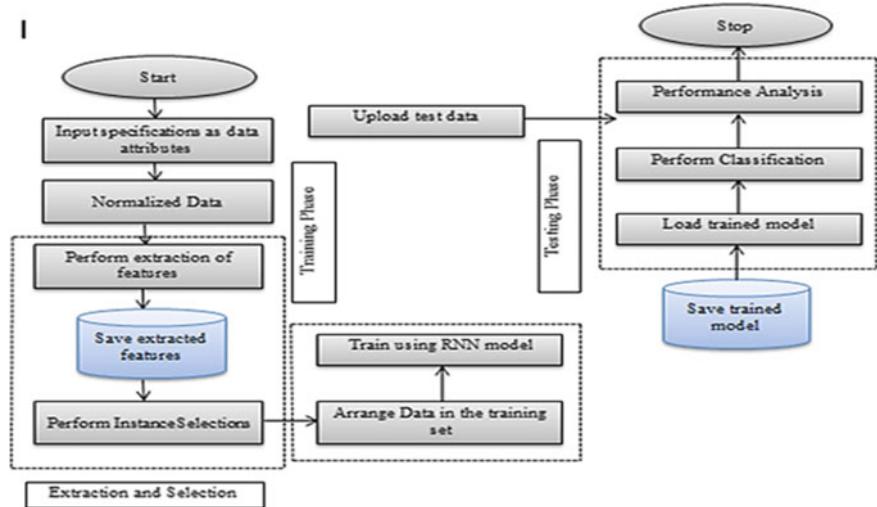


Fig. 1 Flow chart of proposed research work

3.2 Proposed Model

To predict the student's performance, LKPCA-RNN has been used. In this method, we have applied Linear Kernel PCA for feature extraction (FE), which extracts the feature vector. Then optimization feature selection is performed using the Gini index, Information Gain, Weights which checks the distribution of the probability of the specific data point. After the FE process, the training process is done by RNN deep learning classifier which creates an optimized model. A detailed explanation is as follows and Fig. 1 depicts the flowchart of the proposed model.

3.2.1 Data Pre-Processing and Feature Extraction (Linear Kernel PCA)

The first step is to upload the data at the backend and gets stored in the MATLAB environment. The data is then processed for the feature extraction process. The pre-processing step will arrange the data in the multidimensional matrix and is processed smoothly for efficient computations. Feature engineering is a process that uses domain knowledge to extract the properties from raw data. These properties may be used to enhance the system's performance. The PCA extracts the feature vector. The PCA will reduce the non-linearity in the dimensional space because data is arranged in the multi-dimension space and the data points need to map linearly so that there will be proper decision boundaries' during classification. If the data is nonlinear then the data points will not be arranged in their respective distances which can increase the variance and standard deviations on the same.

3.2.2 Optimization Feature Selection Using Gini Index

Further, it is optimized using instance selection which is done using Gini index, weights, and information gain which extracts the relevant instances from the data. Gini index will measure the distribution process and checks the distribution of the probability of the specific data point. Weights will measure the connections among the data points to transfer the information to the classification model.

3.2.3 Classification RNN

After the feature extraction process, the training process is done using RNN deep learning classifier through which the network learns the process and train itself to generate an optimized model. RNN will use the LSTM networks which make use of the logic gates to store and process the information through nodes in the network. The last phase is the testing phase (RNN classifier model) in which the test data which is unknown data is uploaded and processes on which the prediction is performed. The trained model is loaded, and the test data is classified based on the information passing among nodes to perform the classification in terms of student performances. After the prediction process, the performance is evaluated in terms of sensitivity, specificity, accuracy, and f-measure.

- **Pseudocode of the Proposed Model.**

Pseudocode: LKPCA-RNN

```

Begin
Input Specifications  $I_x = \{I_1, I_2, \dots, I_n\}$ 
For  $i = 1: N$ 
     $N(x) = [\text{rearrange } \{I_x\}]$ 
    End for
Extract features as eigen values  $E(x) = \text{eig } \{N_1(x), N_2(x), \dots, N_n(x)\}$ 
Perform instance selections such that
    For  $I = 1: [E(x)]$ 
         $I_s(x) = G_x[E(x)]$ 
         $= I_G[E(x)]$ 
         $= w_x[E(x)]$ 
    End for
Where  $G(x)$ ,  $I_G$ , and  $w_x$  as the GI (Gini Index), IG (Information Gain), and WC (Weight Correlations).
Train model as training  $I_s(x) \rightarrow \text{RNN model.}$ 
    For  $i = 1: I_s(x)$ 
         $MD(x) = \text{Train network } [\{T(x)\}]$ 
    End for
Generate test set  $T_1, T_2, \dots, T_N$ .
 $P_D = \text{Classify } (MD(x), T_N)$ .
Evaluate the performance metrics
Exit

```

4 Results and Discussion

This proposed work has designed a desktop application using the MATLAB simulation tool. This tool presented the GUI (Graphical User Interface) toolbox. This research work has worked on two modules as the Training and Testing module. The training phase is implemented using the GUI toolbox and is made using GUI tools that show the list boxes, pushbuttons, edit text. It will make the user interface for the man-machine interactions which are used for the user to click and see the output easily. Figure 2 shows the training process in which the system is trained using RNN. It shows that because the count of iteration enhances, the accuracy of the system is enhancing which shows that the system is achieved high learning performance in training the system. Also, it can be noticed that the loss is also decreasing which

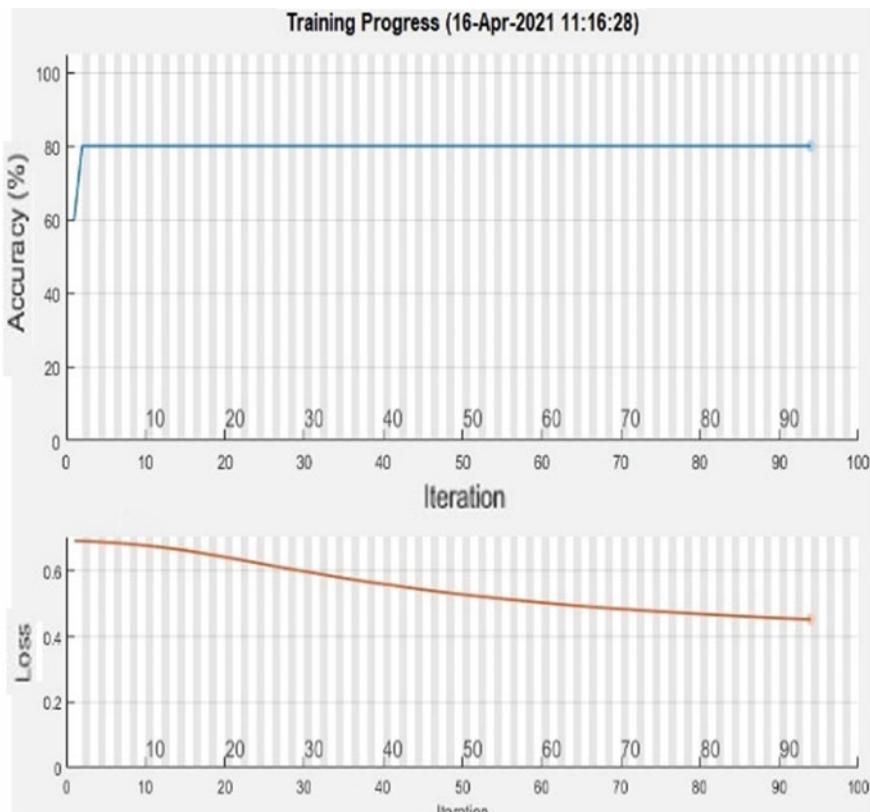


Fig. 2 Training process (TP)

means the training error also decreases. The loss must be low for high accuracy which reduces the overfitting and underfitting of the training model. As it can be seen that the parameter based on the accuracy of the training model is completely dependent on the loss of the system that can lead to the performance of the classification result during the testing phase.

Below Fig. 3 shows the classification outcome which can be seen in the form of labels. It is noticed that the suggested method is able to classify the test data for the individual as pass, fail or withdrawn based on past performance. These are evaluated when the trained RNN model is implemented on the test set and shows the predicted labels based on the test set.

Figures 4, 5, 6, 7 show the performance evaluation using the accuracy, F-measure, sensitivity, and specificity of the proposed approach in comparison with KNN. The proposed approach has clearly achieved more classification accuracy, F-measure, sensitivity, and specificity than the KNN.

Table 1 shows the performance evaluation of the proposed (LKPCA-RNN) model and the existing KNN model. This analysis shows that the proposed hybrid

Fig. 3 Classification outcome

Pass
Pass
Withdrawn
Pass
Fail
Pass
Pass
Pass
Pass
Withdrawn

Fig. 4 Comparison between LKPCA-RNN (proposed model) and KNN (existing model): accuracy rate (%)

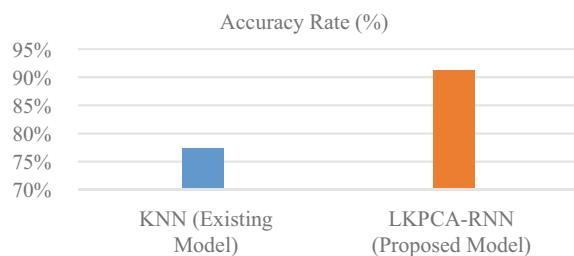


Fig. 5 Comparison between LKPCA-RNN (proposed model) and KNN (existing model): f-measure

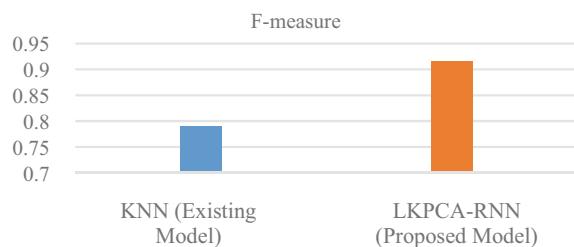


Fig. 6 Comparison between LKPCA-RNN (proposed model) and KNN (existing model): sensitivity

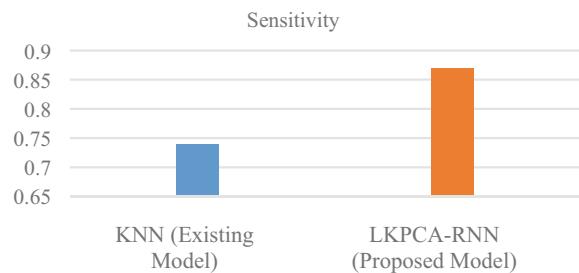


Fig. 7 Comparison between LKPCA-RNN (proposed model) and KNN (existing model): specificity

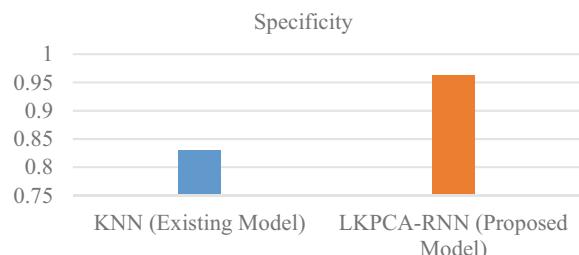


Table 1 Performance evaluations

Parameter	KNN model (existing)	LKPCA-RNN model (proposed)
Accuracy	77.48%	91.29%
F-measure	0.79	0.916
Sensitivity	0.74	0.871
Specificity	0.83	0.963

linear kernel RNN model has achieved accuracy rate 91.29 percent, specificity value 0.963, F-score or F-measure value 0.916, and sensitivity value 0.871. It is evident from the comparative performance analysis that proposed LKPCA-RNN model outperform the KNN.

5 Conclusion and Future Scope

This research work has predicted student academic performance from computer courses. The data is processed for the feature extraction process by using the Linear Kernel PCA technique. After that, optimization is done using Gini index which extracts the relevant instances from the information. The training process is done using RNN deep learning classifier, once the feature extraction process is completed. RNN creates an optimized model. The performance evaluation is done in terms of Sensitivity (0.871), Specificity (0.963), Accuracy (91.29 percent), and f-measure

(0.916). The proposed method is able to classify the test data for the individual as pass, fail, or withdraw better in comparison with other methods.

In the future, the hybrid method of classification using Naïve Bayes (NB), Support Vector Machine (SVM), decision trees (J48) could be applied for better accuracy and to predict the result of students. It can also be used for developing strategies in education for enhancing performance. CNN technique can be used for performing binary classification on temporal educational data. A CNN with data imbalance technique can be used for the course-level student prediction task which can provide a more accurate and early stage prediction. This technique will help to improve the existing problems. Hence, the proposed model can be extended for multiple class classification. Such extension is significant because of data imbalance and overlapping of temporal educational information. Additionally, integrating such designs into the educational decision support system is even useful for decision-making support.

References

1. Chae BK (2019) A General framework for studying the evolution of the digital innovation ecosystem: the case of big data. *Int J Inf Manage* 45:83–94
2. Dhankhar A, Solanki K, Dalal S, Omdev (2021) Predicting students performance using educational data mining and learning analytics: a systematic literature review. *Innov Data Commun Technol Appl* 127–140
3. Chui KT, Fung DCL, Lytras MD, Lam TM (2020) Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Comput Hum Behav* 107:105584
4. Dhankhar A, Solanki K (2021) Comparative analysis of various techniques used for predicting student's performance. In: Proceedings of the workshop on technological innovations in education and knowledge dissemination (WTEK 2021), CEUR workshop proceedings, vol 2869, pp 10–24. ISSN 1613-00731
5. Akçapınar G, Altun A, Aşkar P (2019) Using learning analytics to develop early-warning system for at-risk students. *Int J Educ Technol High Educ* 16(1):1–20
6. Kumar M, Singh AJ (2019) performance analysis of students using machine learning & data mining approach. *Int J Eng Adv Technol* 8(3):75–79
7. Kovalev S, Kolodenkova A, Muntyan E (2020) Educational data mining: current problems and solutions. In: 2020 V international conference on information technologies in engineering education (Inforino). IEEE, pp 1–5
8. Shrestha S, Pokharel M (2019) Machine learning algorithm in educational data. In: 2019 artificial intelligence for transforming business and society (AITB), 1–11, IEEE
9. Aljohani NR, Fayoumi A, Hassan SU (2019) Predicting at-risk students using clickstream data in the virtual learning environment. *Sustainability* 11(24):7238
10. Qiu L, Liu Y, Liu Y (2018) An integrated framework with feature selection for dropout prediction in massive open online courses. *IEEE Access* 6:71474–71484
11. Asif R, Merceron A, Ali SA, Haider NG (2017) Analyzing undergraduate students' performance using educational data mining. *Comput Educ* 113:177–194
12. Dhankhar A, Solanki K, Rathee A, Ashish (2019) Predicting student's performance by using classification methods. *Int J Adv Trends Comput Sci Eng* 8(4):1532–1536
13. Kuzilek J, Hlosta M, Zdrahal Z (2017) Open university learning analytics dataset. *Sci Data* 4(1):1–8

Efficient Spectrum Allocation in Wireless Networks Using Channel Aggregation Fragmentation with Reservation Channels



N. Suganthi and K. Suresh Kumar

Abstract With the rapid development of wireless devices and applications, scarcity in the spectrum band arises. Hence, Cognitive Radio Network (CRN) have emerged that provide fair spectrum sharing among all the users in the system. During spectrum allocation, the key challenge in CRN is to provide good Quality of Service (QoS) to secondary users measured by parameters like blocking probability and forced termination probability. The work introduces fair spectrum allocation methods that use channel aggregation fragmentation and queuing models to improve secondary users' QoS. The system uses three different spectrum allocation methods: aging, round robin priority (RRP), and RRP with reservation algorithms. The system is tested under different scenarios in these three methods, and the QoS are compared on these methods.

Keywords Cognitive Radio Networks · Spectrum allocation · Quality of service

1 Introduction

According to ITU estimation, 53.6% of the global population or 4.1 billion people use the Internet at the end of 2019 [1]. Simultaneously, statistics show that mobile cellular telephone subscriptions are also getting increased year by year. As mobile subscribers and wireless applications get increased, the quality of service (QoS) provided to end-users must be well defined.

Wireless networks used a fixed spectrum assignment policy, where spectrums are assigned to licensed holders by governmental agencies for long-term use in different

N. Suganthi (✉)

Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Ramapuram Campus, Chennai, India

e-mail: suganthn@srmist.edu.in

K. S. Kumar

Associate Professor, Department of IT, Saveetha Engineering College, Saveetha Nagar, Thandalam, Chennai, India

e-mail: sureshkumar@saveetha.ac.in

geographical locations. According to the Federal Communications Commission (FCC) [2], spectrum usage on these licensed portions was non-uniform. Spectrum usage was more on certain portions, while some portions of spectrum remained unutilized. As a result, most of the allocated spectrum remains unoccupied or under-utilized [3]. The utilization rate of assigned, licensed spectrum ranged from 15 to 85% and in specific below, 3GHZ; only 5.2% of bands are used at a specific time or location. Also, a spectrum occupancy measurements study performed at different locations Malaysia [4], China [5], Pakistan [6], and India [7] shows that most parts of the licensed spectrum bands are underutilized. However, the unlicensed portion of spectrum bands are overcrowded and congested [8], and licensed portions are minimally used.

Although a fixed spectrum assignment policy worked well in the past, there is an abundant increase in wireless devices and their applications. The unavailability of unlicensed spectrum resource and the limited usage in the licensed spectrum band triggered a new communication protocol [9]. Cognitive Radio Networks (CRN), also known as Dynamic Spectrum Access Networks (DSAN), has emerged to use intelligent radios. 5G technology, which is expected to be in 2020 [10], aims to provide high-speed uploads and downloads with low latency. Spectrum sharing is one of the key factors to promote 5G technology, and spectrum sharing in Cognitive Radio Networks is a promising technology that propels 5G networks in the future [11].

Cognitive Radios are software-defined radios that are programmed and dynamically designed to use the best available channels. According to its environments in real time, it is radio conscious of its environment and skilled in varying its operating parameters to provide stable wireless spectrum communication anywhere anytime efficient wireless spectrum communication [12].

CRN consists of two types of users: primary users or licensed users and secondary users or unlicensed users. There are two main features for Cognitive Radio Networks: (i) Cognitive radio has the cognitive capability that enables it to detect the unexploited portions of the spectrum at a particular time and location and helps to choose the best spectrum and operating parameters. (ii) Cognitive radio has reconfigurability that enables the radio to be dynamically programmed to perform data communication at diverse frequencies and dissimilar access methodologies [13].

Cognitive Radio Networks require additional spectrum management functionalities since they coexist along with the primary networks. The main design challenges which include (i) interference avoidance with a primary network, (ii) enabling QoS aware communication with dynamic spectrum requirement, and (iii) providing uninterrupted communication.

To efficiently utilize the maximum amount of the available spectrum and equip the above design challenges, Cognitive Radio Network supports four additional spectrum management functions. Figure 1 represents the various spectrum management functions. The first function performed by CRN is spectrum sensing. By spectrum sensing functionality, CRN can identify the unutilized portion of the spectrum called spectrum holes. The second function performed by CRN is spectrum decision.

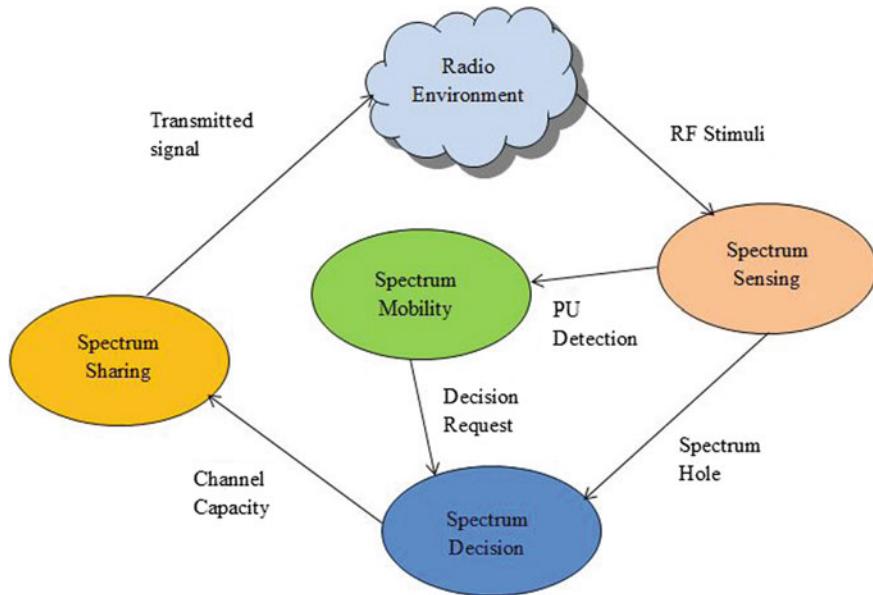


Fig. 1 Cognitive Radio functions

By spectrum decision functionality, CRN can select the best channel among all the available channels by evaluating the channel properties. The third function performed by CRN is spectrum sharing. By spectrum sharing functionality, users in CRN can fairly share the available spectrum without any interference from other network users. Either the spectrum is shared among two secondary users or between a primary user and a secondary user. The fourth function performed by CRN is spectrum mobility. The secondary users in the CRN can perform spectrum handoff by spectrum mobility functionality to vacate the current channel and shift to other available channels if its registered primary user needs the current channel.

2 Related Work

Two-channel assembling strategies are proposed by Jiao et al. in 2012 [14] and Lee et al. in 2010 [15] for channel allocation to secondary users in the network. The strategies include spectrum adaptation and heterogeneous SU traffic. Static channel assembling and dynamic channel assembling are proposed. In static, the same number of channels are aggregated, while in dynamic, a different number of channels are aggregated. A dynamic strategy is proved to be better when compared to a static strategy.

Rahim et al. in 2020 [16] proposed an efficient channel allocation method for SUs based on the best match between the SUs and the available channels considering

the QoS of SUs and the channel's quality. Gale Shapley matching theory was used in channel allocation to select a channel from the resource pool to satisfy the QoS constraint of SUs. Zakariya et al. in 2020 [11] has proposed a novel multiclass SU CR model. In the first case, priority-free among SU is considered, and in the second case, the priority-based SU model is considered. Performance metrics like extended data delivery time and system time are calculated. When SU is interrupted by PU, it waits in the queue of the corresponding channel. The priority-based model shows improvement in network system time.

A spectrum sharing scheme and the successive interference cancellation (SIC) technique are proposed by Zhai et al. in 2020 [17]. Interference cancellation is done either by direct (DIR) method or SICS method at both primary receiver (PR) and secondary receiver (SR). Chakraborty et al. in 2020 [18] proposed a machine learning-based three-phase Target Channel Sequence (TCS) channel allocation scheme. It provides unique TCS for every real-time SUs within a limited time-bound. It also predicts the states of the channel intelligently. The allocation scheme periodically updates the TCS with the help of users and channel allocation vector. As a result, it reduces the call drops and minimizes the cumulative channel idle time. The limitation is that it provides TCS only for real-time SUs, which leads to starvation of non-real-time SUs.

A queuing model for multichannel Cognitive Radio Network is proposed by Balapuwaduge et al. [19]. The scheme employs a channel assembling strategy for channel allocation to secondary users. Two queuing schemes are used in the system to handle heterogeneous SU traffic with different priorities. This leads to the starvation of low-priority SUs in the network. The QoS provided to users in CRN can be measured by various performance metrics. The performance metrics considered in the research work are blocking probability (BP), forced termination probability (FTP), average waiting time (AWT) of SUs, and overall spectrum utilization [14, 19, 20].

3 System Model

The work focuses on the fair scheduling of channels among the SUs and improving QoS to SUs in CRN. The system uses enhanced channel aggregation and fragmentation techniques to efficiently share and use the channels among SUs in the network. The system uses queuing models to store the SUs in the buffer and avoid or reduce the dropping of SUs from CRN. The algorithms for spectrum allocation, namely aging and round robin priority (RRP), aim to reduce starvation and delay SUs in CRN. The reservation algorithm aims to provide continuous, uninterrupted service to both PUs and SUs in CRN.

When a SU requests service through Central Base Station (CBS), it checks for channel availability. If channels are available, then SU can be allocated with the channel. If channels are not available, the system tries to perform spectrum adaptation [21]. By spectrum adaptation, ongoing SUs help the incoming SU by adjusting its channel. If spectrum adaptation is possible, then the incoming SU can be allocated

with the channel. If spectrum adaptation is not possible, SU is passed to the queue selector. Based on the type of data, the SU is to transmit, the queue selector classifies it as real-time and non-real-time and places it in the corresponding queue. RRP algorithm aims to improve the priority of all non-real-time SUs and make them progress some portion of their communication. RRP with reservation algorithm has some portion of the spectrum as reserved bands and uses those reserved bands in emergency cases.

The users in the network are primary users and Secondary users. Secondary users with heterogeneous traffic types are considered in the network. For example, SUs with real-time traffic is referred to as real-time secondary user (RSU) and SUs with non-real-time traffic are referred to as non-real-time secondary user (NRSU). Hence, the users are primary user (PU), RSU, and NRSU. The users' priority is that the PU has the highest priority among all users and can preempt RSU and NRSU. RSU has the next highest priority and can preempt NRSU. NRSU has the least priority among all the users.

The system uses a queuing model with a first come first serve strategy. The model maintains two queues, namely real-time queue (RQ) and non-real-time queue (NRQ). The queues are introduced to buffer the SUs to wait for their opportunity to access the channel. When the SUs fail to acquire a channel in the network, then the queue selector classifies the SU as RSU or NRSU based on its data traffic. The SUs classified as RSU are made to wait in the RQ, and SUs classified as NRSU are made to wait in NRQ.

The SUs in the queues are served in a first come first serve strategy where the SU that entered into the queue first will be given a chance to access the channel when some idle channels become available in the network. According to the user's priority levels, RSU in RQ will have higher priority than NRSU in NRQ. Hence, if idle channels are available, RSU from RQ will be given the opportunity. Only if the RQ is empty, the NRSU from NRQ will get its chance of channel access. This general strategy of queuing model raises the starvation of NRSU in NRQ. Let j_{lq} and j_{hq} represent the length of the queue NRQ and RQ, respectively. Let j_{rq} and j_{nrq} represent the available space in the queue, RQ and NRQ where $j_{rq} \leq j_{hq}$ and $j_{nrq} \leq j_{lq}$.

The arrival rates of users are Poisson distribution and are represented as λ_P , λ_{RS} , λ_{NRS} for the primary user, RSU, and NRSU. The users' service times are exponentially distributed and are represented as μ_P , μ_{RS} , μ_{NRS} for the primary user, RSU, and NRSU, respectively.

4 Aging Algorithm

The aging algorithm [22] aims to reduce the starvation of NRSU. Whenever a secondary user requests channel for data transmission, dynamic spectrum allocation is invoked. If spectrum adaptation fails, then the secondary users are not blocked and are classified into RSU and NRSU based on the type of data it is to transmit and

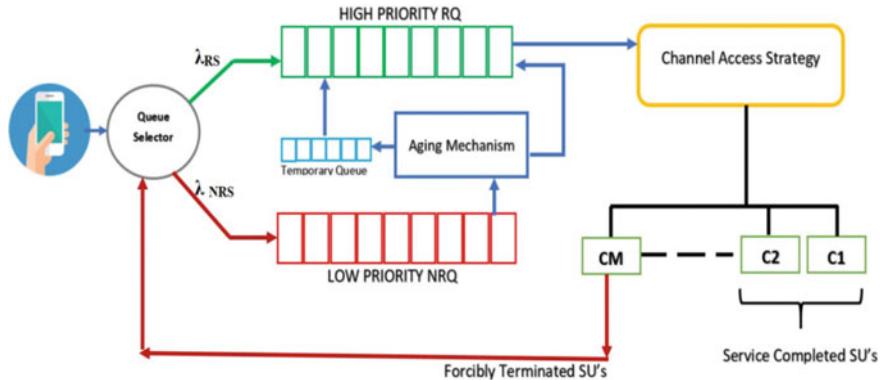


Fig. 2 Aging algorithm

placed in the corresponding queue, as shown in Fig. 2. If SU is RSU, it is placed in a real-time queue (RQ) in a non-real-time queue (NRQ).

By the queuing model and classification, starvation in NRSU, i.e., it gets spectrum only in the absence of RSU in RQ. The aging algorithm aims to reduce the starvation of NRSU and increase the network's overall performance. The aging algorithm specifies a predefined threshold period that defines the waiting time of NRSU in the front of the NRQ. If an NRSU waits for more than this predefined threshold period, the aging algorithm triggers and moves the NRSU in front of NRQ to RQ if space is available in RQ else moves it onto a temporary queue. When an RSU is dequeued from RQ, then NRSU from the temporary queue is moved to the end of RQ. Here, the aging algorithm increases the priority of NRSU in front of NRQ than the next incoming RSU in the system.

Algorithm: Aging (NRQ).

```

1: Begin
2: Dequeue (SUi, NRQ);
3: SUT.NRQ-Out (SUi) = CT; // SUT- SU Table, CT—Current Simulation time
4: if (jrq > = 1) then
5: Enqueue (SUi, RQ);
6: SUT.RQ-In (SUi) = CT;
7: else
8: Enqueue (SUi, temp_queue);
9: While (1)
10: Check (RQ);
11: end if
12: end
13: Check (RQ)
14: Begin
15: if (jrq > = 1) then
16: Dequeue (SUi, temp_queue);
  
```

```

17: Enqueue ( $SU_i$ , RQ);
18:  $SUT.RQ\text{-In} (SU_i) = CT$ ;
19: end if
20: end

```

5 RRP Algorithm

The RRP algorithm [23] aims to reduce the starvation of NRSU. Whenever a secondary user requests channel for data transmission, dynamic spectrum allocation is invoked. If spectrum adaptation fails, then the secondary users are not blocked and are classified into RSU and NRSU based on the type of data it is to transmit and placed in the corresponding queue, as shown in Fig. 3. If SU is RSU, it is placed in a real-time queue (RQ) in a non-real-time queue (NRQ).

By the queuing model and classification, starvation in NRSU, i.e., it gets spectrum only in the absence of RSU in RQ. RRP algorithm aims to reduce the starvation of all NRSU and increase the network's overall performance.

As shown in Fig. 5.1, the RRP algorithm has two phases. In phase 1, least allocation round robin (LARR) algorithm is used, and in phase 2, the most allocation priority (MAP) algorithm is used. During phase 1 and phase 2, all the ongoing RSU services are suspended, and the RRP algorithm uses the channels released by them. In phase 1, all the waiting NRSU from NRQ are taken, and they are provided with a minimum (W) number of channels for the time quantum specified in the round robin algorithm. In round robin fashion, all NRSU's get a chance of W channels for the T time quantum period. Once all NRSU's gets completed, then phase 1 ends.

Algorithm Phase 1: Least_Allocation_RR(NRQ).

```

1: Begin
2: for each  $SU_i$  from NRQ
3:  $N = N_{\text{release1}} / W$ ;
4: Allocate  $W$  channels to  $N$  number of  $SU_i$  in each round
5: end for

```

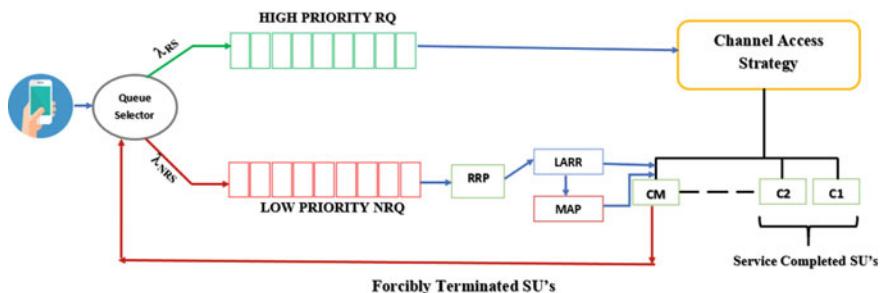


Fig. 3 RRP algorithm

6: end

In phase 2, NRSU's are assigned new priorities based on the remaining data yet to be transmitted by them. The NRSU with the least amount of data to be transmitted has the highest priority, and NRSU with more data to be transmitted has the least priority. Using the most allocation priority scheduling algorithm, the NRSU's with the highest priority gets the maximum (V) number of channels. The next highest priority NRSU gets the channels only if already allocated highest priority NRSU's completes their transmission and releases the channels. By this RRP algorithm, the system ensures the service completion of NRSU with a small service time, and the remaining NRSU at least make progress in its data transmission.

Algorithm Phase 2: Most_Allocation_Priority (NRQ).

//Assign priorities—a high priority for less amount of data to be transmitted.

```

1: Begin
2: for each prioritized  $SU_i$  from NRQ
3:  $N = N_{\text{release1}} / V;$ 
4: Allocate  $V$  channels to  $N$  number of  $SU_i$  according to priority.
5: end for
6: End

```

6 RRP with Reservation Algorithm

RRP algorithm aims to allocate spectrum to SUs such that it reduces the starvation of SUs in the network. However, there are two limitations in the RRP algorithm—(i) suspends the execution of RSU and gives a chance for NRSU to progress, (ii) employs channel aggregation alone to allocate channels to SUs. Some other alternate mechanisms can be included to handle such suspended RSU transmissions. Hence, the third phase of the research work focuses on handling such suspended RSU and providing continuous, uninterrupted service to all the network users. Also, it employs channel aggregation and fragmentation simultaneously to allocate channels to SUs. By CAF, spectrum allocation utilizes the maximum spectrum.

6.1 Service Interruptions

Service given to a user in the network can be interrupted due to any one of the following reasons:

- (i) Any high-priority user can preempt service given to a user. For example, services given to SUs can be preempted by high-priority PU and services given to NRSU can be preempted by high-priority RSU. When preempted, the transmission gets interrupted.

- (ii) The next case when transmission gets interrupted is because of channel failures.
- (iii) The third reason for service interruption is the RRP algorithm in the second phase of the research work.

To handle all the three cases mentioned above, which interrupts the transmissions and provides continuous service to all the users, the third phase includes reservation scheme. The total spectrum bands are divided into reserved channels (RC) and non-reserved channels (NRC). The non-reserved bands are used in spectrum allocation. The reserved bands are used to provide continuous, uninterrupted services to SU and PU in the system. The amount or portion to be reserved can be static or dynamic. In static, the number of reserved bands is fixed irrespective of the traffic flow in the network. In dynamic, the number of reserved bands gets varied according to the rate of traffic flow in the network.

6.2 Static Reservation Scheme

The research work uses a static reservation scheme. The portion of spectrum band that can be reserved should be carefully considered. If more portion of the spectrum band is given for reservation, then the overall blocking probability gets increased. Hence, RRP with reservation scheme allocates 20% of the spectrum for reserved bands, and the remaining 80% of the spectrum band is used as non-reserved bands. Thus, spectrum allocation for the users in the network is completed and allocated from the non-reserved bands.

The reserved bands are used only in three specific cases. The cases are as follows:

- (i) Whenever high-priority users preempt a secondary user, the secondary users are forced to terminate. The SU's that are forced to terminate can use the channels in the reserved band by the reservation schemes.
- (ii) Whenever a channel failure occurs in the non-reserved band, alternate channels from reserved bands are allocated to overcome those failures.
- (iii) When the RRP algorithm is executed, it suspends the RSU in the system. And by the reservation scheme, the reserved band's channels can be used by those suspended RSU.

Figure 4 represents the cases when reserved channels are used. On SU arrival, if enough channels are available in the NRC, then SU starts its service. On the other hand, if enough channels are not available, it follows the RRP algorithm procedure.

During transmission in non-reserved channels, there are three cases during which a transmission gets shifted from NRC to RC. First, if channel failure is true in NRC, the transmission is moved to RC. If RSUs are suspended by RRP, the suspended RSUs shift to RC. The RSUs that entered the network first will get RCs for their uninterrupted communication. Third, if NRSU gets preempted by RSU and RSU gets preempted by PU, they can get channels from NRC.

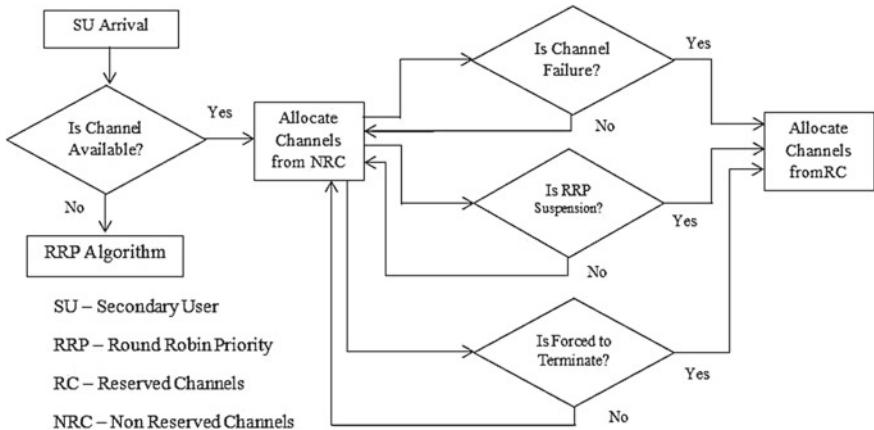


Fig. 4 Flowchart for RRP with reservation algorithm

6.3 Channel Access Strategy

The channel access strategy is dynamic in CRN, and when a reservation scheme is included, the channel access strategy is based on the availability of channels in RC and NRC.

6.3.1 PU Arrival

Upon PU arrival, it checks for channel availability in NRC. If the channel is available, it starts its transmission. If the channel is not available, then it preempts a SU in the system. Next, the interrupted SU tries for spectrum adaptation. If spectrum adaptation is successful, then it can continue its service. If spectrum adaptation is not possible, it performs spectrum handover and tries to get RC channels. If enough channels are not available in RC, then the interrupted SU is forced to terminate from the system.

6.3.2 SU Arrival

Upon SU arrival, it checks for V to W number of channels from NRC to start its service. If channels are available in NRC, SU can start its service; else, it tries to occupy the buffers in RQ or NRQ based on its classification as RSU and NRSU. Then, the SUs gets their channel allocated by the RRP algorithm. Finally, RSUs suspended by RRP competes and gets channel from RC.

6.3.3 SU or PU Departures

Whenever SU or PU completes its service and exits the network, it triggers spectrum handover from RC to NRC. On the other hand, if any communication is going on in RCs, it gets shifted to the freed channels in NRC with preference given to PU for spectrum handover from RC to NRC.

6.3.4 Channel Failures

Channel failures are one of the main reasons for service interruption in CRN. The channel failure may arise in the channel used by PU or SU. If a SU is using the failed channel, it gets interrupted only when aggregated channels become less than W. If not, it can continue its service. If aggregated channels become less than W, then SU tries to perform spectrum adaptation in NRC. If spectrum adaptation is not possible, it performs spectrum handover to RC. If a PU is using the failed channel, then it preempts SU in NRC. The preempted SU tries spectrum adaptation. Suppose not successful tries to perform spectrum handover to RC.

7 Numerical Results and Discussions

The system impartially allocates the spectrum among all users in CRN. As a result, the QoS of secondary users is increased through fair allocation, and the network's overall performance gets increased. The system is simulated in Omnet ++ and NS2 with the help of the CRCN protocol. The system is evaluated by performance measures like blocking probability and forced termination probability.

7.1 Blocking Probability

Blocking probability is defined as the rate at which a secondary user gets blocked on requesting channels for its communication. Blocking probability is defined as the ratio of the total number of secondary users blocked to the total number of requests made by primary and secondary user. It occurs when a secondary user requests the channel.

Equation (1) is the blocking probability represented as the ratio of the number of SUs blocked to the new arrival of users PU and SU in the network. Equation (2) is the formulae to calculate the blocking probability where the arrival rates of PU (λ_P), RSU(λ_{RS}) and NRSU(λ_{NRS}) are considered in finding out the number of SUs blocked and to calculate the blocking probability.

$$BP_{SU} = \frac{\text{Number of SU's blocked}}{\text{Arrival rate of SU's and PU's}} \quad (1)$$

$$BP_{SU} = \frac{SU_{Blocked}}{\lambda_P + \lambda_{RS} + \lambda_{NRS}} \quad (2)$$

Figure 5 gives a graphical representation of the comparison of blocking probabilities in different scenarios in different spectrum allocation algorithms. On comparing the five different allocation methods, RRP with reservation scheme has the least blocking probability in all the scenarios.

Figure 6 gives a pictorial representation of the average blocking probability calculated in five different spectrum allocation methods. Figure 7 represents the percentage difference in the blocking probabilities between the two schemes used for spectrum allocation. It gives a comparison between the proposed schemes and already existing CCAS and VCAS. The aging scheme shows a 12.56% reduction in BP than CCAS and a 9.08% reduction compared with VCAS. The RRP scheme shows an 18.62% reduction in BP compared with CCAS and a 15.14% reduction compared with VCAS. The RRP with Reservation scheme shows a 22.8% reduction in BP than CCAS and a 19.32% reduction compared with VCAS.

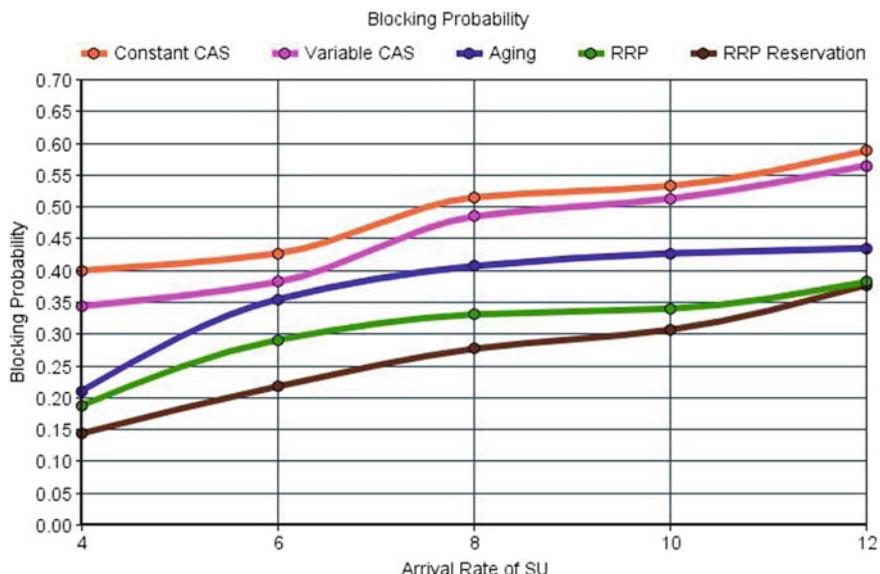


Fig. 5 Blocking probability

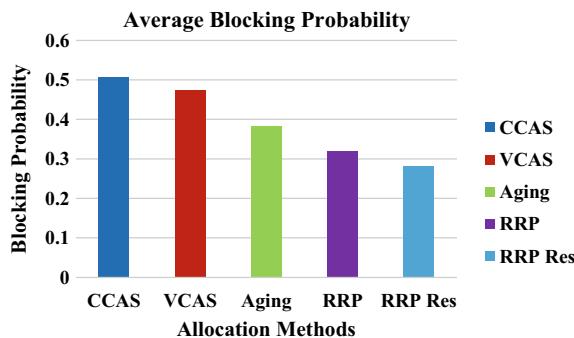


Fig. 6 Average blocking probability

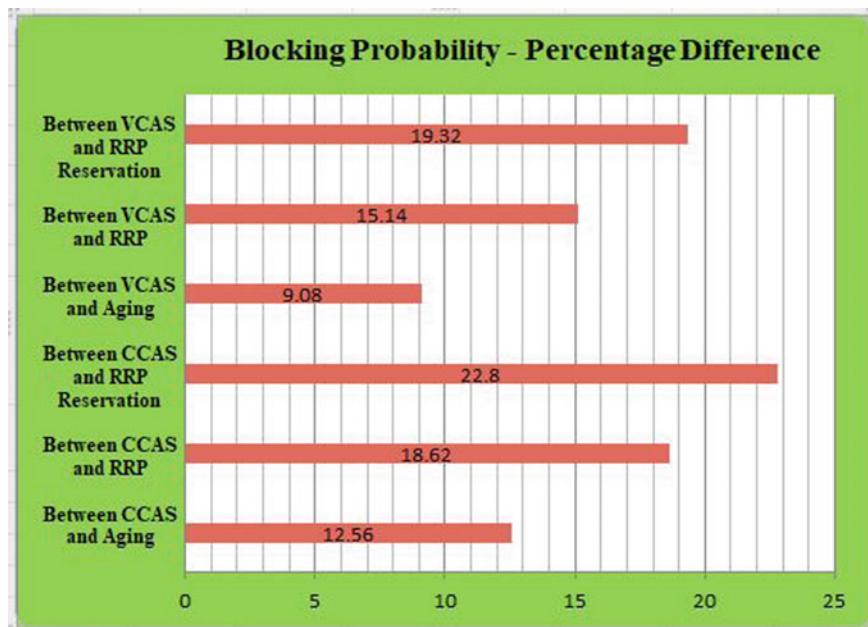


Fig. 7 Blocking probability—percentage difference

7.2 Forced Termination Probability

Forced termination probability (FTP) is defined as the rate at which the ongoing secondary users are forced to terminate the spectrum by high-priority users. Forced termination probability is defined as the ratio of the total number of secondary users forced to terminate the communication to the total number of requests made by

primary and secondary user. It occurs on an ongoing user when a high-priority user makes a request and preempts a low-priority user.

Equation (3) is the forced termination probability represented as the ratio of the number of SUs terminated to the new arrival of users PU and SU in the network. Finally, Eq. (4) is the formula to calculate the forced termination probability where the arrival rates of PU (λ_P), RSU(λ_{RS}), and NRSU(λ_{NRS}) are considered in finding out the number of SUs terminated and to calculate the forced termination probability.

$$\text{FTP}_{\text{SU}} = \frac{\text{Number of SU's terminated}}{\text{Arrival rate of SU's and PU's}} \quad (3)$$

$$\text{FTP}_{\text{SU}} = \frac{\text{SU}_{\text{terminated}}}{\lambda_P + \lambda_{\text{RS}} + \lambda_{\text{NRS}}} \quad (4)$$

Figure 8 gives a graphical representation of the comparison of forced termination probabilities in different scenarios in different spectrum allocation algorithms. On comparing the five different allocation methods, RRP with the reservation scheme has the least forced termination probability in all the scenarios.

Figure 9 gives a pictorial representation of the average forced termination probability calculated in five different spectrum allocation methods. Figure 10 represents the percentage difference in the forced termination probabilities between the two schemes used for spectrum allocation. It gives a comparison between the proposed schemes and already existing CCAS and VCAS. The aging scheme shows a 7.14% reduction in FTP than CCAS and a 4.6% reduction compared with VCAS. The

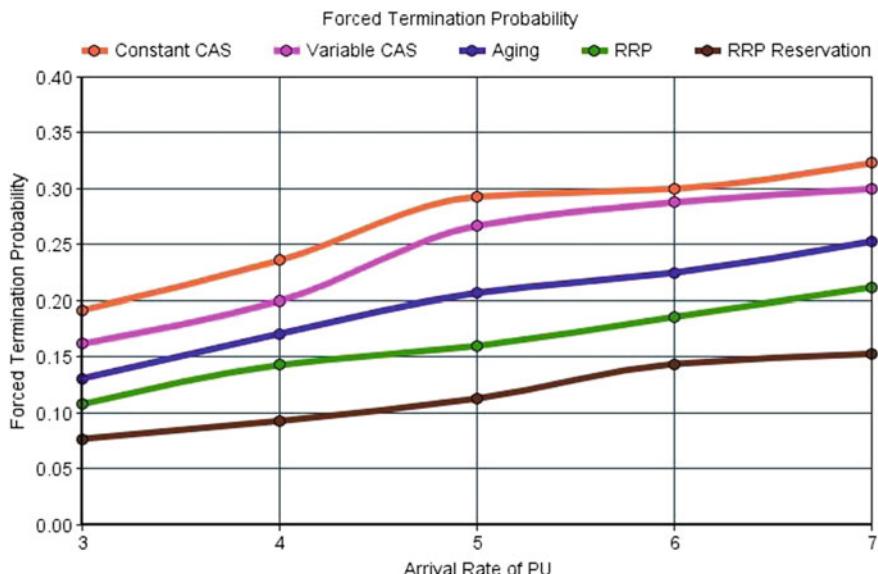


Fig. 8 Forced termination probability

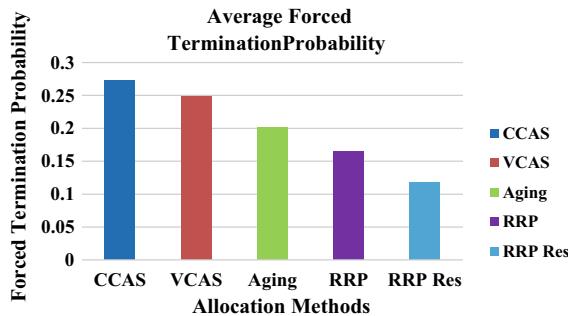


Fig. 9 Average forced termination probability

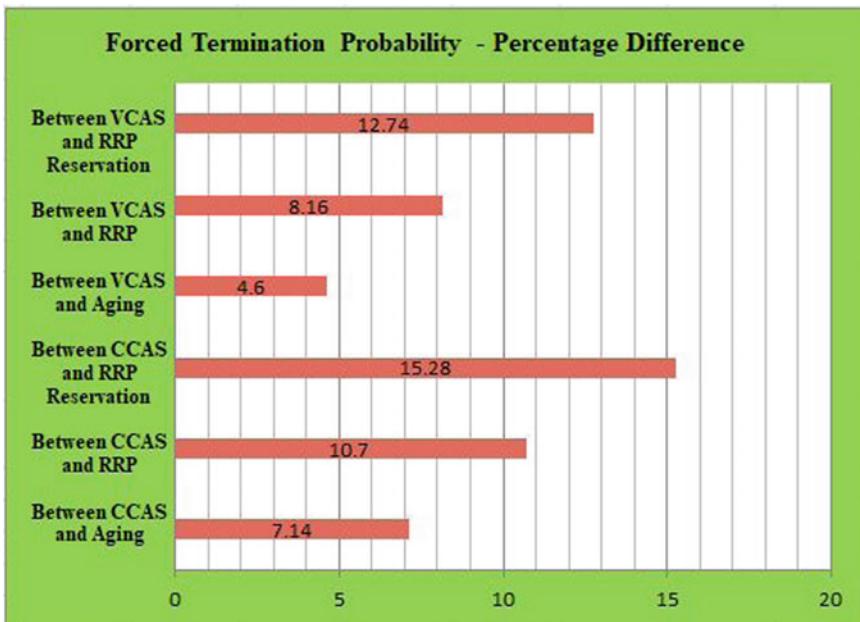


Fig. 10 Forced termination probability-percentage difference

RRP scheme shows a 10.7% reduction in FTP than CCAS and an 8.16% reduction compared with VCAS. The RRP with reservation scheme shows a 15.28% reduction in FTP than CCAS and a 12.74% reduction compared with VCAS.

8 Conclusion

Many research works are carried out in spectrum management functions. One such research area is spectrum sharing and allocation. Even though many research works are carried out in spectrum allocation, the QoS given to secondary users in CRN is not yet up to the mark. The research work provides fair spectrum allocation algorithms that improve the QoS given to SUs in CRN. Three different spectrum allocation algorithms are introduced, namely the aging algorithm, RRP algorithm, and RRP with reservation algorithm. All these allocation algorithms aim to allocate spectrum among users in CRN fairly and improve QoS given to SUs. Performance metrics like blocking probability and forced termination probability are used to measure the QoS of SUs. For example, the SU dropping rate is measured in blocking probability, RRP with reservation scheme shows 19.32% reduction in blocking probability compared with the traditional VCAS.

Due to preemption, SUs will be interrupted by high-priority users and forced to terminate from the system. Also, due to channel failures, PUs and SUs transmission can be interrupted. With the set of reserved channels in the network, the interruption rate decreases, and the system provides continuous, uninterrupted service to the users. The forced termination probability measures the interruption rate. RRP with Reservation scheme shows a 12.74% reduction in forced termination probability when compared with the traditional VCAS.

References

1. <http://www.itu.int/ict/statistics>
2. FCC (2003) ET Docket No 03–222 Notice of proposed rulemaking and order
3. Patil K, Prasad R, Skouby K (2011) “A survey of worldwide spectrum occupancy measurement campaigns for cognitive radio”, In IEEE international conference on devices and communications, ICDeCom, pp 1–5
4. Jayavalan S, Mohamad H, Aripin NM, Ismail A, Ramli N, Yaacob A, Ng MA (2014) “Measurements and analysis of spectrum occupancy in the cellular and tv bands”. Lecture notes on software engineering 2(2)
5. Chen D, Yin S, Zhang Q, Lui M, Li S (2009) “Minimum spectrum usage data: a large scale spectrum measurement study”, ACM
6. Zulfiqar MD, Ismai K, Hassan NU, Hussain S, Zhang M (2019) “Radio spectrum occupancy measurement from 30MHz-1030MHz in Pakistan”, UK/China Emerging Technologies (UCET)
7. Borde S, Joshi K, Patil R (2019) “Quantitative analysis of radio frequency spectrum occupancy for cognitive radio network deployment”, Lecture Notes Netw Syst
8. Hassan MR, Karmakar GC, Kamruzzaman J, Srinivasan B (2017) “Exclusive use spectrum access trading models in cognitive radio networks: a survey”, 4th Quart, IEEE Commun Surveys Tuts 19(4):2192–2231
9. Akyildiz IF, Altunbasak Y, Fekri F, Sivakumar R (2004) Adaptnet: adaptive protocol suite for next generation wireless internet. IEEE Commun Mag 42(3):128–138
10. Andrews JG, Buzzi S, Choi W, Hanly SV, Lozano A, Soong ACK, Zhang JC (2014) What will 5G be? IEEE J Sel Areas Commun 32(6):1065–1082
11. Ahmad et al (2020) 5G technology: towards dynamic spectrum sharing using cognitive radio networks. IEEE Access 8:14460–14488. <https://doi.org/10.1109/ACCESS.2020.2966271>

12. Akyildiz IF, Lee WY, Vuran MC, Mohanty S (2006) NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Comput Netw* 50(13):2127–2159
13. Akyildiz IF, Lee WY, Chowdhury KR (2009) CRAHNs: cognitive radio ad hoc networks. *Ad Hoc Netw* 7(5):819–836
14. Jiao L, Li FY, Pla V (2012) Modeling and performance analysis of channel assembling in multichannel cognitive radio networks with spectrum adaptation. *IEEE Trans Vehicular Technol* 61(6):2686–2697
15. Lee J, So J (2010) “Analysis of cognitive radio networks with channel aggregation”. *IEEE Wireless Commun Netw Conf*
16. Rahim M et al (2020) “Efficient channel allocation using matching theory for qos provisioning in cognitive radio networks”. *MDPI J Sens* 20:1872. <https://doi.org/10.3390/s20071872>
17. Zhai C, Tian J (2020) “Underlay spectrum sharing with adaptive interference cancellation at primary and secondary receivers”. *Telecommun Syst* 73:595–605. <https://doi.org/10.1007/s11235-019-00650-z>
18. Chakraborty T, Misra IS (2020) A novel three-phase target channel allocation scheme for multi-user cognitive radio networks. *Comput Commun* 154:18–39
19. Balapuwaduge IAM, Jiao L, Pla V, Li FY (2014) Channel assembling with priority-based queues in cognitive radio networks: strategies and performance evaluation. *IEEE Trans Wirel Commun* 13(2):630–645
20. Jiao L, Balapuwaduge IAM, Pla V, Li FY (2014) On the performance of channel assembling and fragmentation in cognitive radio networks. *IEEE Trans Wireless Commun* 13(10):5661–5675
21. Ren P, Wang Y, Du Q (2014) CAD-MAC: A channel aggregation diversity based MAC protocol for spectrum and energy-efficient cognitive ad hoc networks. *IEEE J Sel Areas Commun* 32(2):237–250
22. Suganthi N, Meenakshi S (2017) “Channel aggregation with queuing model and starvation mitigation in cognitive radio network”. *J Adv Res Dyn Control Syst* 9(Sp—17):690–706
23. Suganthi N, Meenakshi S (2018) “An efficient scheduling algorithm using queuing system to minimize starvation of non-real-time secondary users in cognitive radio networks”. *Cluster Comput J Netw Softw Tools Appl* Mar 2018. <https://doi.org/10.1007/s10586-017-1595-8>

Toxic Comment Classification Using Bi-directional GRUs and CNN



Ritambhra Vatsya, Shreyasi Ghose, Nishi Singh, and Anchal Garg

Abstract In this era, where Internet communities have a very prominent role in our lives, a notable positive change in our mood can be transpired by just a single post or comment. Just like online communities have its advantages, it also has its downside of making everyone vulnerable to the menaces of maltreatment and harassment online. As a solution to this problem, we have used the corpus provided by Conversation AI for toxic comment classification which contains labeled comments. We have implemented a multi-headed classification model using bi-directional-GRUs and convolution neural networks so that sequential and high-dimensional data are handled efficiently. The GRU being used is bi-directional, so that the model understands the context of data in both forward and reverse direction. This model will detect different types of toxicity like insults, obscenity, threats, etc., and helps in removing those comments which are detected to have any of these characteristics.

Keywords Text classification · Bi-GRU · CNN · Natural language processing

1 Introduction

Social media has a very profound effect on society in both positive and negative ways. While it lets people share engaging and informative content, concurrently it is also making everyone vulnerable to the threat of abuse and harassment online. Health professionals are extremely worried about the impact that social media, and Internet communities have on the mental health of people. Many individuals suffer from harassment online which leads to increased levels of stress and sometimes even affects their psychological health [1]. A survey suggests that the rapid transformation in Internet accessibility has led to escalated growth in Internet communities, especially among teenagers. This aspect of their lives has now become a substantial

R. Vatsya (✉) · S. Ghose · N. Singh · A. Garg
ASET, Amity University, Noida, Uttar Pradesh, India
e-mail: agarg@amity.edu

part of where their value comes from and what has been shaping their identities and status for the past decade [2].

One article on gender-specific hate revealed that women are in general more susceptible to sexual harassment online due to the gender hierarchy in our society and the need of the male gender to keep their high social status given by society by pulling the social status of women down, which in this case is through derogatory and abusive remarks on social media [3]. An article titled “Spiral Of Silence” states that a person’s opinion on a certain social or political situation can lead to them facing hate from the majority whose opinions differ from the individual. This same situation manifests itself in social media where people are harassed because of their opinion which leads to them not voicing their opinions in future and hence affects free speech [4].

Toxic comment classification is extremely useful in monitoring Internet communities and aids in preventing hateful comments from reaching users. Initially, companies identified hate comments manually, which were then reported for, and removed. But in reality, this approach failed since hate speech has many dimensions, and each company has different regulations for this classification. Also, the process of going through every report becomes a colossal problem, given a large number of reported comments [5].

This paper aims to automate the task of classifying the comments with high accuracy, so that toxic comments can be detected, removed, and not cause any harm. This paper uses Jigsaw’s Toxic Comments Dataset. Denoising of the data zooms in on the features required by the model, and then, data cleaning is performed on the data to further clean the textual data to ease the process of model development. This deep analysis allows us to better understand the data and hence will result in a better model. After getting a deep understanding of the data, preprocessing is performed to convert textual features into feature sequences to be fed into the model. It also filters out the irrelevant data as described in the methodology section.

The transformed data is then fed to the model which has been built using layers of bi-directional GRUs, CNN, and a dense layer. Combination of these units result in better contextual understanding by the model. Also, using GRU instead of LSTM decreases the computation power being used. Using this model, the comments are classified as toxic, obscenity, threat, insults, and identity hate with better accuracy.

The structure of our paper is as follows: Sect. 1 introduces the need for toxic comment classification, Sect. 2 will explore the related work on this topic, Sect. 3 will explain our proposed method which will include a description of the data used, steps of data preprocessing and the model. Further, Sect. 4 will show our experimental results, and Sect. 5 will conclude this paper.

2 Related Work

Text classification has drawn lots of attention in recent years, and methods like term frequency-inverse document frequency (TF-IDF) and support vector machine (SVM) are being widely used [6]. Deep neural network models were applied to improve performance and have exhibited the benefit of integrating preprocessing and the CNN-GRU network. Even though CNN originated in the computer vision field, recently, it is being employed in text analytics as well since CNN can learn efficiently the hierarchical structure of a language and effectively handle variable lengths. Zhang and Robinson have experimentally illustrated that text classification using a CNN structure yielded better performance than regular feed-forward networks and SVM [7, 8].

Recurrent neural networks specialize in processing sequential data. Seeing as how the text is inherently a type of sequential data, RNNs are often used for text classification as demonstrated by Liu and Huang in their paper [9]. Sometimes, exploding or vanishing gradient problems occur and cause an issue in learning long-term dependency in textual data. This problem can be resolved by using a long short-term memory (LSTM) or gated recurrent units (GRU) since they contain gated units which act as memory cells [10]. Bi-LSTM demonstrated by Zhou and Zheng also gives slightly better results as compared with vanilla-RNN [11]. GRU is a more computationally efficient case of LSTM. It controls the flow of information through gates without using separate memory cells, and fully exposing the content state unlike LSTM [10]. Many studies have performed tasks related to sentiment analysis using either LSTM or GRU and have demonstrated favourable performances and outcome [12–14]. Combination models for classification such as Bi-RNN, bi-directional-GRU, bi-directional-LSTM along with a pretrained word embedding model developed by Facebook Artificial Intelligence Research (FAIR) were also taken into consideration [11, 15, 16].

3 Methodology

3.1 Dataset

Data being used is downloaded from Toxic Comments Classification Challenge by Jigsaw. This dataset contains comments and its respective one-hot encoding of comment tags: toxic, severe toxic, obscene, threat, insult, and identity hate. An instance of dataset is shown in Fig. 1. Some of the comments are multitagged that can belong to more than one category as shown in one of the bar charts in Fig. 2.

The distribution of dataset for training, testing, and validations are 111,699, 23,936, and 23,936, respectively, from a total 159,571 comments.

	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
"\nNo he did not, read it again (I would have ...		0	0	0	0	0	0
\n Auto guides and the motoring press are not...		0	0	0	0	0	0
"\nplease identify what part of BLP applies be...		0	0	0	0	0	0
Catalan independentism is the social movement ...		0	0	0	0	0	0
The numbers in parentheses are the additional ...		0	0	0	0	0	0
"::::And for the second time of asking, when ...		0	0	0	0	0	0

Fig. 1 Instance of dataset used which is one hot encoded

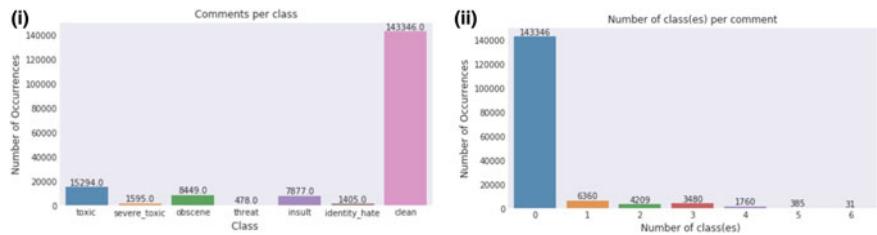


Fig. 2 Bar charts displaying (i) distribution of dataset, (ii) multiple tags in comments

3.2 Data Preprocessing

Before feeding the textual data into our model, it needs to be preprocessed using many methods so that the model is getting only the information that needs to affect classification and not anything else. The tasks that were done in this experiment to preprocess the data are as follows:

1. Dropping null values: Keeping in mind the huge dataset, there were bound to be null values, so those values were dropped as without the textual data, the particular row would have been of no use.
2. Converted all the text to lowercase to obtain uniformity in text.
3. Removal of newline characters: since these comments have been scraped, they have special new line characters, which were removed using regular expressions.
4. Removal of metadata: After analysis of data, it was found that many comments had some metadata like IP addresses and usernames which were irrelevant and hence were stripped down from the text.
5. Expanding abbreviations: Many English words like don't, can't are used as abbreviations of terms, in this case, do not and can not, so to treat them as the same we have replaced these common abbreviations with their respective expanded form.
6. Lemmatization: Further, we have used lemmatization so as to only have the base word and not many forms of the same word [17].

7. Elimination of punctuations: Finally, the text has been stripped of any meaningless punctuations. No stop words were eliminated as they did not show any significant impact on the sequence model [18].
8. Tokenization: The corpus obtained through the above steps of cleaning is then passed through tokenizer, and in this experiment, we have taken the top 100,000 words by frequency. This step converts our text into a sequence of numbers and each index would represent a word that can be accessed through the tokenizer object.
9. Padding the sequence text: The output from converting text to sequences in the above step would have a variable length of sequence for each input but our model would need a constant dimension of sequences which is why we would pad our sequences. In this experiment, we have used the maximum length of the sequence to be 150 [19].

After applying the above steps to our training, validation, and testing dataset, the data is ready to be fed into the model.

3.3 *Model*

We finally developed a deep learning model for the toxic comment classification on the above-described dataset. The architecture of our model contains an embedding layer, a 1D spatial dropout layer, a bi-directional GRU layer, a 1D convolution layer, concatenated average and max-pooling layers, and finally a dense layer. All of these layers have been used from Keras. Each of these layers is described in the sub-sections below:

1. Word Embedding Layer: To obtain embedding vectors of words, pretrained embeddings are used in deep learning models. We have used one of these pre-trained models which is already trained on huge datasets with much more information for our model. In this experiment, we have used the 300-dimensional version of GloVe embedding vectors, and hence, the words in the vocabulary have been mapped to their respective embedding vectors.
2. Spatial Dropout: Next, we have used a one-dimensional spatial dropout layer as it works to drop the entire feature along all of the channels instead of dropping features independently and helps in reducing the effect of overfitting and gives more generalized feature maps. The rate of dropout that we have taken in this model is 0.2.
3. Bi-directional GRU: For sequence modeling, we have used a bi-directional GRU. RNN was not used due to long sequences of comments which would cause the issue of vanishing gradient in this case and we would lose much information. LSTM could also have been used here, but GRUs are much more computationally efficient in this case given the large dataset. In GRU, we have used the bi-directional one, which comprises two GRU units, used for taking in input in both forward as well as backward direction.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 150)	0	
embedding_2 (Embedding)	(None, 150, 300)	30000000	input_2[0][0]
spatial_dropout1d_2 (SpatialDro)	(None, 150, 300)	0	embedding_2[0][0]
bidirectional_2 (Bidirectional)	(None, 150, 256)	329472	spatial_dropout1d_2[0][0]
conv1d_2 (Conv1D)	(None, 149, 64)	32832	bidirectional_2[0][0]
global_average_pooling1d_2 (Glo)	(None, 64)	0	conv1d_2[0][0]
global_max_pooling1d_2 (GlobalM)	(None, 64)	0	conv1d_2[0][0]
concatenate_2 (Concatenate)	(None, 128)	0	global_average_pooling1d_2[0][0] global_max_pooling1d_2[0][0]
dense_2 (Dense)	(None, 6)	774	concatenate_2[0][0]
<hr/>			
Total params: 30,363,878			
Trainable params: 363,878			
Non-trainable params: 30,000,000			

Fig. 3 Summary of the model

4. Convolution and Pooling Layers: Followed by the BiGRU layer, we have used a one-dimensional convolution layer as well as concatenated average and max-pooling layers. The one-dimensional convolution layer takes in the sequence of input, churns out features from it, and maps those internal features of input sequences. The GRU units would help the model in capturing the contextual semantic information of data, and then, CNN would capture the similarities between this captured information.
5. Finally, a dense layer is used with six units since our output needs to have probabilities of the six classes. The activation function used here is the sigmoid function.

The model built was fed with the training and validation data. The model was compiled with binary cross-entropy as the loss function and Adam optimizer with learning rate 0.001. The model was trained with a batch size of 128 and in 5 epochs. The summary of model layers and their respective parameters trained is given in Fig. 3.

4 Experimental Results

The metrics used on training and testing dataset are accuracy, precision, recall, and f1-score. The results on all of these datasets are given in Table 1.

For the testing dataset, the model was evaluated on different threshold values of the predicted probability ranging from 0.1 to 0.9. As per the results, 0.4 is the optimal threshold value. The metric results for all the thresholds tested are given in Table 2. The F1-score for training data came out to be 78.44%, and the same for testing data

Table 1 Result of metrics on the dataset

Metrics	Training dataset	Testing dataset
Accuracy (%)	91.88	96.90
Precision (%)	84.25	75.80
Recall (%)	74.34	75.82
F1-score (%)	78.44	76.02

Table 2 Result of metrics on different threshold values

Threshold value	Precision (%)	Recall (%)	F1-score (%)
0.1	56.1	90.36	69.22
0.2	65.43	85.29	74.05
0.3	71.27	80.85	75.76
0.4	75.80	75.82	76.02
0.5	80.05	71.84	75.72
0.6	83.97	67.07	74.57
0.7	87.48	62.07	72.62
0.8	91.11	54.78	68.42
0.9	95.02	43.99	60.13

on a threshold value of 0.4 came out to be 76.02%. The accuracy for the training and testing data are 91.88% and 96.90%, respectively.

These results are much better than the results we got by only using GRU/Bi-GRU or CNN. The combination of Bi-GRU & CNN result in faster convergence of loss against number of iterations performed by our model, thus making this hybrid model the best choice. Additionally, it also required less computation time than LSTM.

5 Conclusion and Future Scope

In this paper, we have developed a model which classifies the comments as toxic, insulting, obscene, etc. Classifying toxic comments would be extremely useful in clearing out the negative comments on social media which would further make a huge impact on the mental well-being of people. In future, we would like to work on building a model for classifying multilingual comments, so that we are not just limited to English which would represent the real-life scenario of social media comments much more accurately.

References

1. Koutamanis M, Vossen HGM, Valkenburg PM (2015) Adolescents' comments in social media: Why do adolescents receive negative feedback and who is most at risk? *Comput Hum Behav* 53:486–494
2. Boyd D (2008) Why youth (heart) social network sites: the role of networked publics in teenage social life. In: Buckingham D (ed) *Youth, identity, and digital media*. The MIT Press, Cambridge, The John D. and Catherine T. MacArthur foundation series on digital media and learning, pp 2007–16
3. Berdahl Jennifer L (2007) Harassment based on sex: protecting social status in the context of gender hierarchy. *Acad Manag Rev* 32(2):641–658
4. Noelle-Neumann E (1974) The spiral of silence a theory of public opinion. *J Commun* 24(2):43–51
5. Carlson CR, Rousselle H (2020) Report and repeat: investigating Facebook's hate speech removal process. *First Monday*
6. Dadgar SMH, Araghi MS, Farahani MM (2016) A novel text mining approach based on TF-IDF and Support Vector Machine for news classification. In: 2016 IEEE international conference on engineering and technology (ICETECH). IEEE
7. Ouyang X et al (2015) Sentiment analysis using convolutional neural network. In: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing. IEEE
8. Zhang Z, Robinson D, Tepper J (2018) Detecting hate speech on twitter using a convolution-gru based deep neural network. European semantic web conference, Springer, Cham
9. Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. arXiv preprint [arXiv:1605.05101](https://arxiv.org/abs/1605.05101)
10. Chung J et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555)
11. Zhou P, Qi Z, Zheng S, Xu J, Bao H, Xu B (2016) Text classification improved by integrating bidirectional lstm with two-dimensional max pooling
12. Greff K et al (2016) LSTM: a search space odyssey. *IEEE Trans Neural Netw Learn Syst* 28(10):2222–2232
13. Tang D, Bing Q, Ting L (2015) Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1422–1432
14. Seo S et al (2020) Comparative study of deep learning-based sentiment classification. *IEEE Access* 8:6861–6875
15. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *Trans Sig Proc* 45(11):2673–2681
16. Duan Z, Lu R (2017) Bidirectional GRU for sound event detection
17. Balakrishnan V, Lloyd-Yemoh E (2013) Stemming and lemmatization: a comparison of retrieval performances 174–179:1198–1207
18. Munková D, Michal M, Vozár M (2013) Data pre-processing evaluation for text mining: transaction/sequence model. *Procedia Comput Sci* 18:1198–1207
19. Dwarampudi M, Reddy NV (2019) Effects of padding on LSTMs and CNNs. arXiv preprint [arXiv:1903.07288](https://arxiv.org/abs/1903.07288)

VGG-16-Based Framework for Identification of Facemask Using Video Forensics



Sunpreet Kaur Nanda, Deepika Ghai, and Sagar Pande

Abstract In the context of the COVID-19 disease outbreak, organizations such as the universities are at risk of being essentially shut around the world if the overall condition does not improve. The other name for COVID-19 is a serious acute respiratory syndrome, a virus that causes serious respiratory problems. Corona virus-2 is a contagious agent spread through droplets in the air from an affected patient. This spreads easily by direct contact with affected patients or touching the objects which all already touched by the affected patients. Even if there are many vaccines available to defend against COVID-19 across the globe, still there is a high necessity to consider the precautions for avoiding infection. The major aspect for preventing the infection using a facemask that protects a person from entering the virus into the body through the nose and mouth of a person. The other major aspect for preventing the infection by washing hands using and washes or sanitizers. In the present article, the major and popular advanced technique used for image-based detection and classification is the Deep Learning-based VGG-16 technique. The deep learning technology is used in the analysis to identify face mask recognition and determine whether or not the individual is carrying a facemask. VGG-16 is the CNN (Convolutional Neural Network) framework utilized for the present study. The Kaggle dataset considered consists of 25,000 images with each of the images having 225×225 pixels as the resolution, and the proposed model performed with a 96% accuracy.

Keywords Facemask identification · Image classification · Deep learning · VGG-16 · COVID-19

S. K. Nanda (✉)

Electronics and Telecommunication Engineering, P.R. Pote Patil College of Engineering and Management, Amravati, India

S. K. Nanda · D. Ghai · S. Pande

School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, Punjab, India

e-mail: deepika.21507@lpu.co.in

1 Introduction

Across the globe, the COVID-19 or the coronavirus pandemic has infected many individuals. It suppressed the whole nation's economic prosperity across the globe. This disease is a new respiratory infection triggered by the coronavirus of the extreme acute respiratory syndrome. As of June 10, 2020, the outbreak had affected almost 8 M individuals and killed 0.5 M individuals [1]. To prevent the infection from spreading, the WHO has mandated that people wearing face masks, maintain strict social distancing in public areas, and washing or sanitizing their hands regularly. According to various researches, wearing a facemask is critical for preventing the transmission of the infection. According to researches, N95 and surgical masks are 91 and 68% efficient in the prevention of infection through virus transmission [2]. The individuals who wear these masks efficiently disrupt aerial viruses, preventing diseases from reaching an individual's respiratory system, and is a low-cost way to reduce fatal accidents and respiratory infectious illness. Nonetheless, because of insufficient facemask usage, the efficiency of these masks in prohibiting infection transmission in the public at large has been reduced. It is critical to implement an automated facemask identification application that can provide individual security while also preventing a regional outbreak [3].

The Deep learning technique is the development of combination with computer vision convergence provide a technological revolution in a variety of domains. DNNs, the key aspect of deep learning techniques, have whatever they need to identify the objects, classification of images, and segmentation of images. CNN's are a form of DNN that is commonly utilized in computer vision applications [4]. Through the immense extraction of features that can able to store image sequence information, CNNs can distinguish and recognize face mask images although with small variations after training the classifier model. The proposed research focuses on the application of deep learning techniques to create a classifier model that will gather photographs of people with face masks and those that don't from a repository and distinguish between the two classes of having-facemask and not-having-facemask [5]. The ANN is a powerful methodology for extracting features from unprocessed files. The usage of a CNN to construct the facemask classifier model and the impact of the no. of layers having convolution operation on predictive performance was proposed in this report. This research makes use of the OpenCV and TensorFlow libraries along with the Python language.

The present article is organized into various sections. Section-II deals with the literature survey of the recent articles, section-III deals with the methodological discussion of the proposed framework and then followed by the dataset description, section-IV deals with the discussion of environment and system description for the development of the proposed framework and then the discussion of results obtained from the proposed framework, section-V deals with the discussion of conclusion and future scope for the proposed framework.

2 Related Work

In 2020, Militante et al. [6] proposed a facemask recognition system with the aid of a deep learning technique. This is a real-time application developed using Rasberry Pi. This application generates an alarm when an individual is identified without wearing a face mask. The proposed model able to attain better performance in terms of accuracy as a performance metric is 96%. Face detection has become one of the most commonly utilized as a key aspect of biometric identity technologies, thanks to its accelerated growth [7]. It can be used for a variety of purposes, including defense, military, transportation, schooling, and others. Face identification is becoming increasingly common. After all, there are certain shortcomings in modern methods. As a simple and contactless means of correct identity authentication, face identification has become indispensable in our everyday lives. Individuality checking at automated border checkpoints and safe authentication to digital applications are being heavily reliant on all these techniques. The latest COVID-19 pandemic has highlighted the importance of sanitary, contactless identity authentication. The pandemic, on the other hand, resulted in the widespread usage of face-masks, which were necessary to maintain the pandemic within the regulation [8]. The impact of having a mask on face identification in a cooperative way is a significant yet underdeveloped topic right now.

To model practical usage cases, Damer et al. [9] presented a specially assembled repository comprising three sessions, each with three separate collection orders. The authors also look at how masked face queries affect the performance of three of the best-performing face identification technologies, two academic implementations, and one industrial off-the-shelf application. Face detection is a computationally difficult process that individuals master with ease. Despite these challenges, this extraordinary talent works best with recognizable faces than with strangers. Latest studies say that distinct characteristics are utilized for the portrayal of recognizable and unknown faces to accommodate for individuals' remarkable capability of recognizing the amicable faces. In this research, in 2019, Abudarham et al. [10] used a reverse engineering methodology to determine which facial characteristics are important for recognizing a recognizable face. The authors noticed that the similar subset of features that are utilized for mapping unknown faces are often utilized for mapping and identification of recognizable faces, contrary to popular belief [11]. The authors also demonstrated that a DNN face identification methodology utilized these functions. To compensate for individuals' exceptional identification of recognizable faces, the authors suggest an advanced paradigm that considered similar visual portrayal for all faces and combines awareness and interpretation.

In 2019, Zhi et al. [12] used a genetic methodology to incorporate a facial identification scheme. Lighting adjustment in video images, feature extraction, feature collection, labeling, recognition, and view are all part of the application. An individual face identification system with decent reliability and feasibility has a recognition score of over 90% [13]. It has provided a solid basis for further research into facial identification technology [14]. They suggested a facemask identification

approach based on the Gaussian Mixture framework proposed by, in 2018, Chen et al. [15] for the avoidance of various frauds. In the domain of economical protection alert system can solve the issue of recognizing the faces along with the masks. The authors also demonstrated the way to use OpenCV and dlib to recognize and remove faces. The GMM is used to build the individual face framework. the authors measured the resemblance among the face specimen and the framework based on this [16]. The authors predicted whether the image for testing is an individual face with or without a mask by studying and learning the characteristics of faces. In comparison with other conventional facial identification methods, the proposed solution aims to improve the capability to identify unusual faces such as sunglasses, goggles, and respirators, as well as decrease the possible threat of these rare faces in the secured aspect. It is easy to quantify and has a greater level of precision. Furthermore, our approach has improved the application's mask identification durability. Police officers could use accurate facial identification techniques to recover suspects' faces from security camera feeds, and cross-border passengers could pass through a face verification screening line without the intervention of officers. However, since public safety depends largely on such smart devices, the programmers should take into account the major threats aimed at deceiving such devices that use facial identification.

In 2018, Zhou et al. [17] proposed an advanced threat on face identification applications that involves highlighting the object with infrared light based on disruptive attacks produced by the proposed framework, allowing face identification applications to be circumvented or fooled whereas the infrared disturbances are invisible to the naked eye [18]. An intruder can not only avoid video surveillance when executing the same kind of threat. More specifically, if only the perpetrator's profile is obtained by the intruder, he will interrogate his intended target and bypass the face verification device. The threat is almost undetectable by surrounding citizens, not just due to the illumination is transparent, yet also due to the application used to unleash it is compact sufficiently [19]. As per the author's research on a massive dataset, hackers have a very good rate of success in discovering such a confrontational instance that can be applied using infrared, with a rate of success over 70%. The author's best knowledge to show the seriousness of the challenge posed by infrared confrontational instances of face identification. In 2018, Masi et al. [20] offered a thorough overview of recent advances in deep (Face Recognition) FR, addressing important issues such as algorithm architectures, repositories, interfaces, and implementation scenarios. In the first phase, the authors reviewed the various network structures and loss mechanisms that have been introduced in the accelerated development of deep FR approaches. In the second phase, identified two types of similar FR methods such as "one-to-many augmentation" and "many-to-one normalization." Next, for both framework training and assessment, the authors reviewed and evaluated the most widely utilized repositories. In 2017, Mahmood et al. [21] gave a rundown of the most up-to-date FR methodologies, based on their results on readily accessible datasets. The circumstances of the image repositories are highlighted in terms of the identification performance of each solution. This is helpful for clinicians to select an approach for their specific FR application as well as a short analysis outline. The article separates the FR implementations into three groups to furnish a concise

overview such as intensity, video, and 3-dimensional-based FR approaches [22]. The most widely utilized approaches and their efficiency are recorded on standard face repositories in each group, along with concise serious analysis.

In 2003, Zhao et al. [23] presented an up-to-date crucial overview of facial identification studies using still and video-based images. The authors presented this survey for two reasons: the primary reason to update the analysis of the current research and the second reason to generate some observations into the research of face identification by machines [6]. The authors categorized the current identification strategies and offer thorough explanations of specific strategies inside every division to generate a systematic sample. Related subjects such as psychophysical research, device assessment, and lighting and posture variance are also discussed [24].

From the review, it is concluded that it mainly deals in two different aspects. The primary aspect dealt with the discussion of the identification of face masks. The secondary aspect dealt with the discussion of face recognition. This discussion also included various methods involved in the identification of facemasks as well as the faces. Similar methods can be adopted for the current discussing framework.

3 Proposed Methodology

The proposed methodology is framed for the identification and classification of images for the facemask of the individuals. This methodology is mainly based on CNNs (Convolutional Neural Networks). CNN identifies and categorizes images based on previously learned attributes. When acquiring and evaluating the appropriate attributes of images in a multi-layered framework and it's very successful. ANN, which incorporates mathematical models through neurons and connected networks, mimics the human nervous system functioning. The significance of the ANN is discovered via the supervised learning method. CNN refers to the advancement of ANNs, which are usually focused on frameworks of recurring patterns in domains such as pattern detection or visualization. The main feature of ANNs is that, in comparison with traditional feedforward neural networks (FFNNs), they require fewer neurons due to the layering technique utilized. The generalized CNN architecture which reflects the proposed methodology for the detection of face masks can be mentioned in Fig. 1.

The generalized CNN architecture is made up of several layers. These layers are convolutional layers with ReLU activation function, maximum pooling layers, and fully connected layers. The features are extracted with the utilization of the convolutional layer and pooling layer. The extracted features are passed into flatten layer to get into a single-dimensional array to pass into fully connected layers to identify whether the recognized face had a mask or not,

The CNN is a hierarchical deep learning framework that has made a significant contribution to a wide range of computer vision as well as image-based technologies. Face identification, object identification, visual categorization, and other areas where CNN is widely utilized. The CNN framework's features are mentioned in Fig. 2. The

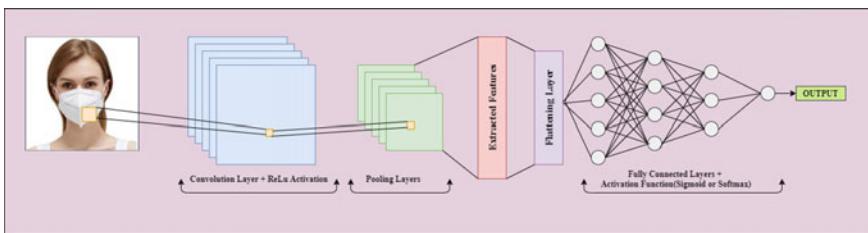
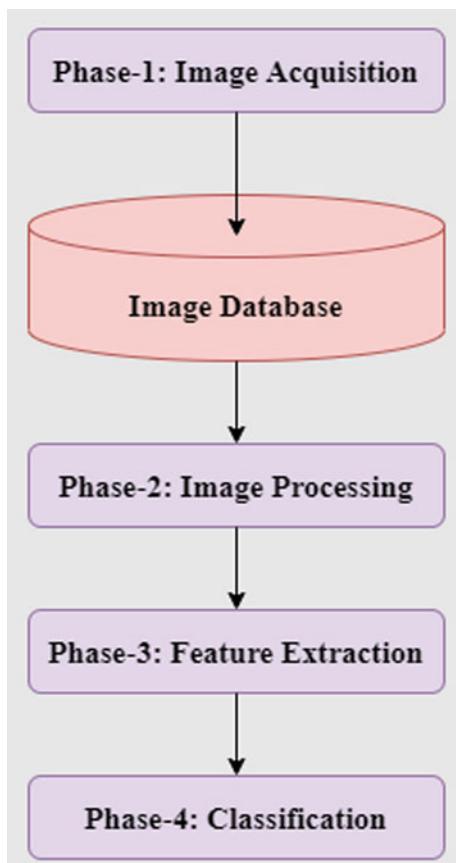


Fig. 1 Generalized CNN architecture

Fig. 2 Face mask identification framework



use of such components varies depending on the network's architecture. The proposed framework consists of various phases such as image acquisition, image processing, feature extraction using CNN-based framework, and finally, classification using fully connected layers at the end of CNN-based framework. The various phases that are involved in the proposed methodology are enlisted and discussed as follows.

3.1 Image Acquisition Phase

Image acquisition is the first phase in the real-time facemask recognition scheme. Digital cameras, cellphone cameras, and scanners are used to capture high-quality photographs of the person standing with and without a facemask.

3.2 Image Processing Phase

While processing, the collected images that will be used in a pre-processing phase are improved explicitly for the camera functionality. The internal phase of segmentation separates the images into parts, which are then used to separate the face mask-covered regions of the individual's face from the context.

3.3 Feature Extraction Phase

This phase contains the convolutional layers that receive image attributes from the resize maps, as well as the ReLU, which is connected after each of the convolutions. The scale of the function extraction is reduced by pooling the maximum and minimum values. To produce those image features, both the convolutional and pooling layers serve as filters.

3.4 Classification Phase

The last phase is to identify and classify the input images and training the CNN-based framework on the way to identify and categorize images based on observed patterns in feature extraction using the labeled images.

The CNN framework was utilized for the proposed research statement for identifying the facemask. The CNN framework considered for this research problem is VGG-16. It was proposed by K. Simonyan and A. Zisserman from the University of Oxford. This framework is used for classification based on images. The VGG-16 showed an incredible improvement over the Alexnet framework. VGG-16 was one of the most successful architectures in the 2014 ILSVRC competition. With a top-5 classification error of 7.32%, it came in second place in the classification challenge, very next to the GoogleNet. It also took first place in the localization task, with an error of localization about of 25.32%. Considering the usage and applicability of VGG-16, the same framework was utilized as a part of the proposed framework.

Table 1 Dataset details

Dataset	No. of instances
Complete dataset	25,000
Training dataset (75%)	18,750
Testing dataset (25%)	6250
Facemask having instances	15,000
No facemask having instances	10,000

4 Experimental Results and Discussions

The experimental results and discussion is conversed in this section.

4.1 Dataset Description

The Kaggle dataset consists of various images that consist of faces having masks along with faces not having masks. The size of the dataset is about 25,000 images collected from various sources. This dataset consists of 15,000 images having faces with masks and the remaining 10,000 images are of faces not having the masks. The complete dataset having two classes such as having-facemask and not having-facemask and the details of the dataset are mentioned in the following Table 1.

4.2 System and Environmental Specifications

The proposed framework was developed using open-source python language with the aid of prominent libraries such as TensorFlow and OpenCV. The proposed framework is based on the VGG-16 CNN framework. The proposed framework was utilized for supervised learning by dividing the dataset into training and testing datasets with the proportion of 75 and 25% respectively. The proposed framework is implemented on a gaming laptop and its characteristics are mentioned in the following Table 2 and

Table 2 Hardware specifications

Hardware	Specifications
Processor	Intel Core i7- 8750
Processor speed	2.20 GHz
Graphics card	NVIDIA Geforce GTX 1054
Graphic card capacity	4 GB
RAM capacity	6 GB
SSD capacity	256 GB

Table 3 Confusion matrix details

Predicted	Actual instances		
		With facemask	Without facemask
With Facemask	3,617	109	
Without facemask	131	2,143	

the details of the dataset. The proposed framework is developed using the python programming language with the aid of popular libraries such as TensorFlow and OpenCV.

The proposed framework is utilized based on the VGG-16. The number 16 indicates the total number of convolutional layers and fully connected layers. The training dataset consists of 18,750 images (about 75% of the whole dataset) and the testing dataset consists of 6, 250 images (about 25% of the whole dataset). The training and testing of the proposed framework learning the related aspects through supervised learning. The proposed framework utilized 100 epochs, a learning rate of 0.0001, the loss function utilized is Adam optimizer, and a dropout ratio of 0.3.

4.3 Results & Discussion

The performance of the proposed framework was identified using performance metrics such as accuracy, precision, recall, and F1-score. The obtained confusion matrix is represented as mentioned in the following Table 3. The obtained results for the testing images having as well as having no facemasks are represented in Fig. 3 and 4. The performance metrics obtained from the proposed framework can be mentioned in the following Table 4. The obtained performance metrics compared using a bar plot as mentioned in the following Fig. 5. While testing and training the proposed framework using the considered dataset, training accuracy, testing accuracy, training loss, and testing loss are obtained and these metrics are represented in graphical visualization as mentioned in Figs. 6 and 7.

5 Conclusion and Future Scope

This article described research on real-time facemask detection for an alert application using CNNS as part of deep learning methods. Facemask identification is made more accurate and faster using this method. The test findings indicate a high level of precision in identifying people who are having or are not having a facemask. Utilizing the VGG-16 CNN algorithm, the qualified framework was able to complete



Fig. 3 Results obtained for testing images having no facemask



Fig. 4 Results obtained for testing images having facemasks

Table 4 Performance metrics details

Performance metrics	Values in percentages
Accuracy	96.04
Precision	96.51
Recall	97.08
F1-score	96.79

its task with an output accuracy of 96%. Furthermore, by identifying whether or not a human is wearing a facemask, the research provides a valuable method in combating the transmission of the COVID-19 infection. The future scope may involve

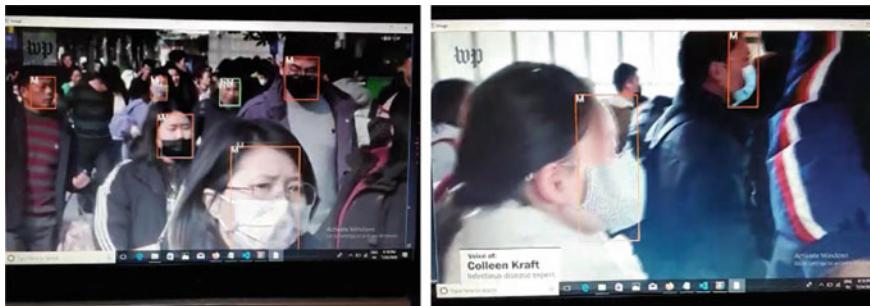


Fig. 5 Results obtained for testing images having facemasks with multiple people in a single image

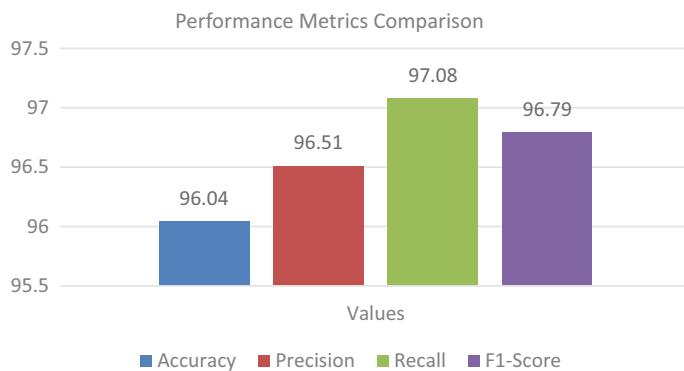


Fig. 6 Performance metrics comparison

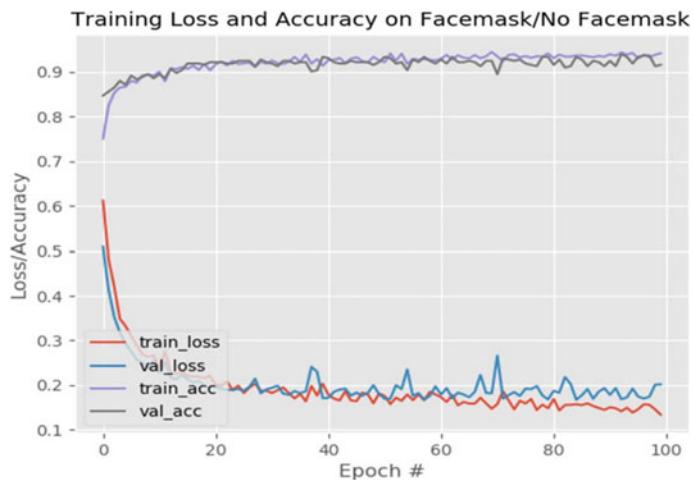


Fig. 7 The performance metrics (accuracy, loss) obtained while training and testing the proposed framework

the inclusion of social distancing, in which the camera senses whether or not an individual is having a facemask while still measuring the difference among each of the persons and sounding an alert if the social distancing is not followed correctly. It is recommended that multiple CNN frameworks be combined and that each framework be compared with the best performance accuracy throughout testing to improve performance in identifying and identifying individuals having facemasks. A separate optimizer, improved parameter configurations, fine-tuning, and the use of dynamic transfer learning frameworks are also recommended by the scientists.

References

1. Militante SV, Dionisio NV (2020) Real-time facemask recognition with alarm system using deep learning. 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), pp 106–110
2. Damer N, Grebe JH, Chen C, Boutros F, Kirchbuchner F, Kuijper A (2020) The effect of wearing a mask on face recognition performance: an exploratory study. 2020 International Conference of the Biometrics Special Interest Group (BIOSIG), 2020, pp 1–6
3. Abudarham N, Shkeller L, Yovel G (2019) Critical features for face recognition. *Cognition* 182:73–83
4. Zhi H, Liu S (2019) Face recognition based on genetic algorithm. *J Vis Commun Image Represent* 58:495–502
5. Chen Q, Sang L (2018) Face-mask recognition for fraud prevention using Gaussian mixture model. *J Visual Communication and Image Representation* 55:795–801
6. Zhou Z, Tang D, Wang X, Han W, Liu X, Zhang K (2018) Invisible mask: Practical attacks on face recognition with infrared. [arXiv:1803.04683](https://arxiv.org/abs/1803.04683)
7. Masi I, Wu Y, Hassner T, Natarajan P (2018) Deep face recognition: A survey. The 2018 31st SIBGRAPI conference on graphics, patterns, and images (SIBGRAPI), 2018, pp 471–478
8. Mahmood Z, Muhammad N, Bibi N, Ali T (2017) A review on state-of-the-art face recognition approaches. *Fractals* 25(2)
9. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: A literature survey. *ACM Computing Surveys (CSUR)* 35(4):399–458
10. Sun Y, Liang D, Wang X, Tang X (2015) Deepid3: Face recognition with very deep neural networks. [arXiv:1502.00873](https://arxiv.org/abs/1502.00873)
11. Kortli Y, Jridi M, Al Falou A, Atri M (2020) Face recognition systems: A survey. *Sensors* 20(2):342
12. Geng L, Zhang S, Tong J, Xiao Z (2019) Lung segmentation method with dilated convolution based on VGG-16 network. *Computer Assisted Surgery* 24:27–33
13. Srivastava S, Kumar P, Chaudhry V, Singh A (2020) Detection of ovarian cyst in ultrasound images using fine-tuned VGG-16 deep learning network. *SN Computer Science* 1(2):1–8
14. Rezaee M, Zhang Y, Mishra R, Tong F, Tong H (2018) Using the VGG-16 network for individual tree species detection with an object-based approach. 2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS), pp 1–7
15. Islam S, Khan SIA, Abedin MM, Habibullah KM, Das AK (2019) Bird species classification from an image using the VGG-16 network. *Proceedings of the 2019 7th International Conference on Computer and Communications Management*, 2019, pp 38–42
16. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: A survey. *Comput Electron Agric* 147:70–90
17. Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Dean J (2019) A guide to deep learning in healthcare. *Nat Med* 25(1):24–29

18. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Chintala S (2019) Pytorch: An imperative style, high-performance deep learning library. [arXiv:1912.01703](https://arxiv.org/abs/1912.01703)
19. Zhang Z, Cui P, Zhu W (2020) Deep learning on graphs: A survey. IEEE Transactions on Knowledge and Data Engineering
20. Ayyappa Y, Neelakanteswara P, Bekkanti A, Tondeti Y, Basha CZ (2021) Automatic face mask recognition system With FCM AND BPNN. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp 1134–1137
21. Lee SH (2020) Deep learning-based face mask recognition for access control. J Korea Academia-Industrial Cooperation Society 21(8):395–400
22. Liu S, Agaian SS (2021) COVID-19 face mask detection in a crowd using multi-model based on YOLOv3 and hand-crafted features. Multimodal Image Exploitation Learn 11734:117340M
23. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement 167:108288
24. Damer N, Grebe JH, Chen C, Boutros F, Kirchbuchner F, Kuijper A (2020) The effect of wearing a mask on face recognition performance: an exploratory study. 2020 International Conference of the Biometrics Special Interest Group (BIOSIG), 2020, p. 1–6

BlockSIoT: A Blockchain-Based Secure Data Sharing in SIoT



J. Chandra Priya, R. N. Karthika, K. Suresh Kumar, and P. Valarmathie

Abstract The advancements of the Internet of Things (IoT) lead to the deployment of smart objects on online social networks to offer a different paradigm, referred to as the Social Internet of Things (SIoT). The social networking things evolve with intelligence in autonomously establishing social links to explore the objects. Nevertheless, the streaming data originated from billions of linked gadgets pose a challenge to render elasticity and security to the users. Cloud and edge computing provides seamless services toward the elasticity of data for SIoT users. However, security in the cloud is a point of debate. Among the varied security and storage aspects, we concentrate on secure data transmission, storage mapping, and their purpose in service provision to improve information sharing. This paper concentrates on offering a secure and reliable transmission among the Things of social networks to withstand single-point failure. We introduced the blockchain module on top of the SIoT network to come up with the novel framework referred to as BlockSIoT to achieve a level of integrity and steer the intercommunication and community interest between the Things to crowd SIoT.

1 Introduction

Social Internet of Things (SIoT) is an innovative paradigm that blends the renowned networking technologies such as Internet of Things networks with social networks [1, 2]. IoT has connected diverse networking components by integrating different het-

J. Chandra Priya (✉)

Kings College of Engineering, Pudukkotai, Tamil Nadu, India

R. N. Karthika · K. Suresh Kumar · P. Valarmathie

Saveetha Engineering College, Thandalam, Tamil Nadu, India

e-mail: karthikarn@saveetha.ac.in

K. Suresh Kumar

e-mail: sureshkumar@saveetha.ac.in

P. Valarmathie

e-mail: valarmathie@saveetha.ac.in

erogeneous technologies and interactions [3]. Social networks are emerging at times as one of the most potent forms of data communication and sharing. SIoT strives to present the IoT network in a different prospect [4], and visualization to execute it socially by slightly manipulating its architecture and assigning social responsibilities to the Things. SIoT promotes object collaboration to build a network of smarter and socially liable data objects [5, 6]. We can configure the SIoT network as per the requisites on the network navigability to discover objects and services and its efficiency to have the collective intelligence emerging in social networks as a fascinating phenomenon [7]. IoT consists of a network of devices and sensors that generates a large scale of live data that needs to be processed and to be exchanged [8]. Most of the time, the data sharing involves massive safety-critical data that holds secrecy and privacy-sensitive information and targeted by cyber-attackers [9]. The existent security frameworks are highly server-centric [10] and cannot be adopted for IoT architecture to withstand single point failure. In line with the distributed environment, the group communal sharing can be correlated with behavioral objects which retain collective relevance. A reliable communication can be set off to leverage the degree of collaboration among the smart objects that are designated as ‘friends’ in a social network [11]. Blockchain technology possesses the same distributed topological structure as an IoT network [12, 13], where the related data is decentralized and managed by all the agents involved in the system. The vision of establishing decentralization in the networking environment is prominent. Blockchain technology offers a pseudonymous and distributed peer-to-peer networking storage model [14]. The nodes of the network cooperatively execute the operation towards the attainment of consensus. Blockchains are among the most widely discussed research thrust to yield blockchain-oriented cryptographic protocols [15]. Consequently, the blockchain has the potential to be applied to SIoT for its decentralization. In comparison with the conventional security schemes for SIoT architecture based on centralized administration, a blockchain-based social networking scheme would be efficient. The contributions of this research are reviewed as follows:

- Designing a novel blockchain-based trust model for SIoT, with the Ethereum framework and InterPlanetary File System (IPFS) to provide a secure data sharing mechanism.
- Leverages blockchain and the IPFS to address the present issues of social network to create enhanced opportunities in communication and the ability for users to express on secure, censorship-resistant networks.
- Deployment of smart contracts on top of the Ethereum blockchain for object collaboration and social association.
- Evaluation of the proposed model and its gas consumption on the Ethereum network.

The remainder of this research paper is arranged as follows. Section 2 reveals the related literatures that motivates this research work. Section 3 conveys the proposed approach on decentralized blockchain technology for SIoT network. Section 4 presents the system overview with requisites of our proposed framework. Section 5 states the assessment report of the proposed methodology. Finally, Sect. 7 includes the concluding remarks of the paper with a gist of accomplishments and directs to the future research.

2 Motivation

In the literature of Roopa et al., SIoT tends to be used to refer to Social IoT network that fuses the physical devices with information networks. It further put forth on the fact of how this new networking paradigm improves information sharing among the friend objects. Online social networks (OSN) lack in privacy preservation due to its centralized architecture [16], whereas in distributed online social networks (DOSN), there arises the problem of data availability and access control. Hence, the researchers Jiang et al. manipulate the blockchain technology to propose a novel DOSN structure that inherits the particular properties of OSN and DOSN. By incorporating smart contracts, blockchain has been implemented as a trusted server [17, 18] to support a central control service. Simultaneously, there are distinct storage amenities, so that the users have extensive power over their data. The research by Kowshalya et al. [19, 20] discussed the issues of launching a SIoT network that needs the integration of heterogeneous technologies and connectivity policies. The paper concentrates on nodal communications to be secure and reliable with the dynamic computation of trust between adjacent nodes by exchanging secret codes (Table 1).

Table 1 Focused attack vectors of SIoT

IoT process	Attack vectors
Data storage	Availability
	Access control
	Integrity
	Denial of service (DoS)
	Impersonation attacks
Data transmission	Channel security
	Session hijacking
	Routing attacks

3 Problem Analysis

3.1 *Problem Definition 1: Issues in Addressing the Dynamism, Scalability, and Heterogeneity*

The large-scale deployment of social networks exposes much dynamism and openness while rendering social services to the socially interconnected heterogenous IoT objects. In a long run, there are feasibilities for objects to be dynamically deployed or detached from the existing network to cope up with the requisites on network functionalities. The prevailing agent-based solutions to scale SIoT network leverages edge computing models to address dynamicity and interoperability. However, it reflects on the negative impact on scalability on network overload.

3.2 *Problem Definition 2: Service Rendering in Dynamic Network*

When the network size is increasing, the magnitude of sharing the information increases. Information sets are maintained at remote centralized servers to retain the references to the origin of the information. It increases the cost of communication and open to single point failures.

4 The Decentralized Approach

The SIoT model presents the social networking services' societal interests, which involves the productive profiling method that handles contextual data on several social perspectives, such as connections, trusts, and interests. Despite this, the perception of sociality in SIoT is extended beyond man to machines that incorporate social objects capable of computing and networking required to be intelligent components. As Fig. 1 depicts, the primary societal feature defines social behaviors in the SIoT structure and the complex blockchain environment. Personality is the leading character of social entities that empower the objects to project their social role in the initialization of relationships, interaction, information exchanges, requests grant, or access revocation to the resources based on preferences. Social network entities are allowed to socialize with each other to provide or receive services to reach out to their expectations. In this regard, data objects can collaborate with objects to infer or instantly request the other objects. The locality trait enables social entities to be linked within a local network of connections and interactions. Finally, trust enables social things to produce and consume data to react, generate, and maintain social connectivity based on the reliability. When a SIoT node tends to publish its data on the blockchain-

based communication platform, the data gets uploaded onto the IPFS private network of the node initially, and the obtained content address is directed to the Ethereum blockchain. From the blockchain, the intended recipients are known as ‘friends’ in the group ‘listens’ via decentralized applications and are notified that a peer in their group has published a new data. When two friends access the content, the data are copied locally in their own IPFS private network. If a third friend accesses the same data, then the data would be served from two nodes instead of one, and so on, to make the routing easier. The nodes have control over the information they come upon and can determine the sources of that information and the importance of that information based on the value others are assigning to it. The nodes are entirely in command of the type of content they want to subscribe to. The data shared on the platform are logged as transactions on the Ethereum blockchain, creating a permanent communication record. In this paradigm, there is no intermediary as blockchain removed the layer associated with censorship. For instance, consider a SIoT network for monitoring heart functioning. If a node chooses to publish using the tag, “heart rate” all users listening to the heart rate tag receives the information in their stream. The nodes only can decide whether their content is unlisted on the network. Rather than censorship of content, this framework allows its users to moderate content, designating it as high quality, or other. Since the data is located on the IPFS by a hash uniquely generated from the data file itself, if any node tried to upload the same content to the framework, the network would direct that node to original data and its license. Perhaps blockchain’s immutable architecture would serve as a disincentive for nodes to publish other copyrighted content. Blockchain can be used as a storage entity that holds the Reference Integrity Matrix (RIM) of information set. With immutability feature of blockchain, RIM can be accessed by all IoT objects with the guarantees on data integrity of RIM. At any instance, the obligations of information set can be traced and verified for its integrity from its origin by comparing it with RIMs maintained on blockchain (Fig. 2).

5 System Overview with Requisites

5.1 System Components

5.1.1 Smart Contracts

BlockSIoT and Trans are the smart contracts deployed and executed on top of the Ethereum blockchain. BlockSIoT is responsible for the creation of BlockSIoT token (Block Social Internet of Things token). We customize this token exclusively for our research on the evaluation of characteristics of the Ethereum blockchain in the context of SIoT. We ignored the economic effect of this custom token, as it goes beyond the

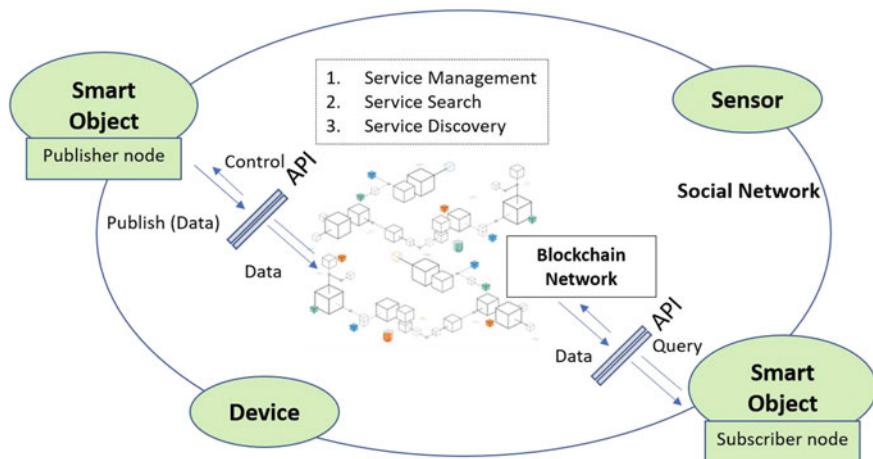


Fig. 1 Proposed systemic design outline

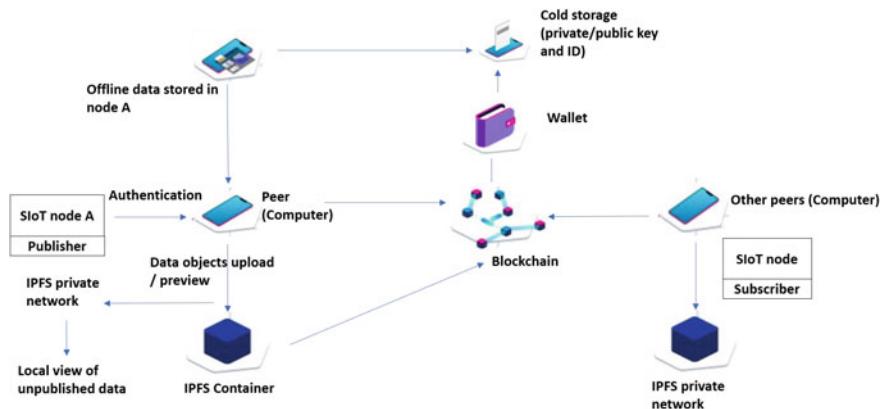


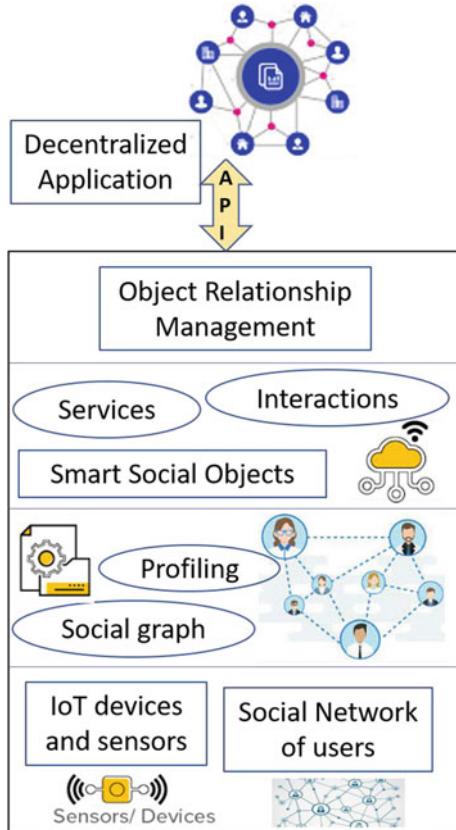
Fig. 2 BlockSIoT system insights

scope of this research work. Trans smart contract is programmed to record and trace all registered devices and sensors, initiating and logging the transactions concerning SIoT data.

5.1.2 Decentralized Application (DApp)

It aids an interface concerning the SIoT users and the blockchain that facilitates user interaction with the smart contracts and data access. It aids in visualizing the (Fig. 3) transaction logs.

Fig. 3 BlockSIoT layered architecture



5.2 User Classes

5.2.1 Publisher Smart Contracts

The smart contracts are executed by the publisher node that holds the private key of the wallet. It is authorized to adjust the gas price of BlockSIoT token and to regulate the ether balance regarding the BlockSIoT contract.

5.2.2 SIoT Sensor Traders

It is functioning to register new sensors and devices in the system. It approves the transfer of BlockSIoT tokens for ethers and vice versa. It also aids in trading BlockSIoT tokens in exchange for vending the data.

5.2.3 Subscribers

It is the set of nodes that receives the data produced by the registered SIoT sensors/devices. It allows the trading of BlockSIoT tokens towards ether and vice versa. The subscribers are recognized to buy SIoT data to hold access to it.

5.3 *Usecase Scenario*

The SIoT sensor traders initiate the decentralized application. The user registers a new SIoT sensor or device by invoking the Trans smart contract on a transaction, including the relevant information such as device/sensor type and its locality signed by the private key from the wallet. The local storage of the user maintains a RESTful API that provides the measurements of each registered sensor/device on request (Fig. 4). The sensor details are cached within the blockchain, and its generated data are accessible to the subscribers. The use case involves the subsequent steps

- i. The subscriber initiates the decentralized application
- ii. User searches for the open sensors/devices and services
- iii. The user discovers a sensor/device whose data stays impressive
- iv. The user picks the sensor/device
- v. The user determines the period for which the sensor measurements are required
- vi. The user initiates a transaction towards the blockchain to purchase data
- vii. if the user's wallet has enough BlockSIoT Tokens, the transaction is committed, tokens are paid to the publisher's account, and the transaction can be viewed. However, if the subscriber's BlockSIoT Token balance is insufficient, then the transaction aborts. Any user can buy BlockSIoT tokens at any point in time by providing ether.

The value of the token in ether can be configured and decided by the contract holder. We set the exchange rate to be one and BlockSIoT token to be one ether. The users can trade BlockSIoT tokens in return to the contract by a value not higher than the token cost. BlockSIoT smart contract accept ethers when the buyer purchased BlockSIoT tokens. The proprietor of the smart contracts can receive the ethers from

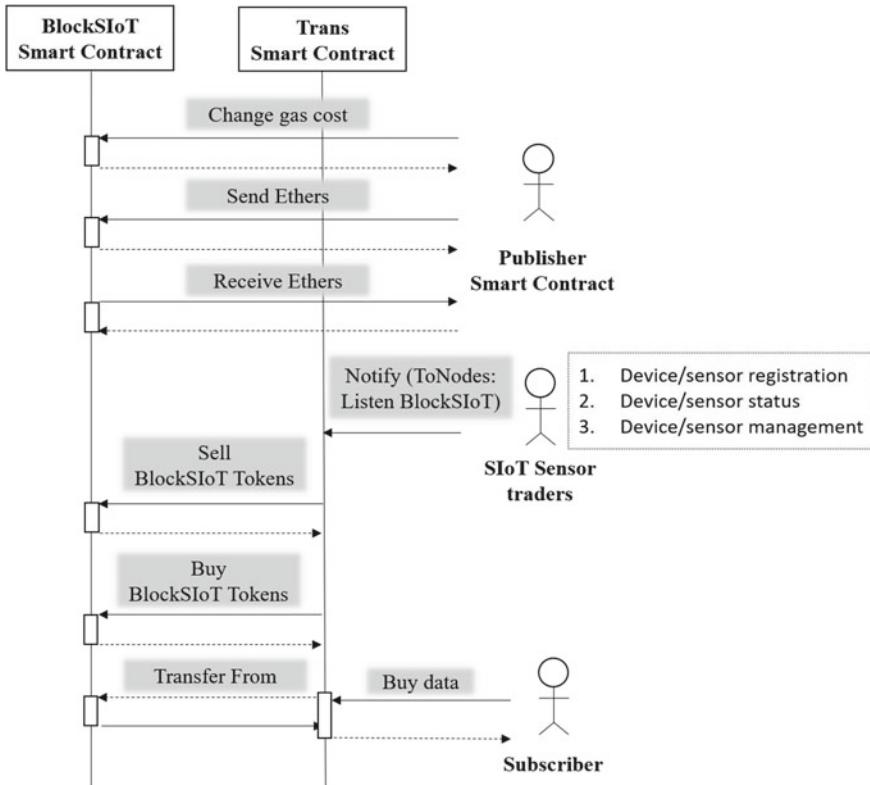


Fig. 4 SIoT: smart contract deployment

the balance of BlockSIoT contract or grant more ethers to the contract. The user can also modify the cost of the token.

Algorithm 1: SIoT: addCommunalNode

```

Input :  $f_1$ 
Output:  $b_o$ 

1 begin
2   if  $m_s \neq O$  then
3     | throw;
4   end
5   if  $f_1 \exists\exists$  then
6     | return false;
7     |  $c_1 [f_1] \leftarrow$  true;
8   return true
9   end
10 end
```

Algorithm 2: SIoT: removeCommunalNode

```

Input :  $f_1$ 
Output:  $b_o$ 

1 begin
2   | if  $m_s \neq O$  then
3   |   | throw;
4   | end
5   | if  $f_2 \neq$  then
6   |   | return false;
7   |   |  $c_1 [f_1] D$  true;
8   |   | return true
9   | end
10 end

```

6 Performance Evaluation

6.1 Experimental Setup

The testbed is implemented and the experiment is performed on I7 desktop computers which act as master and two clusters which are running Ubuntu 18.04 and equipped with Intel Core TM I7 processor with 2.40 GHz and 8 GB RAM. As for blockchain, selected the fabric as the underlying technology of the blockchain-based settlement system, which is an open-source permission blockchain technique hosted by the Linux Foundation. The implementation of Blockchain interfaces with Ganache CLI is used, where 10 testbed accounts are given with 10 private keys, every account of the testbed contains 100 ethers each for testing Smart contract through Remix and MetaMask IDE, where Web3.js is a script based on javascript for facilitating easier communication with smart contracts from Web applications. Truffle takes this a step further and enables the Web3.js interface from in the Truffle console. Metamask is an IDE used for setting up an environment for smart contract, and this helps to provide the ethers which are used to connect user machine with the Ethereum network. Ethers are also used as a gas fee for the transaction. Miners are rewarded for mining the block where the test network is not payable in which the browser works with a centralized network. Rinkeby is used to gain the ether in which blockchain is used to know the transaction address and detail, token. Truffle is a tool that is used for developing a smart contract. Ganache is an Ethereum client is used as the part of truffle ecosystem. To initiate the ganache, the node package manager is introduced.

6.2 Result Analysis

We develop the proposed smart contracts using the solidity programming language and implemented upon the Rinkeby Ethereum testnet. Gas prices of the deployed contracts are portrayed in Table 3. Each transaction that is initiated for invoking a function of a smart contract needs the compensation of a payment for exchanging with the miner to execute and store the transaction and storing within the blockchain. The transaction fee is in the form of gas which is a constant rate of performing blockchain operations, and the users can buy gas from the miner by transferring Ethers.

We closely examined the performance of our model in line with its time consumption on enciphering the plain text and locating a block in the chain. Figure 5 depicts the data searching capability of BlockSIoT implemented with Blockchain IPFS in comparison with the traditional blockchain technology. When the network scales, the searching time is also increasing in both the system. However, it is worth to note the comparatively low data searching time for the IPFS coupled blockchain to avoid unnecessary overhead (Table 2). Figure 6 shows the time to encipher the data by IPFS private network of BlockSIoT in comparison with other encryption algorithms such as RSA, Elliptic Curve Cryptography (ECC), and a fully Homomorphic Encryption (HE). We choose 128 bit prime numbers for both RSA and ECC and consider $y_2 = x^3 + ax + b$ as the elliptic curve. The plot concludes that IPFS outperforms well when compared to ECC, RSA and HE in terms of ciphering speed. Moreover, the replication mechanism of blockchain where every node holds a complete ledger offers more faster data searches. The encryption algorithms consume more amount of time when the data size is very large, whereas the hashing way of enciphering maintains almost consistent time for yielding the ciphers. IPFS is optimized by Secure Hashing Algorithm to yield a overall better performance of BlockSIoT.

Fig. 5 Data searching capability of traditional and IPFS blockchain

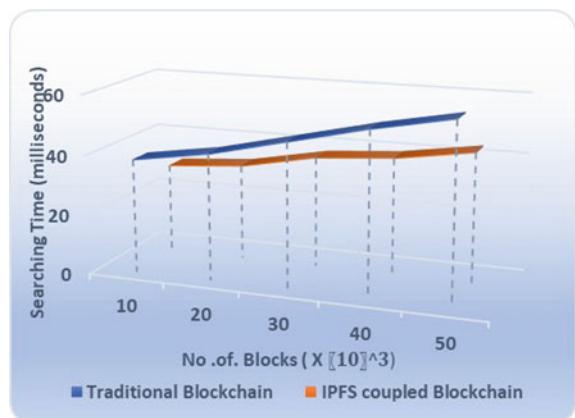
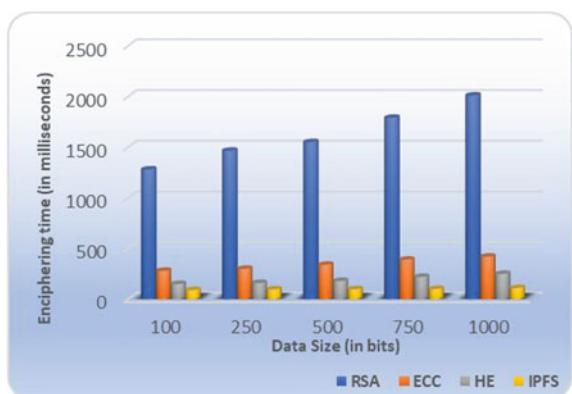


Table 2 Gas usage of contract deployment and execution

Operations	Gas costs
Transaction base fee	21,000
BlockSIoT token deployment	1,000,496
Fallback function	21,320 (for owner invocation) 30,292 (for a historic invocation)
Change gas cost	27,114–27,562
Receive ethers	30,184–30,568
Transfer	37,000–53,000
Trans token deployment	1,052,408
Notify	142,800
buyData	120,000
dataPublish()	145,625
dataUpdate()	84,336
dataDelete()	84,462

Fig. 6 BlockSIoT: enciphering time

7 Concluding Remarks

The rapid progress of the Social Internet of Things takes IoT to the next level. But it is improbable to neglect a serious question on its security issues. The networking devices and applications cannot be manufactured or designed, forecasting all the security and privacy attacks. Deploying these vulnerable devices on the network makes it unsafe for an IoT network like information integrity, privacy, and secrecy. The SIoT devices are the main target by the hackers and intruders. We focus on the demand for study on the implications of the prevailing security frameworks in SIoT. Hence, we incorporate blockchain module to achieve a peer-to-peer SIoT network.

We also understand that it is crucial to research further the distinct roles of BlockSIoT on learning methods for incorporating artificial intelligence in a secure context with a human-machine communication using the Social Internet of Things.

References

1. Zannou A, Boulaalam A, Nfaoui EH (2021) SIoT: a new strategy to improve the network lifetime with an efficient search process. Future Internet 13(1):4. <https://doi.org/10.3390/fi13010004ch>
2. Aljubaiby A, Zhang WE, Sheng QZ, Alhazmi A (2020) SIoTPredict: a framework for predicting relationships in the social internet of things. In: Dusdar S, Yu E, Salinesi C, Rieu D, Pant V (eds) Advanced information systems engineering. CAiSE 2020. Lecture Notes in Computer Science, vol 12127. Springer, Cham
3. Meena Kowshalya A, Valarmathi ML (2018) Dynamic trust management for secure communications in social internet of things (SIoT). Sådhanå 43:136
4. Loscri V, Manzoni P, Nitti M, Ruggeri G, Vegni A (2019) A social internet of vehicles sharing SIoT relationships. PERSIST-IoT workshop in conjunction with MobiHoc. Catania, Italy. <https://doi.org/10.1145/1122445.1122456.hal-02136896>
5. Roopa MS, Santosh P, Rajkumar B, Venugopal KR, Iyengard SS, Patnai LM (2019) Social internet of things (SIoT): foundations, thrustareas, systematic review and future directions. Comput Commun 139:32–57
6. Afzal B, Umair M, Shah G, Asadullah Ahmed E (2019) Enabling IoT platforms for social IoT applications: vision, feature mapping, and challenges. Future Gener Comput Syst 92:718–731
7. Atzori L, Iera A, Morabito G (2011) SIoT: giving a social structure to the internet of things. IEEE Commun Lett 15(11):1193–1195
8. Jiang L, Zhan X (2019) BCOSN: a blockchain-based decentralized online social network. IEEE Trans Comput Soc Syst 6(6):1454–1466
9. Chen Y, Xie H, Lv K, Wei S, Changzhen H (2019) DEPLEST: a blockchain-based privacy-preserving distributed database toward user behaviors in social networks. Inform Sci 501:100–117
10. Panda GK, Tripathy BK, Padhi MK (2017) Evolution of social IoT world: security issues and research challenges. Internet of Things (IoT). CRC Press, pp 77–98
11. Teixeira FA, Pereira FMQ, Wong H-C et al (2019) SIoT: securing internet of things through distributed systems analysis. Future Gener Comput Syst 92:1172–1186. <https://doi.org/10.1016/j.future.2017.08.010>
12. Jiang T, Fang H, Wang H (2018) Blockchain-based internet of vehicles: distributed network architecture and performance analysis. IEEE Internet Things J
13. Aitzhan N, Svetinovic D (2016) Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans Dependable Secure Comput
14. Jadhav B, PatilSC (2016) Wireless home monitoring using social internet of things (SIoT). In: 2016 international conference on automatic control and dynamic optimization techniques (ICACDOT). IEEE, pp 925–929
15. Atzori L, Campolo C, Da B et al (2018) Social-IoT enabled identifier/locator splitting: concept, architecture, and performance evaluation. In: 2018 IEEE international conference on communications (ICC). IEEE, pp 1–6
16. Luigi Atzori, Antonio Iera, Giacomo Morabito, Michele Nitti (2012) The social internet of things (SIoT)—when social networks meet the internet of things: concept, architecture and network characterization. Comput Netw 56(16):3594–3608

17. Priya JC, RK SBP (2018) Disseminated and decentred blockchain secured balloting: apropos to India. In: 2018 tenth international conference on advanced computing (ICoAC). IEEE, pp 323–327
18. Nitti M, Girau R, Atzori L (2014) Trustworthiness management in the social internet of things. *IEEE Trans Knowl Data Manag* 26(5), PP 1-11*IEEE Trans Parallel Distrib Syst* 2:847–859
19. Meena Kowshalya A, Valarmathi ML (2017) Trust management in the social internet of things. *Wirel Pers Commun* 96(2):2681–2691
20. Meena Kowshalya A, Valarmathi ML (2017) Trust management for reliable decision making among social objects in the social internet of things. *IET Netw* 69(4):75–80

Hybrid Feature Selection Method for Binary and Multi-class High Dimension Data



Ravi Prakash Varshney and Dilip Kumar Sharma

Abstract We have proposed a hybrid feature selection method that combines the filter-based feature selection and ensemble learning-based wrapper feature selection method. The proposed method extracts the strengths of both the feature selection methods while minimizing the overall drawbacks. We have assessed our hybrid model against the already defined model over the benchmark datasets available from the UCI Open-Source Repository. We evaluated the classification accuracy by using a Random Forest classifier. We have considered four classification metrics, namely F1-Score, Recall, Precision and Accuracy. It is evident from the study results that the proposed model excels in all the metrics on all the datasets over the existing hybrid model. For the clean dataset, our model provided **94.93%** accuracy over 90.79%. For the Libras Movement dataset, our model worked with an accuracy of **88.24%** over 87.78%. For the Ionosphere dataset, our mode's accuracy was **95.68%** over 95.28%.

Keywords Feature selection · Filter method · Wrapper method · ANOVAF value · AdaBoost · Decision tree

1 Introduction

The taste of a recipe majorly relies on the quality and relevance of the ingredients. If the ingredients are of low or bad quality, the recipe is ought to taste odd or bitter. With the advent of machine learning, many models and algorithms have been proposed and devised. But the accuracy and performance of these models majorly depend on the set of features used as an input in the models. So, for a successful machine learning or a deep learning model, the pre-requisite is a highly relevant feature subset. Not all the features in the dataset have a predictive value but are noisy, irrelevant and

R. P. Varshney (✉) · D. K. Sharma
GLA University, Mathura, India

D. K. Sharma
e-mail: dilip.sharma@gla.ac.in

redundant features. Feature selection is a key and indispensable step in the machine learning or deep learning process.

Also, these learning models do not fare well with many input features. So, the model should have a near to the optimal number of features. However, the dimensions of the dataset have increased many folds irrespective of type and class of dataset. This poses a severe challenge to the performance of the models as they need the dimension of the input dataset to be optimal. To overcome this challenge, the dimensions of the dataset need to be reduced; such a technique is known as feature selection. Feature selection is the technique of classifying the important features from the feature subset and abandoning the redundant features, thus reducing the dataset's dimension. A feature selection model comprises of a search technique for identifying the relevant features and a ranking mechanism that grades the variables as per their relevance or predictive value.

Feature selection proves as a boon to the learning process in many ways. It surges the learning speed of the model as the model takes less training time for a lesser number of features. Reduction in the feature dimension reduces the overfitting of the features and enhances the generalization capability of the model. It not only makes the model simpler but easy to interpret, understand and implement. A lesser number of relevant features reduces the risk of data errors and eradicates candidate variable redundancy. It also saves the learning process from bad learning behavior in high dimensional datasets and ultimately enhances the accuracy of the learning models. Also, the machine learning model does not perform well with dataset having a higher number of features with a low number of instances. In scenarios like this, the search space is sparsely populated, and the model fails to differentiate the relevant data and noise.

In general, feature selection strategies are classified into three categories: filter methods, wrapper methods and embedded methods. Filter methods rely on the characteristics of the feature instead of any machine learning algorithm. Hence, these methods are machine learning model agnostic and are computationally less expensive. These methods are efficient for quick screening and removal of irrelevant or redundant features. Filter methods can be categorized into two classes, univariate and multivariate. Univariate filter methods rank features independently of other features in the feature space, and hence, the methods may select redundant features. Various statistical strategies are used to rank the features like Chi-square (Fisher score), ANOVA, Mutual Information and Variance, whereas multivariate methods analyze and consider the relationship between features in the process and hence can handle redundant and correlated features.

Whereas Wrapper methods use the machine learning models to evaluate and rank the feature subset. Wrapper methods are classified as greedy algorithms because the methods evaluate all possible combinations of feature to identify the feature subset, which gives the optimal machine learning model performance. It implies that they either add or remove feature by feature-based solely on the machine learning model's performance until a specific number of features is reached, or a specified predefined criterion is met. By doing so, they ensure that the optimal subset of features for the specific machine learning model is guaranteed as output. Since the machine

learning model needs to be run on all possible feature subsets, it is computationally very expensive. Since this method is coupled with a machine learning algorithm, the optimal feature subset is mostly mapped to that machine learning model. In simpler words, one feature subset for a machine learning model may not be optimal for another machine learning model, and these methods are not machine learning model agnostic. These methods can be categorized into three: step forward feature selection methods, step backward feature selection and exhaustive search methods.

Finally, the embedded methods perform the feature selection process during the construction of the model. So, as the name suggests, these methods are embedded in the model's algorithm as its functionality. It respects the interaction between the features and model. Moreover, they are computationally less expensive than the wrapper methods as the machine learning model runs only once.

All these methods, despite making the learning process better, still suffer from some disadvantages. For example, the filter method's accuracy is less as they do not consider the classifier. Similarly, the wrapper methods are computationally expensive and overfitting. So, by combining both types of methods, we could derive benefits of both and minimize the overall drawbacks.

A hybrid method utilizes both types of feature selection methods. Firstly, a single or a combination of filter methods are used to remove the irrelevant or redundant features quickly and then followed by a wrapper method to identify the optimal feature subset. Overall, the accuracy of hybrid models is better than each of these methods if done individually.

In our study, we have taken three benchmark datasets from UCI Repository [1], analyzed the earlier feature selection hybrid model recommended by Venkatesh et al. [2] and then proposed our hybrid feature selection method and assessed the accuracy of both model against each other on the same datasets.

Further, the remaining of the paper is structured as follows: The related work is illustrated in Sect. 2. Section 3 explains the proposed work and used methodology. Section 4 hosts the results and experimental outcome. In the end, Sect. 5 details the conclusion.

2 Related Work

Study and work have been done in the area of hybrid feature selection methods. We can group these works on the type of input dataset on which the models were proposed. For example, A. Jashki et al. [3] proposed a hybrid approach on text data. Another hybrid model on text data was proposed by J. Hu et al. [4]. Y. Yang et al. [5] in their study recommended a hybrid feature system for fault bearing analysis.

Venkatesh and Anuradha [2] recommended a feature selection approach by joining a filter method and a wrapper method. They first employed Mutual Information, a filter method to reduce the dimensions of the dataset and then used Recursive Feature Elimination, which is a wrapper method to identify the optimal feature subset. Additionally, they tested their hybrid model on three benchmark datasets and used

Random Forest Classifier for comparing the classification accuracy of methods with each other.

Hsu et al. [6] designed a feature selection approach by merging the filters method and the wrapper method. Firstly, they used the filter methods, F-Score and Information Gain, to quickly screen and remove the redundant features, and the remaining features are tagged as candidate features. Then these candidate features are evaluated through the sequential floating search method (SFSM), a wrapper method and provides an optimal feature subset.

Tong Niu et al. [7], in their work, proposed a multistage feature selection method. The method worked in two stages. Firstly, they used a filter method RReliefF in stage one to quickly eliminate the irrelevant features. In the second stage, they employed the multi-objective binary gray wolf optimization algorithm collectively with a cuckoo search operator and an ELM.

Solorio-Fernández et al. [8], in their work, devised an amalgam feature selection strategy. They amalgamated the filter method and wrapper method in their study. Firstly, they sorted the features based on the Laplacian score and selected the top features above a specific threshold value which were then fed into the wrapper stage. In the wrapper method, they used the Calinski-Harabasz index for indexing the features.

Lin et al. [9] suggested a strategy where they combined filter and wrapper methods to decrease the dimensions of metabolome data. They firstly used Mutual Information with artificial variables to remove the non-informative or irrelevant and noisy variables from the metabolome data. Then, they applied Support Vector Machine -Recursive Feature Elimination (SVM-RFE) to identify the best feature subsets.

Tan et al. [10], in their study, combined GA with other strategies for feature selection. They consummated that the hybrid methods provide better results and are competent against individual methods alone. Lu H et al. [11], in their work, combined filter and wrapper method too. For the filter method, they used the mutual information maximization implementation. For the wrapper method, they used an adaptive genetic algorithm. They applied the model in gene data to reduce its dimensions.

Rouhi et al. [12] worked on reducing the dimensions of microarray data and proposed a hybrid R-m-GA method. In the hybrid model, they united a genetic algorithm, the relief filter method and minimum redundancy and maximum relevance.

Rouhi A and Nezamabadi-Pour [13], in their study, explained the most widely used feature selection methods and approaches to reduce the dimension of data and their performance in different circumstances. They covered filter-based methods, wrapper-based methods and hybrid methods in their study.

3 Proposed Work

The generic architecture of the method is explained in Fig. 1. The hybrid feature selection combines the filter method followed by a wrapper feature selection method and then a classifier to compare the accuracy. So, overall, it is a feature selection

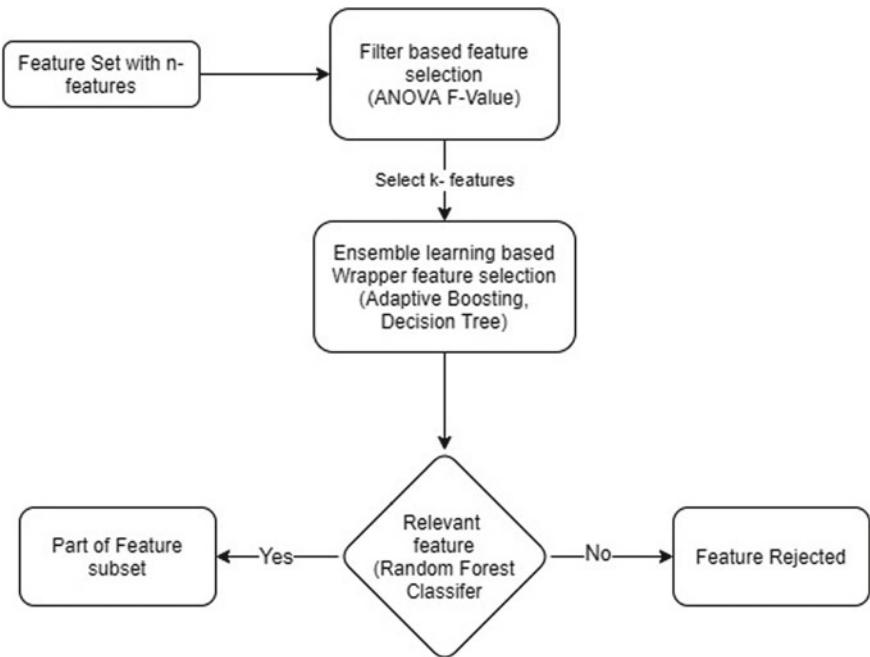


Fig. 1 Architecture for hybrid feature selection approach

model which operates in two stages. Firstly, a filter selection method extracts the most relevant features out of the entire feature set. ‘k’ derives its value from the optimal threshold value. Subsequently, the identified k-features are used as an input to the ensemble-based wrapper method, which then selects the best possible combination of feature subset.

We analyzed three datasets (Ionosphere, Clean and Libras Movement) from UCI [1] chosen for this study and then categorized the dataset into two different categories. Category 1 is a binary dataset comprising of ionosphere and clean datasets. Category 2 is a multi-class dataset having Libras Movement dataset. We have used ensemble learning methods in wrapper-based feature selection. For different types of the dataset, we have used different learning method. For the binary classifier dataset, we have used adaptive boosting (AdaBoost) classifier, and for the multi-class classifier, we have used the decision tree. Filter-based method remains the same in both types of dataset.

3.1 ANOVAF Value

ANOVA stands for analysis of variance. It is a statistical method that is used to check the means of two or more groups that are different from each other. ANOVAF

value is a statistical hypothesis test for determining whether variances of two or more groups come from the same distribution or not. An F-test or F-statistic is a simple ratio of the variance between groups and variance within groups. Variances measure the scattering and how far the data is scattered from the mean. A larger variance signifies that there is larger dispersion. The variance of a feature helps in identifying its impact on the target feature. The feature with a very low variance has no influence or impact on the target variable.

$$F = \frac{(X_1^2/(n_1 - 1))}{(X_2^2/(n_2 - 1))} \quad (1)$$

$$F - \text{Value} = \frac{\text{Variance between groups}}{\text{Variance within groups}} \quad (2)$$

3.2 Select Features Based on the Model—Adaptive Boosting

We can identify the essential or relevant features from the set by mapping a coefficient to each feature's importance in a model. Using one of the tree-based models and training the training set on, it can provide the values of coefficients for each of the features. If the assigned coefficient is large, it means that the feature is essential as it has more impact on the prediction. If the coefficient is small or zero, then it does not have any impact on the prediction and can be removed. We used adaptive boosting to train the dataset.

Adaptive boosting [14], commonly known as AdaBoost, is a boosting ensemble method. Adaptive boosting reassigns the weight to each instance and, more importantly, higher weights to the incorrect classified instances. As it learns from its mistakes, it is adaptive. Boosting helps the method to decrease the bias and variances, which occurs when models fail to notice any pertinent trends in the data. Boosting fixes the gap by analyzing the difference between the target value and the genuine value. It is based on the principle where a set of models or decision trees are trained in a sequential fashion, and the subsequent model or decision tree learns from the previous model's mistakes. Overall, except the first model, each subsequent model adapts itself from the prior model. As a part of the process, the models are sequentially added until either the prediction target is reached or an agreed-upon number of models have been added. The strength of the method resided in the principle that when we combine multiple weak classifiers together, each classifier gradually learns from objects misclassified by others, and we can build such a strong model.

Once the iterations are over, and each tree has its own final weight, its weight is multiplied with its prediction to retrieve the tree's prediction value. The prediction value of all the trees is then added to give the final prediction. Apparently, in this process, the larger the tree's weight, the larger its impact on the conclusion.

3.3 Select Features Based on the Model—Decision Trees

Decision trees are used to develop a model which learns decision rules from the training instances and predicts the target feature's value. In a decision tree, an attribute is represented by an internal node and the categories are represented by a leaf node. Choosing the root node from 'n' attributes is a complicated step and involves the use of criteria like information gain. In these criteria, values are calculated for each attribute, and then the values are sorted. The highest value attribute is selected as the root node, and this procedure is recursively followed for the remaining nodes, and the remaining nodes are selected using the same strategy.

4 Results and Discussion

In our work, we have sequenced the use of the filter feature selection method and wrapper method with an aim to boost the accuracy of the model. For implementation and verification of the proposed strategy, we used the same three datasets [2] from the UCI open-source repository [1]. The details of the dataset are listed in Table 1

4.1 Implementation

The proposed model is a multistage filtering model and works in two stages. Firstly, the ANOVAF value method identifies the candidate features from the entire feature set. Later in the subsequent stage, the ensemble-based wrapper method is used to select the best set of features subset from these candidate features.

In ANOVAF filter method, the F-test score for all features of the dataset is calculated based on the relationship between the features and the target variables. These F-scores are then normalized to get a more accurate and holistic view. Min–Max normalization is used to normalize the data. In this process, scores are normalized for each feature, one by one. The minimum score gets recorded as 0, while the maximum score gets recorded as 1. For all the remaining values, a value between 0 and 1 is considered. Then, all the features of the dataset are sorted in the order of normalized F-test values. All the top k-features with normalized F-score value above

Table 1 Dataset description

Dataset name	No. of instances	No. of features	Type of dataset
Clean	476	167	Binary
Libras movement	360	90	Multi-class
Ionosphere	351	33	Binary

the threshold value are selected. For our model, we have set the threshold to 0.005. These k-features are then used as an input set for the next stage.

In the second stage, the select k-features are then evaluated by the adaptive boosting (AdaBoost) classifier ensemble method. The adaptive boosting classifier attaches a specific weight to each feature. The greater the value of the weight, the more the predictive value of the feature. The optimal number of feature subset is selected after several repetitive adaptive iterations. The number of features selected and marked as relevant by each of the methods is shown in Table 2. We have used a Random Forest classifier to measure the classification correctness of the feature subset.

To compare the proposed framework with the existing framework [2], we have used the same four classification measures used by Venkatesh et al. [2] to measure the classification accuracy of the proposed framework. We have used F1 Score, Precision, Recall and Accuracy. Table 2 explains the values of all these scores for the following methods.

- Original—Scores based on all the feature set
- ANOVAF—Scores based on only ANOVAF feature selection
- AdaBoost—Scores based on only AdaBoost feature selection
- Decision Tree—Scores based on only decision tree feature selection

Table 2 Results and classification metrics for all datasets

Dataset name	Method	No. of features	F1-score	Recall	Precision	Accuracy
Ionosphere	Original	33	93.13	92.71	93.71	93.4
	ANOVAF value	27	94.74	94.82	95.23	94.82
	AdaBoost	19	92.99	93.1	93.44	93.1
	Existing model	15	95.09	94.65	95.70	95.28
	Hybrid (proposed)	13	95.63	95.68	95.97	95.68
Clean	Original	167	89.28	88.71	90.09	89.92
	ANOVAF value	143	92.64	92.7	93.6	92.7
	AdaBoost	70	92.92	93.03	93.47	93.03
	Existing model	75	90.21	89.78	90.79	90.79
	Hybrid (proposed)	62	94.9	94.93	95	94.93
Libras movement	Original	90	78.01	78.15	80.96	78.15
	ANOVAF-value	88	76.4	77.77	79.93	77.77
	Decision tree	35	80.35	80.67	81.68	80.67
	Existing Model	40	86.60	87.50	87.02	87.78
	Hybrid (Proposed)	32	87.97	88.23	88.64	88.24

- Existing Model—Scores of the existing model for the same data proposed by Venkatesh et al. [2]
- Hybrid Model—Score based on our proposed hybrid model

4.2 Results

The results of the work in illustrated in Table 2. It explains the values of all four classification measures for all the methods, including filter method, wrapper method, existing approach and the proposed approach. It is apparent from the values of the classification metrics that the proposed method outperforms the existing previous model [2] in all the four classification measures for all the datasets. For example, for the clean dataset, the accuracy of the proposed model is **94.93%** vs 90.79% of the existing model. Even the accuracy is achieved with a smaller number of features. The proposed model predicts higher accuracy with **62** features as compared to 75 features used by the existing model. The proposed method is better than the individual methods and classification metrics and the number of features.

Equally for other datasets like Libras Movement and Ionosphere, the proposed feature selection model outclasses all the classification metrics with a smaller number of features. A receiver operating characteristic (ROC) curve is plotted for both the binary datasets, clean and ionosphere. A ROC curve [15] shows the performance of a classification model at various classification threshold. The curve plots two parameters, namely true positive rate and false positive rate (Figs. 2 and 3).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (4)$$

Fig. 2 ROC clean dataset

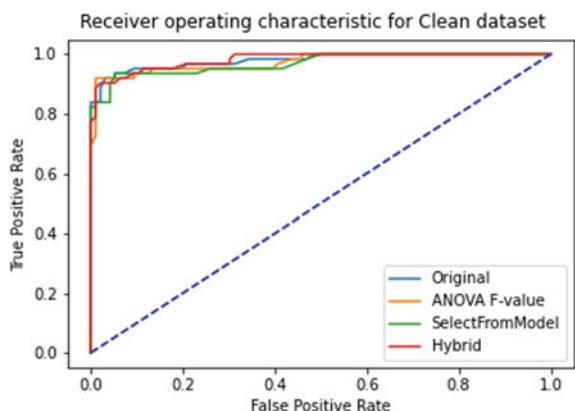
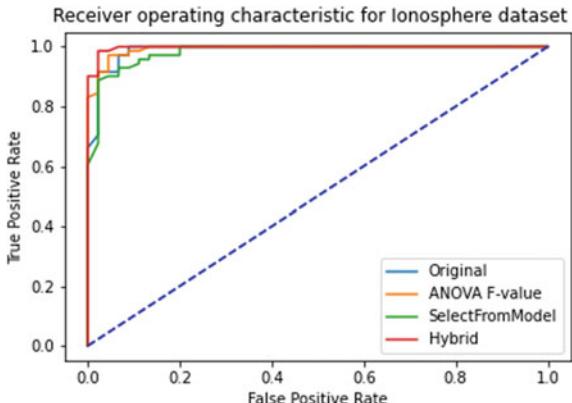


Fig. 3 ROC ionosphere

where

TPR = True Positive Rate

FPR = False Positive Rate

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

5 Conclusions

This paper advocated a hybrid feature selection method that utilizes both the filter method and wrapper method. We compared our model with an existing feature selection model over the same benchmark datasets and obtained better classification accuracy across all the datasets. For the Clean dataset, our model provided **94.93%** accuracy over 90.79%. For the Libras Movement dataset, our model worked with an accuracy of **88.24%** over 87.78%. For the Ionosphere dataset, our mode's accuracy was **95.68%** over 95.28%. We improvised on the existing proposed model [2] and demonstrated that depending upon the type of dataset, ensemble-based wrapper methods can enhance the performance of the feature extraction method. Future studies on the evaluation of the proposed method on time-series financial dataset need to be conducted and discover the performance.

References

1. Dheeru D, KarraTaniskidou E (2017) UCI machine learning repository
2. Venkatesh B, Anuradha J (2019) A hybrid feature selection approach for handling a high-dimensional data. Innovations in Computer Science and Engineering, pp 365–373. Springer, Singapore
3. Jashki A, Makki M, Bagheri E, Ghorbani AA (2009) An iterative hybrid filter-wrapping approach to feature selection for document clustering. Proceedings of the 22nd Canadian Conference on Artificial Intelligence (AI'09)
4. Hu J, Xiong C, Shu J, Zhou X, Zhu J (2009) An improved text clustering method based on hybrid model. *Int J Mod Educ Comput Sci (IJMECS)* 1(1):35
5. Yang Y, Liao Y, Meng G, Lee J (2011) A hybrid feature selection scheme for unsupervised learning and its application in bearing fault diagnosis. *Expert Syst Appl* 38(9):11311–11320
6. Hsu HH, Hsieh CW, Lu MD (2011) Hybrid feature selection by combining filters and wrappers. *Expert Syst Appl* 38(7):8144–8150
7. Niu T, Wang J, Lu H, Yang W, Du P (2020) Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Appl* 148:113237
8. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF (2016) A new hybrid filter-wrapper feature selection method for clustering based on ranking. *Neurocomputing* 214:866–880
9. Lin X, Yang F, Zhou L, Yin P, Kong H, Xing W, Lu X, Jia L, Wang Q, Xu G (2012) A support vector machine recursive feature elimination feature selection method based on artificial contrast variables and mutual information. *J Chromatogr B* 910:149–155
10. Tan F, Fu X, Zhang Y, Bourgeois AG (2008) A genetic algorithm-based method for feature subset selection. *Soft Comput* 12(2):111–120
11. Lu H, Chen J, Yan K, Jin Q, Xue Y, Gao Z (2017) A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* 256:56–62
12. Rouhi A, Nezamabadi-pour H (2017) A hybrid feature selection approach based on ensemble method for high-dimensional data. 2017 2nd conference on swarm intelligence and evolutionary computation (CSIEC). IEEE, pp 16–20
13. Rouhi A, Nezamabadi-Pour H (2020) Feature selection in high-dimensional data. In Amini M (eds) Optimization, learning, and control for interdependent complex networks. Advances in Intelligent Systems and Computing, vol 1123. Springer, Cham
14. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm, vol 96
15. Li P, Abdel-Aty M, Yuan J (2020) Real-time crash risk prediction on arterials based on LSTM-CNN. *Accident Analysis & Prevention* 135:105371

An Innovative Approach to Establish, Maintain and Review Quality Standards in Higher Education through Quality Assurance Tool



Sangeeta Arora and Anil Ahlawat

Abstract A lot of diversity is found in the education system in the world at present. There is an earnest need to improve and ensure quality to achieve excellence in this diverse education system. In the fast-changing environment, it is also necessary to follow already set quality benchmarks as well as creating new benchmarks. For quality Assurance, different accreditation standards across the world have been laid down such as the Accreditation Board for Engineering and Technology (ABET), National Assessment and Accreditation Council (NAAC), National Board of Accreditation (NBA). These accreditations evaluate quality based on factors like Program Objectives, Teaching-Learning process, Students Profile, Infrastructure, Faculty Empowerment, Student Empowerment, Health Care, and many more. A fuzzy inference system was developed in MATLAB using these quality factors to evaluate the quality index of any Higher Education Institute. To test this tool, a survey was conducted by presenting a questionnaire consisting of 30 questions to the people working at various capacities such as administration and faculty at Higher Education Institutes (HEIs) of different parts of the world. The sample adequacy is checked through Kaiser-Meyer-Olkin (KMO). The study also depicts the factors which are mostly considered by HEIs.

Keywords Accreditation · Higher education institutes · Quality standards · Education system · Fuzzy inference system

1 Introduction

Education plays a vital role in everyone's life on their way toward success and progress. Education changes the world of individuals and contributes to the improvement of society and the country. The job of HEIs is to convey ability-based instruction to understudies for their future. HEIs continuously work toward improvement to provide quality education. For this purpose, they need to undergo a quality

S. Arora · A. Ahlawat (✉)

Department of Computer Applications, KIET Group of Institutions, Ghaziabad, India
e-mail: anil.ahlawat@kiet.edu

measurement plan through a theoretical assessment. To establish the standards, many autonomous bodies like. ABET, NAAC, NBA, etc. are working continuously toward providing a better assessment of quality.

ABET is one of the non-governmental and non-profit organizations for quality evaluation having ISO 9001:2015 certification. Programs accredited regionally in the USA or nationally accredited institutes for other countries are eligible to get ABET. (<https://www.abet.org>). University Grants Commission (UGC) in India has built a self-sufficient body National Assessment and Accreditation Council (NAAC) for the assessment of the nature of advanced education (<http://mhrd.gov.in/college-and-advancededucation1>). The significant worry of the NAAC is to evaluate the nature of advanced education establishments, colleges, and so forth and certify them. The NAAC accreditation assumes an essential job in the recognizable proof of qualities and shortcomings of the advanced education framework and this audit helps toward progress. All the NAAC licensed foundations must keep up an Internal Quality Assurance Cell (IQAC) for consistent support of the value (<http://www.naac.gov.in>). All India Council for Technical Education (AICTE) setup in November 1945 as a national-level Apex Advisory Body. It conducts review assessing accessible offices for specialized instruction and to advance improvement in India in an organized and incorporated way. AICTE established NBA in 1994 for quality assessment at diploma, graduate, and postgraduate levels in Engineering and Technology, Pharmaceutical, management disciplines. On 7th January 2010, the NBA started working as an autonomous body for quality assurance in engineering and technical disciplines. NBA is the permanent signatory member of Washington Accord for tier 1 institutes. The Washington Accord provides an international standard to accredited institutes. (<http://www.nbaind.org/default.aspx>).

At a higher educational level, quality management detects flaws in educational areas and works toward achieving the optimum standards and satisfaction of customers [13, 16, 12]. The Establishment of IQAC is a system which ensures that consistent improvement of quality in the education system. The strategy of IQAC is to ensure the continuous performance of administrative and academic tasks. Its main function is to scale the standards for different activities in the organization [5]. Continuous assessment of learning activities scales the standards of quality and the factors to affect the teaching-learning process [1] determined the role of IQAC for maintaining quality standards. For this, [2] collected data from 29 colleges, and based on received responses, a conclusion was drawn that various activities improved the quality of the teaching-learning processes.

Kahveci et al. [8] designed a strategic Information System to execute the quality efficiently. This system incorporates three modules: strategic management, process management, and measurement monitoring. The strategic management module helps to identify a goal for the quality assertion in higher education with the help of external trends. The process management module is defining processes for the achievement of identified goals and measurement monitoring module is for monitoring the performance success of goals. The assistance is a tool for peer reviews among faculty members through (strength, weakness, opportunity, and threat) analysis on 27 samples [17]. It was emphasizing on the professional development of teacher

learning for professional development [6]. All Quality administration standards for the improvement and establishment of value which moved in the direction of the drawn-out objectives plans, and usage [18], evaluated the total quality management on different factors education, research, career support, and other amenities like canteen, gym, etc. with the viewpoint. Karahan [10] presented the model to evaluate the current state of education. This evaluation helps to define the action for improvement in quality in the education system [4].

Stura et al. [15] emphasized that the Accreditation process done by the committee which does not belong to the institute is a way to check the quality of study programs. In continuation, this accreditation excellence was accomplishing by standardizing work. The national accreditation impacts the enhancement of practices of HEIs with the viewpoint of management. Management examines the assessment given by these accreditation frameworks and constitutes the practices based on that. Kooli [11] finds that these national accreditation practices are doing for the liability. This does not effectively make up for the development of HEIs. Snijders et al. [14] examined that quality relationship among students and faculty improves the loyalty of students. This student's expectations must be evaluated regularly for quality enhancement. Diez et al. [3] focused on two dimensions, i.e., educational policy, management. The actions are carrying in direction achieving quality education and taking care of quality. Another focus is on how management contributes to the enhancement of quality education. Generally, quality measurement is subjective, it is difficult to measure the quality quantitatively on different aspects of the institute. Quality has linguistic value, i.e., low, high, etc., which varies person to person. For this type of process, a system is developed using different soft computing techniques. Fuzzy logic is a soft computing technique that was first introduced by Jadeh in 1994 to work with systems having uncertainties [7].

2 Methodology

2.1 Sample Characteristics

The proposed system finalized factors with the help of different standards, i.e., ABET, NAAC, NBA, etc. (shown in Fig. 1). The sample data was collected from different institutes of the world, persons holding different positions administrative positions, faculty, etc. Most respondents having more than 5 years' experience at their position and gone through the process of accreditation standards.

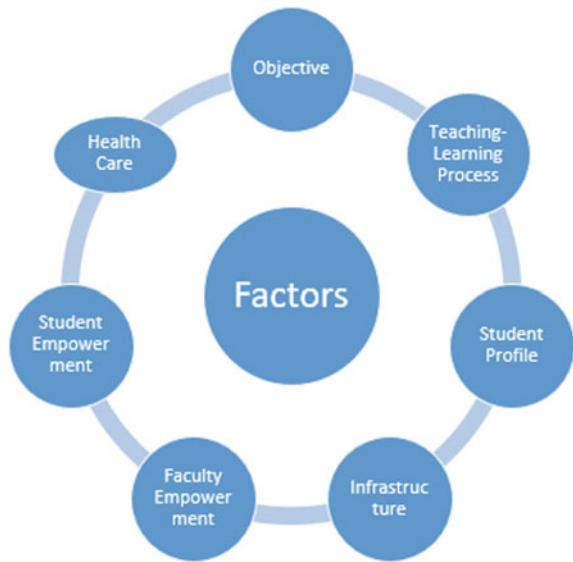


Fig. 1 Factors for evaluation of the quality

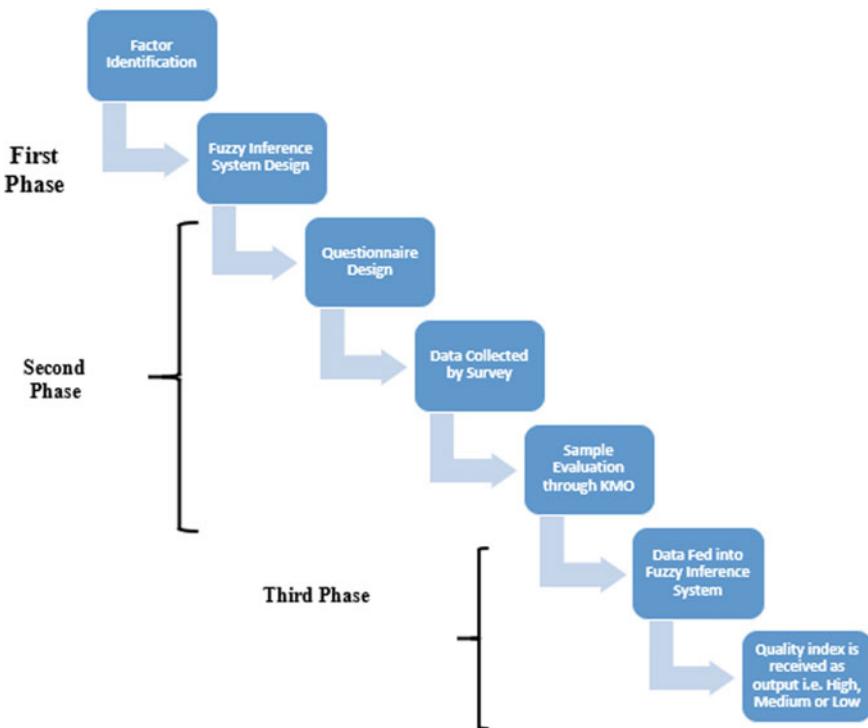


Fig. 2 . Methodology followed

2.2 Proposed Model

The proposed model has 3 phases. Figure 2 is showing the methodology step by step. The first step is the identification of quality factors. A Fuzzy Inference System designed using quality factors. To test the system data collected through a questionnaire. This information took care of into the Fuzzy Inference System and the quality record checked. A system will use for the quality indicator of HEIs and help the institutes to improve in their gray areas.

In the first phase, a Mamdani fuzzy inference system developed for evaluation of the quality of HEIs. The identified inputs are taken from survey data collected from several academicians and the administrative staff of several HEIs. In the next phase, the survey data collected from several teaching professionals and administrative staff of HEIs. The sample was analyzed in SPSS 20.0 using Kaiser-Meyer-Olkin (KMO) [9] measure of sampling adequacy. Exploratory factor analysis was done using KMO which used to simplify with an expectation of underlying variable advancement to manifest variables. Cronbach's alpha value for our sample is 0.869 which shows the acceptability. These inputs have three membership functions, i.e., LOW, MEDIUM, and HIGH. In this, each section can have an overlapping of 10% to 50%. All crisp values are fuzzified using the fuzzifier, which gives input to the inference system. This inference system collects rules from rule base based on fuzzy input. The output of the inference system is de-fuzzified to obtain the final output. For the fuzzy inference system, 2187 unique rules generated. These rules make the system as time-consuming and relatively incompetent. Because of this, we store rules obtained from the survey.

$$\begin{aligned}
 \text{LOW} &= \begin{cases} 0 & x < 1 \\ \frac{x-1}{2.5-1.0} & 1 \leq x \leq 2.5 \\ \frac{4-x}{4-2.5} & 2.5 \leq x \leq 4 \\ 0 & x > 4 \end{cases} \\
 \text{MEDIUM} &= \begin{cases} 0 & x < 3 \\ \frac{x-3}{5-3} & 3 \leq x \leq 5 \\ \frac{7-x}{7-5} & 5 \leq x \leq 7 \\ 0 & x > 7 \end{cases} \\
 \text{HIGH} &= \begin{cases} 0 & x < 6 \\ \frac{x-6}{8-6} & 6 \leq x \leq 8 \\ \frac{10-x}{10-8} & 8 \leq x \leq 10 \\ 0 & x > 10 \end{cases}
 \end{aligned}$$

In the third phase, the survey results fed into a fuzzy inference system to generate the quality index.

3 Result and Discussion

A Fuzzy Inference System utilized the rule base to assess the quality list of the framework. As discussed above that these questions categorized into seven factors: objective, teaching-learning process, student profile, infrastructure, faculty empowerment, student empowerment, and health care. Each factor has a different number of questions. The responses received normalized on a scale of 1 to 10. The rating of each factor is shown in Table 1. Based on the percentage of yes and no evaluated from the received responses, i.e., yes, no, and the total number of questions in particular factor the factor-wise percentage of yes and no and prioritization of factors is shown. This showed that most of the institutes set their objective to achieve the quality and they are taking care of healthcare activities. The second priority of HEIs is Student Profile toward the achievement of excellence. HEIs are working toward Student Empowerment for the overall development of the students. The next primacy of HEIs is Faculty Empowerment and Infrastructure of the institute. The survey results show that less care toward the teaching-learning process. The percentage-wise chart is shown in Fig. 3. In results, the objective has an important role in the working of HEIs. If any Education Institute has its vision mission and planning to achieve the objective, the Institute can succeed. Healthcare activities are the primary for HEIs. For this, many HEIs are working toward the eco-friendly environment, motivating students and staff for plantation, and providing healthcare services within the campus. HEIs are following a transparent admission system, organizing orientation programs, and support for SC/ST which strengthen the student profile. For students, many activities, i.e., NSS/NCC, career counseling, competitive exam classes organized to empower the students. HEIs are motivating students by giving scholarships for their achievements.

Table 1. Major factors rating

Major factors	High	Medium	Low
Objective	51	0	49
Teaching-learning process	45	33	22
Student profile	38	30	32
Infrastructure	53	21	26
Faculty empowerment	48	25	27
Student empowerment	49	26	25
Health care	38	33	29

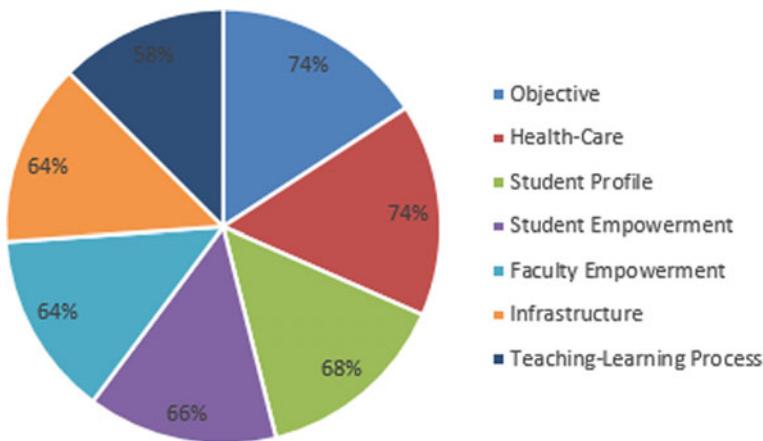


Fig. 3 Priority given in present scenario

4 Conclusion

In this paper, a fuzzy inference system developed using MATLAB for measurement of the quality of HEIs. For this, the first seven factors identified. The survey was done, and the questionnaire sent to 145 persons, are faculty members or at administrative positions. A hundred responses have received from respondents. The sample adequacy checked through Kaiser-Meyer-Olkin (KMO). A fuzzy inference system introduced for the evaluation of quality. The percentage-wise result was evaluated based on responses. The discussion was done based on percentage and their build upon issues. It is found that the major focus of HEIs was on setting up objective and healthcare issues and the teaching-learning process will be toward improvement by setting learning outcomes and association with alumni and different stakeholders for improvement and bridge the gap between industry and academia. The weakest factor is the Teaching-Learning Process, which must be the strongest factor in HEIs. It is necessary to have audits during academic sessions by internal committees formed by the institute for improvement.

References

1. Barrio MIP, Escamilla AC, García MNG, Fernández EM, García P (2015) Influence of assessment in the teaching-learning process in the higher education. *Procedia—Social and Behavioral Sciences*, Vol No. 176:458–465
2. Sawant DG (2016) Role of IQAC in maintaining quality standards in teaching, learning and evaluation. *Pacific Science Review B: Humanities and Social Sciences* 2(2):66–69
3. Diez F, Villa A, Lopez AL, Iraurgi I (2020) Impact of quality management systems in the performance of educational centres: educational policies and management process. *Heliyon* (Elsevier) 6(4)

4. Gullickson AM, King JA, LaVelle JM, Clinton JM (2019) The current state of evaluator education: A situation analysis and call to action. *Evaluation Program Plann* 75:20–30
5. Gupta AK, Goyal R, Panjla AK (2016) IQAC as a tool for improving quality education in higher educational institutes. *Int J Latest Trends in Engineering and Technology* 7(2):544–547
6. InkenGast SK, Jan T, Veen VD (2017) Team-based professional development interventions in higher education: A systematic review. *Review of Educational Res* 87(4):736–767
7. Jadeh LA, Klir GJ, Yuan B (1994) Fuzzy sets, fuzzy logic, and fuzzy systems. Publisher World Scientific, ISBN 978-981-02-2421-9
8. Kahveci TC, Uygun O, Yurtserver U, Ilyas S (2012) Quality assurance in higher education institutions using strategic information systems. *Procedia—Social and Behavioral Sci* 55:161–167
9. Kaiser H (1974) An index of factor simplicity. *Psychometrika* 39:31–36
10. Karahan M, Mete M (2014) Examination of total quality management practices in higher education in the context of quality sufficiency. *Procedia—Social and Behavioral Sci* 109:1292–1297
11. Kooli C (2019) Governing and managing higher education institutions: The quality audit contributions. *Evaluation and Program Planning* (Elsevier), Vol 77
12. Middlehurst R (1995) Leadership quality, and institutional effectiveness. *Higher Education Quart* 49(3):267–285
13. Seymour D (1991) Beyond assessment: Managing quality in higher education. *Assessment Update: Progress, Trends, and Practices in Higher Education* 3(1):1–10
14. Snijders I, Wijnia L, Rikers RMJP, Loyens SMM (2020) Building bridges in higher education: Student-faculty relationship quality, student engagement, and student loyalty. *International J Educational Research* (Elsevier), Vol. 100
15. Stura I, Gentile T, Migliaretti G, Vesce E (2019) Accreditation in higher education: Does disciplinary matter? *Studies in Educational Evaluation* (Elsevier) 63:41–47
16. Tannock JDT (1992) A new approach to quality assurance for higher education. *Higher Education Quart* 46(1):108–123
17. Thomas S, Chie QT, Abraham M, R. S. J., Beh, L. (2014) A qualitative review of literature on peer review of teaching in higher education an application of the SWOT framework. *Rev Educational Res* 84(1):3–46
18. Todorut AV (2013) The need for Total Quality Management in higher education. *Procedia—Social and Behavioral Sci* 83:1105–1011

Assessment of 3-Dimensional Hand Pose by PosePrior Network for Images



Pallavi Malavath, Nagaraju Devarakonda, Zdzislaw Polkowski,
and Challapalli Jhansi rani

Abstract Deep learning is a function of artificial intelligence which imitates the human brain. In this work, we introduced a technique which estimates 3D HPE for images. This technique overcomes uncertainties caused by missing of depth information in images. Finally, we introduce a deep neural network that acquires knowledge about 3D articulation of a Network –Implicit method. This network gives good 3D hand pose estimation along with the detection of key points in images. We have compared our results to previous experiments, our proposed system gives impressive results. We present a dataset of large-scale 3D HPE that depends on the synthetic human hand models to train the networks. Along with other datasets, sign language recognition is also taken as another dataset to illustrate the viability of 3D HPE on images.

Keywords Convolution Neural Networks (CNN) · Computer Vision (CV) · Hand Pose Estimation (HPE) · Hand Pose Discriminator (HPD) and Hand Pose Generator (HPG)

1 Introduction

The role of HPE is the finding key points/joints of the hand from various video frames or from single image. The process usually involves in detecting the joints on a human hand, later the analysis of hand pose is done by deep learning algorithms. For humans, hands are the important operating tools. So therefore, the articulation, orientation

P. Malavath (✉) · N. Devarakonda · C. J. rani

School of Computer Science & Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

e-mail: pallavimalavath.20phd7126@vitap.ac.in

C. J. rani

e-mail: jhansirani.20phd7097@vitap.ac.in

Z. Polkowski

Technical Sciences, Jan Wyzykowski University, Polkowice, Poland

e-mail: z.polkowski@ujw.pl

and its location are important in many applications. The innovation of deep learning makes people to apply the techniques in various fields of computer vision. For man-machine interaction gesture recognition, sign language, object handover in robotics are using hand as an input device is shown in Fig. 1. The application of HPE which is used in AR/VR headset has shown. Because of many uncertainties and self-occlusion, full 3D HPE from images is difficult. Hence specific equipments like markers or data gloves are used. Latest works depend on depth image.

In this work, we introduce a technique to grasp full 3D HPE related to the images without using any specific equipment. To overcome uncertainties in image data we use deep networks. To cover the crucial subtasks regarding 3D pose, our technique has three deep networks. Generally to localize the hand, hand segmentation is required. Therefore the task of first network is hand segmentation. Localization of key points is done by second network. Extracting the 3D pose from the 2D key points is done by the third network. To make this task easy we present a canonical hand pose representation is shown in Fig. 2.

Unavailability of data is the main problem in 3D pose estimation. To train the network, dataset with ground truth is required. Due to the unavailability of data set,

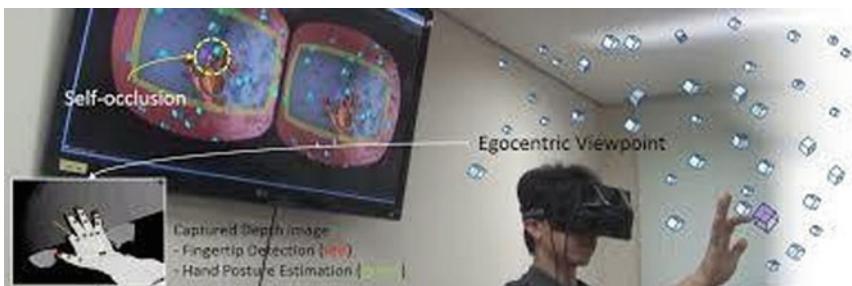


Fig. 1 Application of HPE

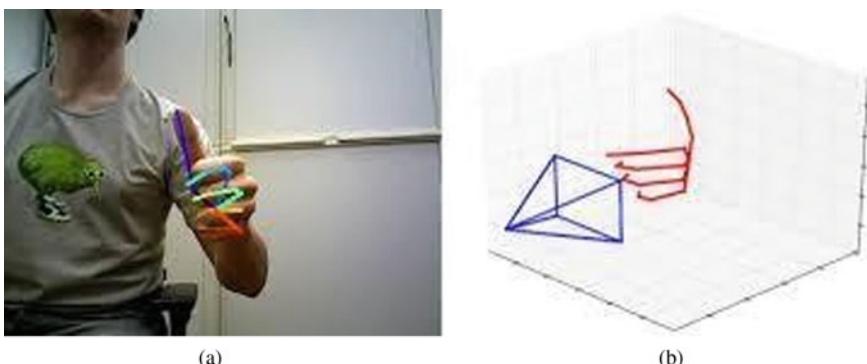


Fig. 2 **a** Given an RGB image with 2-dimensional estimation of hand joints **b** 3-dimensional estimation of hand joints

we introduced a synthetic data set with different augmentation options. Outcomes of 3D HPE give optimistic results, related to both quantitatively and qualitatively on some datasets. We also explain the use of 3D HPE which is used in sign language recognition.

2 Literature Review

2.1 Hand Pose Estimation

In this paper, we proposed a skeleton for 2-D/3-D HPE. So, kinematic model is suited for the skeleton formation for joints of a hand. For pose estimation, we have:

Shape model: This model describes the shape of a hand.

Kinematic Model: The kinematic model is a graph model that describes hand structure/human body structure like skeleton representation. It consists of a set of joints and the orientations of the fingers to represent hand structure. The model has the advantage of being able to represent various textures and shapes.

This process consists of 3 basic building blocks. First, within the image, the localization of the hand is done by segmentation network (HandSegNet). Correspondingly the input image is cropped based on the hand mask and feeds to PoseNet. Localization of hand joint is exemplified by score maps, eventually, PosePrior network determines 3D structure of the image based on score maps. The overall approach is shown in Fig. 3.

With the increasing interest in 3-dimensional HPE, many models are proposed for extensive variety of hand shapes. There is different popular hand body structures used in deep learning-based 3D HPE methods for recovering 3D hand mesh is shown in Fig. 4.

Sclaroff and Athitsos [1] proposed a technique based on the detection of a single frame which is related to the chamfer matching and edge maps. The research is concentrated on estimation of the hand pose by using low-cost depth cameras. Sharp et al. [2] created a precised number of hand poses and overcome from the depth

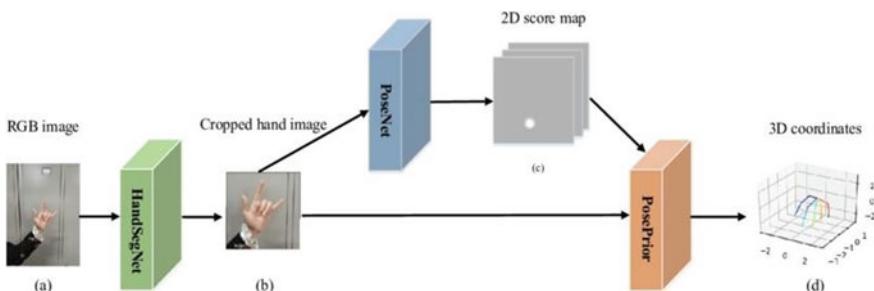


Fig. 3 Process of PoseNet

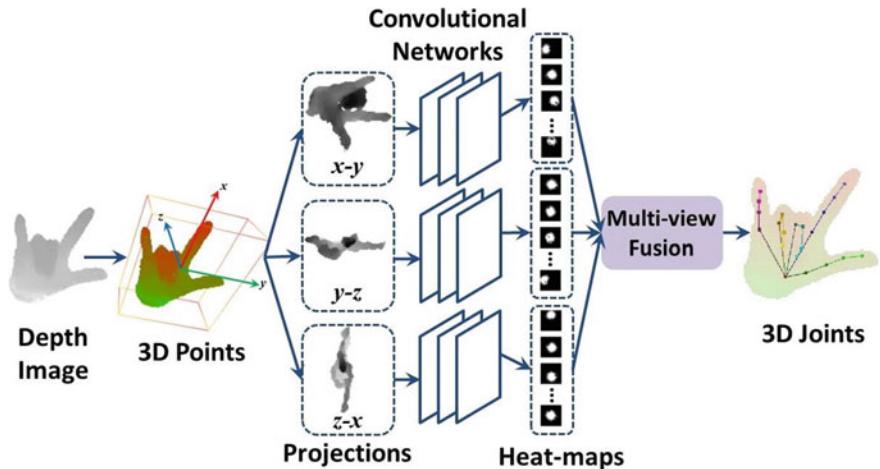


Fig. 4 Depth-based HPE

image. Oikonomidis et al. [3] proposed an approach related to (PSO) Particle Swarm Optimization. Tompson et al. [4] used convolution neural network for the detection of 2D hand key points, which is constrained on multi-resolution image pyramid. By resolving the problem in inverse kinematics optimization the 3D pose is recovered. Zhou et al. [5] instead of Cartesian coordinates it evaluates angles between the bones of kinematic chain. Ober wager et al. [6] from a given hand poses estimate it utilizes a convolution neural network (CNN) technique that can synthesize the depth map. By reducing the distance between the depth image and observed image it allows to successively refine HPE. Techniques related to Zhou et al. [7] or Ober wager et al. [6] have shown the probability to encode the correlations between compressing bottleneck and hand key point coordinates. Caggianese et al. [28] give a précis HPE that can improve the experience of users in VR systems by empowering the performance of realistic hand movements which is virtual, and better consideration of human actions via HCI systems which enables the interaction between computers and humans.

2.2 2-Dimensional Hand Pose Estimation

The research in the area of CNN has made huge progress in the last years. From color input image, Toshev and Szegedy [8] proposed a convolution neural network that directly regresses 2D Cartesian coordinates. Thomson et al. [9] focused on regressing score maps. Figure 6 describes the two HPE. Here, estimation is done based on the location of the key points/joints. Sridhar et al. [10] followed an approach depend on finger actions. Yin et al. [55] utilized HPE to design a model which can recognize sign

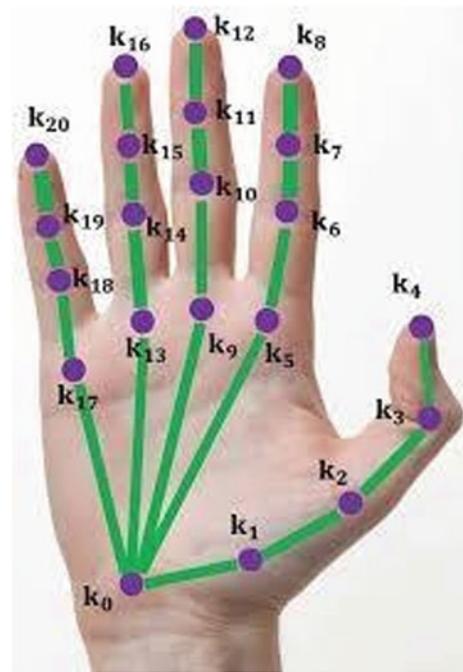


Fig. 5 A hand model with key points



Fig. 6. 2D hand pose estimation

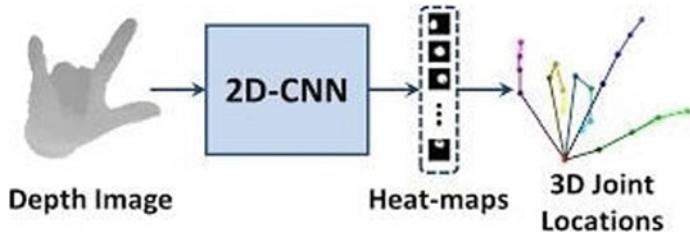


Fig. 7 Real-time 3D HPE using CNN

language. Chaang et al. [11] followed finger tip detection & identifies a technique to read alphabets shown by the finger. Rohrbach et al. and Shlizerman et al. used hand and body pose estimations for identifying body movements of the piano player is shown in Fig. 7.

2.3 3-Dimensional Hand Pose Estimation

Very close to our approach many researchers have used pipeline with two parts [8, 10, 12, 13]. To utilize the selective power of convolution neural network techniques, they first extract the 2D keypoints, and then the extractions from 2 to 3D space are done. For better representations various approaches have been introduced, Chen et al. [14] by using a 2D to 3D correspondences, nearest neighbor matching is used based on a given prediction. Tome et al. [15] generated a probabilistic 3-dimensional human pose model depends upon PCA bases. Paavlakos et al. [16] followed a volume-based technique which uses pose estimation depends on voxel prediction in a coarse to fine style, gives a good depiction of data. Several approaches have been introduced that can apply techniques of DL for processing 2D key points to 3D HPE [7, 17, 18]. Bogo et al. [19] enhances the re-projection error between 2-dimensional prediction and 3-dimensional joint/keypoint position of a statistical shape of human body. All these researches are related to 3D HPE, which is significantly harder because of self-occlusion and stronger articulation, as well as insufficient data availability. Baiek et al. [21] followed GAN [20] to predict human hand pose by building relation between 3D hand pose models and depth disparity maps. In this research one domain is 3D representation of hand key points and the other is depth map of a human hand. Baek et al. utilized a Hand Pose Discriminator (HPD) & Hand Pose Generator (HPG) have been used in their research. The task of the HPG is to create a hand, depends on 3D depiction of keypoints and the task of HPE is to depict the 3D hand pose, which is related to depth map is shown in Fig. 8.



Fig. 8. 3-Dimensional HPE

3 Representation of Pose of a Hand

To infer a 3D hand pose of a single hand in a given color image $I \in \mathbb{R}^{P \times Q \times 3}$. Taking coordinates $w_i = (x_i, y_i, z_i)$ to define the hand pose, which describes the annotations of J joints related to 3-dimensional space; $i \in [1, J]$ taking $J = 22$.

There is a scale ambiguity, among other uncertainties. By training a network to determine normalized coordinates

$$w_i^{\text{norm}} = \frac{1}{k} \cdot W_i \quad (1)$$

where $k = \|w_{k+1} - w_k\|^2$ is a constant which classifies interval between perpetual unit length and joints/keypoints. To the first bone related to the index finger, we have chosen k such that $k = 1$. We have taken 3D coordinates to translate the representation of human hand poses. This can be done deducting the position of a defined root joint keypoint. The 3-dimensional coordinates and relative are given by

$$W_i^{\text{rel}} = W_i^{\text{norm}} - W_r^{\text{norm}} \quad (2)$$

where ‘ r ’ is the root index. We use $r = 0$ in our work because the palm joint is the stable landmark.

This picture shows PosePrior network of the proposed architecture. The two streams which are symmetric determine the coordinates of the viewpoint based on canonical coordinate system. The mixing of both extractions gives an estimation of (w_{rel}).

3.1 3-Dimensional HPE

From an input image, we predict the 3-dimensional hand coordinates w^{rel} . An outline about the HPE procedure is shown in Fig. 9: We provide details about the components in the following sections.

Segmentation of the hand with HandSegNet

We establish new network architecture for hand segmentation, which is predicated by a person locator by Wei et al. [2]. They have shown a problem of 2-dimensional person estimation, by determining the score map based on the center part of the human body. Because of drastic changes in the size of a hand across the images it depends mainly.

[...]on the image articulation and the hand localization. The HandSegNet of network related to Wei et al. [2] trained to the dataset of hand pose which we have taken. The mask of the hand is given by HandSegNet permits us to normalize and crop the input image (down sampling), thereby simplifying the learning job of the PoseNet.

Scoremaps of the Keypoints with PoseNet

We construct an annotation of 2D joints as prediction of 2-dimensional score maps $g = \{g_1(a, b) \dots g_J(a, b)\}$.

To estimate J score maps $g_i \in \mathbb{R} P \times Q$, we train the network and each score map contains the information about the feasibility of keypoints/joints present in spatial location. Similar to Pose Network related to Wei et al. [2], the network used an encoder-decoder architecture. An initial score map is estimated for a given image features depiction given by the encoder and later refinement of resolution is done. We

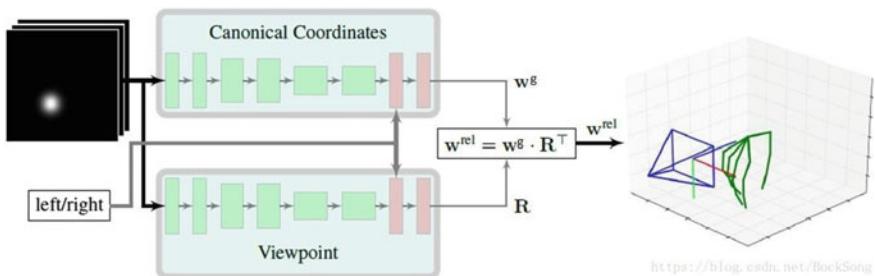


Fig. 9 The proposed architecture

instigated with weights related to Wai et al. [2], where it has applied and retraining of the network related to hand joints detection is done.

PosePrior Network for 3-Dimentional Hand Pose

PosePrior network estimates the normalized 3D coordinates and relative constrained on incomplete or score maps which are noisy $g(a, b)$. Finally, it must grasp the identification of possible articulations relative to the hand, and the most probable 3D configuration is given from the 2D evidence found on the score maps. In the proposed system, we are training the network to identify coordinates within the canonical frame, then it also predicts the transformation into the canonical frame.

The proposed relative normalized coordinates that uses frame w^g , which relates correlates to w^{rel} in the following way: The presentation is as follows

$$w^{g*} = R(w^{\text{rel}}) \cdot w^{\text{rel}} \quad (3)$$

with $R(w^{\text{rel}}) \in \mathbb{R}^{3 \times 3}$, a 3D rotation matrix is performed in two steps. First, finds out the R_{xz} (rotation) around the x-axis and z-axis and the keypoint/joint Wa^{g*} is positioned along the y-axis related to the canonical frame.

$$R_{xz} \cdot w_a^{g*} = \lambda \cdot (0.1.0)^T \text{ with } \lambda \geq 0 \quad (4)$$

After that, the rotation Ry on the y-axis is determined su

$$R_y \cdot R_{xz} \cdot w_a^{g*} = (\eta, \zeta, 0) \quad (5)$$

Here $\eta \geq 0$ to the designated keypoint index 0. The transformation between original frame and canonical frame is shown as

$$R(w^{\text{rel}}) = R_x R_y \quad (6)$$

To handle the uniformity between right and left hands, we twist the right hands toward z -axis.

$$w_i^g = \{(x_i^{g*}, y_i^{g*}, z_i^{g*})\} \text{ if it is left hand} \quad (7)$$

$$w_i^g = \{(x_i^{g*}, y_i^{g*}, -z_i^{g*})\} \text{ if it is right hand} \quad (8)$$

This symbolizes our proposed canonical system. By using the definition of a canonical frame we trained our network for determining the 3-dimensional coordinates with respect to frame (w^g) and also determines $R(w^{\text{rel}})$ -rotational matrix by following axis & angle notation related to x, y, z coordinates. Estimation of transformation R , almost similar to determining different viewpoints of our sample data related to the canonical frame. This problem is referred to as estimation of the viewpoint. PosePrior network architecture has two parallel processing streams which is

shown in Fig. 9. The two streams follow identical architecture. With ReLU nonlinearities, it primarily processes the score maps J with the series of 6 convolutions and later the information related to left-hand side pose and right-hand side pose is linked with the feature prediction and later processing is done through FC layers. The streams with the FC layers give the estimations for viewpoints & canonical coordinates. Finally, two estimations finally leads to determination of w^{rel} .

Network Training—Training of HandSegNet

A softmax cross-entropy is taken for L_2 loss and L_2 belongs to PoseNet. A PosePrior network follows two terms related to loss, primarily a L_2 loss which is squared and relates to canonical coordinates.

$$L_g = \|w_{\text{gt}}^g - w_{\text{pred}}^g\|^2 \quad (9)$$

based on the ground truth w^g and network predictions w^g . Later, the L loss (squared) is manipulated on gt the canonical transformation matrix.

$$L_r = \|R - R_{\text{gt}}^g\|^2 \quad (10)$$

The unweighted sum of L_g and L_r yields the total loss function. For training the network We have used Tensor Flow (Fig. 10).

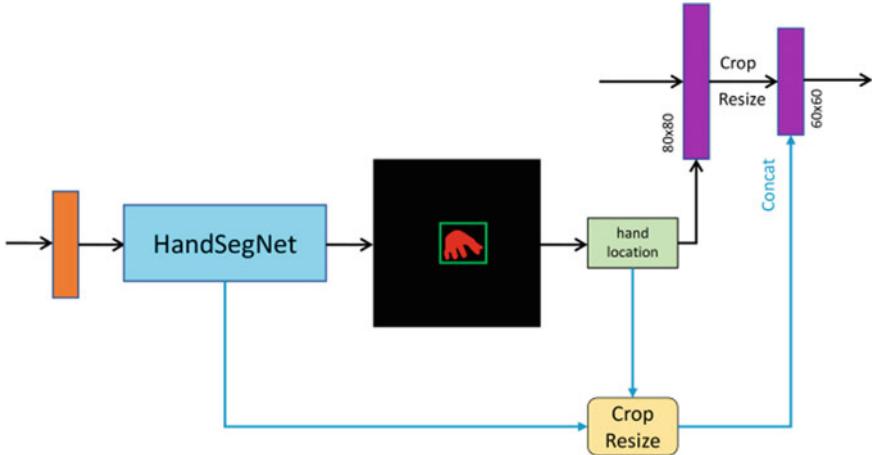


Fig. 10 Process of HandSegNet

4 Datasets for HPE

Two datasets are available which we can apply to the current problem. They come up with 3D notation and RGB images. The *Stereo Hand Pose Tracking Benchmark* provides 18,000 stereo pairs with 640×480 resolutions for both 3D and 2D notations with key points 21. This dataset shows a left hand of a single person with different lighting conditions and in different backgrounds. This dataset is divided into a training set with 15,000 images (*S-train*) and evaluation of about 4000 images (*S-val*).

Another dataset called *Dexter* provides 3129 images with two operations implementing various manipulations with the cuboid in an indoor setup. This dataset gives depth maps, notations for tips of a finger, color images & cuboid corners. Then, spatial resolution of a color image is 640×320 . We use Dexter dataset to explore the cross dataset generalization, due to the insufficient hand notation. This test is called *Dexter*.

By downsampling the datasets into 320×240 resolution, it is now compatible with our dataset. When we disclose pixel accuracies, the transformation of our results back to pixel coordinates is done.

5 Delineated Dataset–Hand Pose

The above-mentioned datasets were not adequate to train deep networks because of incomplete notations and limited variations. Therefore for training purpose, we compliment the dataset with new dataset. To overcome the destitute labeling performance, we use available 3D models is shown in Fig. 11.

Our dataset consists of 39 actions from the performance of 20 different characters. We have split the dataset into training set-(R-train) and a validation set-(R-val). Our split results into 4 characters with 8 actions related to training dataset and 16 characters belong to 31 actions for validation dataset. For our dataset, to enhance the visual diversity, we follow some settings by applying global illumination and apply lighting from 0 to 2 light sources, in a way that background image color is matched. Furthermore, we changed the intensities and the position of light. Finally, by using lossy JPG compression all the renderings are saved by the quality factor from low compression up to 70% is shown in Table 1.

Here, the top rows (GT) gives the performance of ground truth for PoseNet for cropped images & the bottom rows (Net) gives the results based on the generation of the cropping of hand (HandSegNet). HandSegNet trained on R-train and PoseNet trained on both S-train and R-train. AUC represents the errors related to the uncropped image.

Finally, our dataset contains 2128 images for evaluation and 41,269 images for training dataset having a resolution of 320×320 pixels. Our samples provides full notations of 21 key points skeleton model for both hands and 33 segmentation masks are considered for each palm and finger with three segments. In Figure every finger

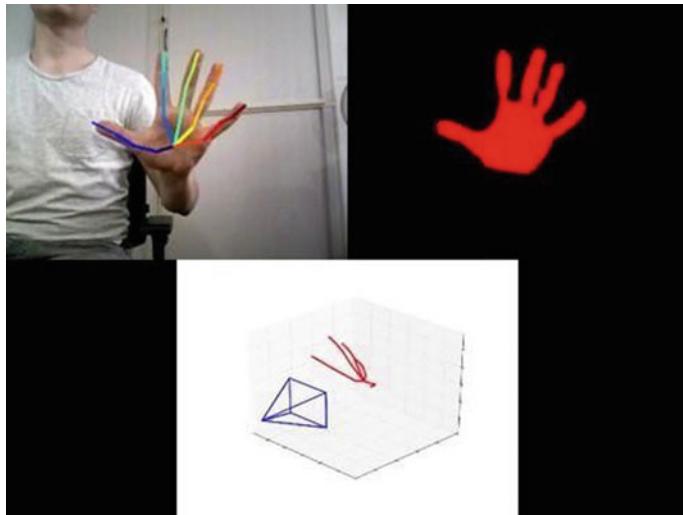


Fig. 11. 3D kinematic model

Table 1 Comparisons of Dexter, S-val and R-val

		AUC	EPE median	EPE mean
GT	R-val S-val	0.724 0.816	5.001 5.014	9.134 5.544
Net	R-val	0.678	6.746	18.741
	S-val	0.568	5.545	18.576
	Dexter	0.465	13.684	25.170

of the hand is represented by 4 key points: two intermediate key points, tip of the finger & the end location of the palm. Every key point has a information about whether it is occluded in the image or it is visible.

6 Experiments and Results

We examined the overall approach: (1) A HandSegNet (hand segmentation network) detects joints/keypoints of a PoseNet; (2) learned 3-dimensional PosePrior and the 3D HPE. Finally, we implemented this procedure on sign language recognition. We evaluated two different cases: first, with the ground truth (GT), hand is cropped and second, used predictions from HandSegNet for cropping the image.

The first case depicts the implementation of poseNet, whereas second one shows the implementation of 2D joint/keypoint estimation. The outcomes show how the proposed method worked on our synthetic dataset and stereo dataset. Dexter dataset

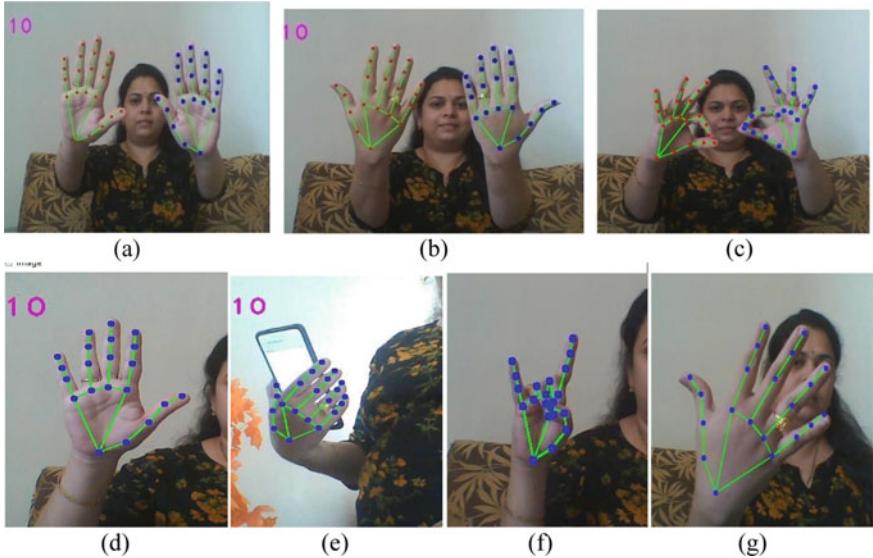


Fig. 12. 2D keypoint localization results show samples from Dexter, R-val and S-val

is strenuous because of unavailability of the dataset related to occlusions of the hand. In the training process, we did not have datasets with occlusion.

Figure 12 evaluative results of our proposed system are shown. RGB image is the input. The network estimates joints in 2-dimensional and produces its most likely 3-dimensional pose. Here, Some samples are from the dataset.

We recorded for qualitative evaluation, and some samples are from the dataset related to sign language, and one image sample is extracted from S-val.

Here, networks are trained on R-train. The average median error for every keypoint of a predicted 3D pose from various lifting techniques has given a noisy 2D ground pose.

Figure 13 presents results on 2-dimensional HPE by taking different training sets related to PoseNet. PCK has shown over for defined threshold produced on Dexter. Training for S-train and R-train together yields the best results.

6.1 Lifting the Hand Pose Estimation to 3D

Hand Pose Representation

To predict the 3-dimensional hand pose from 2-dimensional joints/keypoints, the proposed canonical frame portrayal is evaluated. Compare the result with various alternatives. With the spatial resolution, all variants were trained on score maps. To avoid overfitting, we intensify the score maps through disturbing the joints location

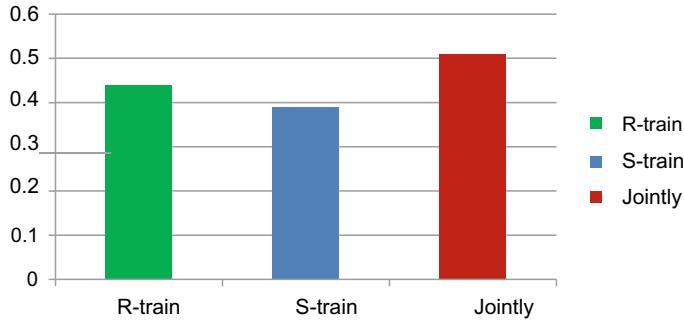


Fig. 13 Results on 2-Dimensional HPE

with Gaussian noise. Finally, scaling and translation of scoremaps are done. In Table 2, the resulting endpoint errors for every keypoint/joint are shown in Fig. 14.

Figure 15 shows, S-val Results for our proposed system compared to the approaches from [7] and [22]. PCK is shown for respective thresholds in mm. PoseNet is trained on S-train, PosePrior is trained on R-train and HandSegNet is trained on R-train.

Table 2 Results of R-train and R-val

	Direct (%)	Bottleneck (%)	NN (%)	Local (%)	Prop (%)
R-train	20.29.3	21.115	0.0–100	36.190	18.5
R-val	20.911.3	21.816	27.843	39.1109	18.6

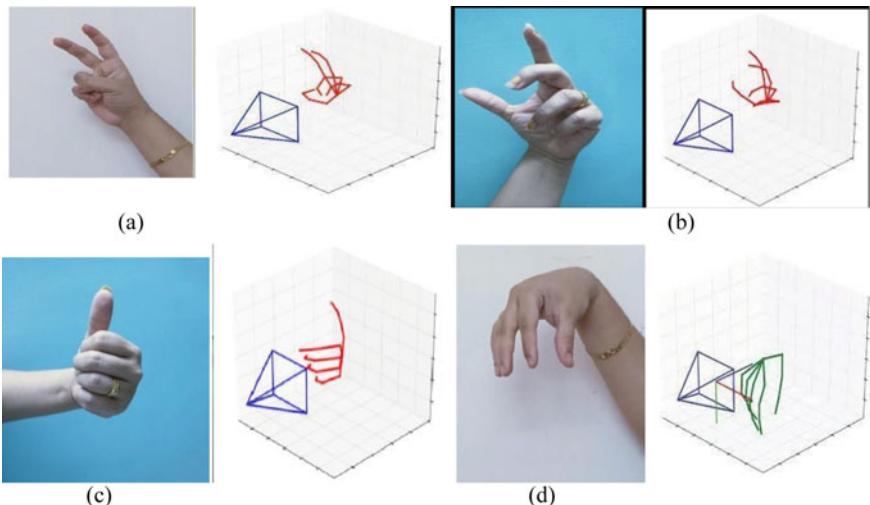


Fig. 14 Evaluative results of our proposed system

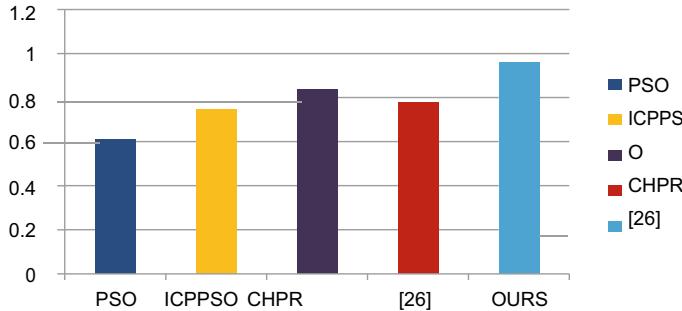


Fig. 15 Results for our proposed system

The normal approach follows the kinematic model and uses the network to predict the coordinates of the model. It estimates not only the two angles of the hand but also finds the bone length [7], the angles give the rotations related to local coordinate system related to hand bone. This technique can be implemented if the hand position is in one direction only, it cannot determine with omni pose.

At last, the NN technique relates to the identical sample related the training dataset and determines the 3D coordinates. This technique works better on the available training, it doesn't estimate the coordinates of a new data sample datasets. The objective of the different approaches is really good with similarities in errors for validation set and training set. The proposed technique works good and is used in our experiments.

7 Recognition of Sign Language

Earlier estimation of hand pose approaches build upon the depth data and these approaches cannot be applied to many datasets relative to the sign language recognition, because they come with images which are in different colors. Ultimately an ideal experiment, we have taken our proposed HPE system and later it is we have trained a classifier for the purpose of gesture recognition. The extractor is a FC 3-layer network with ReLU capabilities and functions. We have reported the results relative to the RWTH German Finger spelling Database [23]. It holds in 35 hand pose gestures exemplifying the alphabet letters. It represents the numbers from 1 to 5 and German umlauts. Dataset consists of 25 persons, recorded two times for all gestures. Almost all gestures are undeviating gestures excluded for the letters A, K, J, U and O, which are dynamic. To keep our experiment uncomplicated we conducted experiments on the 30 static gestures.

Table 3, provides error rates in percentages belongs to the RWTH German Finger spelling Database relative to the ineffective gestures. Results of Dreuw et al. [13] on the subset from [23].

Table 3 Error rates related to RWTH

Methods	Error Rate (%)
Drew et.al. [13]	35.7
Drew on subset [23]	36.76
Ours 3D	33.5

We have used one camera for sequences of short videos having resolution of 320×240 pixels. We have taken frame which is in the middle of each video as RGB image and gestures as training data. The dataset contains 1162 static images, separated by signers to make a validation set consists of 233 images and a training set consists of 928 images. We have resized each image sample to 320×320 pixels and later those resized images are trained on sampled crops. To overcome the compression artifacts, we classified 55 images belongs the training set with the keypoints/joints, used to fine-tune proposed PoseNet. Subsequently, the part related to the hand pose estimation is fixed and training is done to finally. Our proposed system records corresponding results of Drew et al., [13] to the hand Gestures, we have used for correlation purpose.

8 Conclusion

We have introduced the best system which learns the estimation of 3D HPE for an input image. We presented a large synthetic dataset which enables us to train the networks. We have presented a network acquires a 3-dimensional pose which helps us to predict the 2-dimensional key points in real-world images. The accomplishment of the system is ruthless to techniques which uses depth maps. There are still improvements are there which uses the approach of depth maps. Due to lack of explicate datasets of various real-word images with multiple pose statistics. We have compared our results to previous experiments and our proposed system gives good results. This technique can be used in sign language recognition, hand gesture recognition, AR/VR/MR systems. Our proposed system learns the detection of joints accurately thereby increasing the performance of the system. This model can be used in real-time applications also. While the performance of the network is even competitive to approaches that use depth maps, there is still much room for improvements. The performance seems mostly limited by the lack of an annotated large-scale dataset with real world images and diverse pose statistics.

References

1. Athitsos V, Sclaroff S (2003) Estimating 3d hand pose from a cluttered image. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2, pp– 432. IEEE
2. Sharp T, Wei Y, Freedman D, Kohli P, Krupka E, Fitzgibbon A, Izadi S, Keskin C, Robertson D, Taylor J, Shotton J, Kim D, Rhemann C, Leichter I, Vinnikov A (2015) Accurate, Robust, and Flexible Real-time Hand Tracking, pp 3633–3642. ACM Press
3. Oikonomidis I, Kyriazis N, Argyros A (2011) Efficient model-based 3d tracking of hand articulations using Kinect. British Machine Vision Conference (BMVC), 1, p 3
4. Tompson J, Stein M, Lecun Y, Perlin K (2014) Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. ACM Transactions on Graphics 33(5):1–10
5. Zhao R, Wang Y, Martinez A (2016) A Simple, Fast and Highly-Accurate Algorithm to Recover 3d Shape from 2d Landmarks on a Single Image. [arXiv:1609.09058](https://arxiv.org/abs/1609.09058)
6. Oberweger M, Wohlhart P, Lepetit V (2015) Training a feedback loop for hand pose estimation. In Proc of the International Conf on Computer Vision (ICCV), pp 3316–3324.
7. Zhou X, Wan Q, Zhang W, Xue X, Wei Y (2016) Modelbased deep hand pose estimation, pp 2421–2427
8. Toshev A, Szegedy C (2014) Deeppose: Human pose estimation via deep neural networks. In Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp 1653–1660
9. Tompson JJ, Jain A, LeCun Y, Bregler C (2014) Joint training of a convolutional network and a graphical model for human pose estimation. In Proc of the Conf on Neural Information Processing Systems (NIPS), pp 1799–1807
10. Sridhar S, Mueller F, Zollhofer M, Casas D, Oulasvirta A, Theobalt C (2016) Real-time joint tracking of a hand manipulating an object from rgb-d input. In Proc of the Europ Conf on Computer Vision (ECCV)
11. Jin Chang H, Garcia-Hernando G, Tang D, Kim Y-K (2016) Spatio-temporal hough forest for efficient detection–localisation–recognition of finger writing in egocentric camera. Computer Vision and Image Understanding 148:87–96
12. Chen C-H, Ramanan D (2016) 3d Human Pose Estimation= 2d Pose Estimation+ Matching. [arXiv:1612.06524](https://arxiv.org/abs/1612.06524)
13. Dreuw P, Deselaers T, Keysers D, Ney H (2006) Modeling image variability in appearance-based gesture recognition. In ECCV Workshop on Statistical Methods in Multi-Image and Video Processing, pp 7–18, Graz, Austria, May
14. Chen X, Yuille AL (2014) Articulated pose estimation by a graphical model with image dependent pairwise relations. In Proc of the Conf on Neural Information Processing Systems (NIPS), pp 1736–1744
15. Tome D, Russell C, Agapito L (2017) Lifting from the deep: Convolutional 3d pose estimation from a single image. arXiv preprint [arXiv:1701.00295](https://arxiv.org/abs/1701.00295)
16. Pavlakos G, Zhou X, Derpanis KG, Daniilidis K (2016) Coarse-to-fine volumetric prediction for single-image 3d Human Pose. arXiv preprint. [arXiv:1611.07828](https://arxiv.org/abs/1611.07828)
17. Popa A-I, Zanfir M, Sminchisescu C (2017) Deep multitask architecture for integrated 2d and 3d Human Sensing. arXiv preprint. [arXiv:1701.08985](https://arxiv.org/abs/1701.08985)
18. Moreno-Noguer, 3d Human Pose Estimation from a Single Image via Distance Matrix Regression. arXiv preprint [arXiv:1611.09010](https://arxiv.org/abs/1611.09010)
19. Bogo F, Kanazawa A, Lassner C, Gehler P, Romero J, Black MJ (2016) Keep it SMPL: Automatic estimation of 3d human pose and shape from a single image. European Conference on Computer Vision, pp 561–578. Springer
20. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems, pp 2672–2680
21. Baiek S, Kim KI, Kim T-K (2018) Augmented skeleton space transfer for depth-based hand pose estimation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), [arXiv:1805.04497v1](https://arxiv.org/abs/1805.04497v1)

22. Zhang J, Jiao J, Chen M, Qu L, Xu X, Yang Q, 3d hand pose tracking and estimation using stereo matching. arXiv preprint [arXiv:1610.07214](https://arxiv.org/abs/1610.07214)
23. RWTH German Fingerspelling database. <http://www-i6.informatik.rwth-aachen.de/~dreuw/fingerspelling.php>. Accessed 1 March 2017
24. Kingma DP, Ba J (2014) A method for stochastic optimization. CoRR, abs/1412.6980
25. Mehta D, Rhodin H, Casas D, Sotnychenko O, Xu W, Theobalt C (2016) Monocular 3d human pose estimation using transfer learning and improved CNN supervision. arXiv preprint. [arXiv:1611.09813](https://arxiv.org/abs/1611.09813)
26. Wei S-E, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp 4724–4732
27. Sarafianos N, Boteanu B, Ionescu B, Kakadiaris IA (2016) 3d Human pose estimation: A review of the literature and analysis of covariates. Comput Vis Image Underst 152:1–20
28. Caggianese G, Capece N, Erra U, Gallo L, Rinaldi M (2020) Freehand-steering locomotion techniques for immersive virtual environments: A comparative evaluation. Int J Hum Comput Interact 36:1734–1755

Climate Dependent Crop Management Through Data Modeling



Narinder Kaur and Vishal Gupta

Abstract Climate change and agriculture are interdependent on each other. Climate change has an impact (positive/negative) on agriculture, and vice versa. To improve the agricultural system, Artificial Intelligence (AI) algorithms play an important role and are highly accurate algorithms, which are emerging nowadays. Artificial Intelligence can be applied throughout the lifecycle of a plant. In this paper, we have briefly summarized research done based on application of Artificial Intelligence or machine learning techniques in crop management of the agricultural system. The motive of this paper is to discuss how agriculture benefits from machine learning technologies and how the future of agriculture can be improved based on current and archived data. It also shows how huge amount of data can be utilized in improving the agricultural system. The work discussed in this paper is focused on Crop management which in turn is categorized into four sub-categories: Crop Yield Prediction, Crop Quality Prediction, Crop Disease Detection, and Weed Detection.

Keywords Machine learning · Crop management · Crop yield prediction · Crop quality prediction · Crop disease detection · Weed detection

1 Introduction

The world's population is expected to rise from its current population of 7.3 billion to 9.7 billion by the year 2050. The "Food and Agriculture Organization (FAO), the United Nations" agency suggested to increase agricultural production by 70% by the year 2050 so as to satisfy the demand according to the expected population in 2050 [1]. The most difficult task is to increase the crops production with minimum

N. Kaur (✉)

Department of Computer Science Engineering, AIACT&R, GGSIP University, Delhi, India
e-mail: narinderkaur@mait.ac.in

V. Gupta

Department of Computer Science and Engineering, NSUT East Campus (Erstwhile AIACT&R, GGSIP University), Delhi, India
e-mail: vishalgupta@aiactr.ac.in

use of available resources like land, water, fertilizers, etc. Several initiatives have been taken in order to fulfill these increasing demands since the 1990s. Nowadays, “Cloud Computing” [2], “Remote Sensing” [3], and “Internet of Things” (IoT) [4] play an important role in agricultural enhancements which leads to smart farming. Data science helps farmers and agricultural professionals by connecting interrelated devices and the Internet. It is helpful in exchanging and sharing data easily. Data analytics can be applied to archived/current data to obtain important decision-making information which farmers and agricultural professionals can use so as to improve and enhance the agricultural system. Data analytics can be applied during the plant life cycle; from planning, seed planting, germination of seed, seedling, vegetative, flowering, harvesting, and all the way to its marketability.

Agriculture plays a crucial role in the economy of developing countries and also in the field of global economy. Climate change and the agricultural system are inter-dependent to each other and take place at a global level. The agriculture system gets affected by the Global warming, which includes effects of changes in temperature, precipitation, and climatic disasters (Tornado, floods, etc.) on the agriculture; alterations in crop pests and crop diseases; changes in atmospheric level of CO₂ and ozone concentrations at ground level; changes in the nutritional value/quality of foods. Climate change affects agriculture directly or indirectly and the effects are non-uniformly distributed throughout the world.

Change in climate has negative as well as positive effects on the agricultural system (Fig. 1).

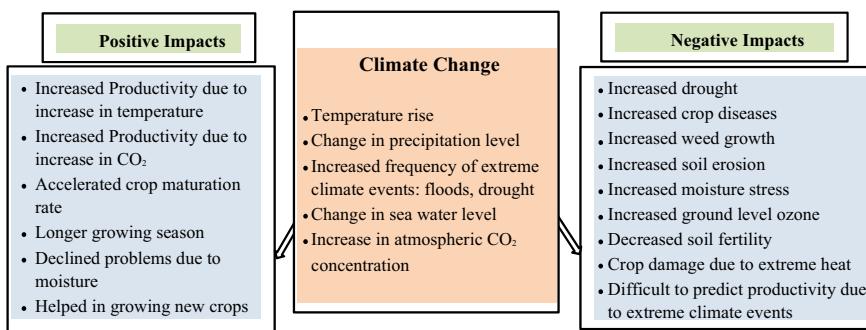


Fig. 1 Positive and negative impacts of climate change on agriculture

Key areas in Agriculture Data Analytics are:

- Livestock management: Livestock Production.
- Species management: Species Breeding, Species Recognition.
- Field conditions management: Soil management, Water Management.
- Crop management: Crop Yield Prediction, Crop Quality Prediction, Crop Disease Detection, Weed Detection.

Tables 1 and 2 indicate abbreviations used in the paper:

2 Introduction to Machine Learning

Machine Learning (ML) is a subset of AI. It deals with a learning process that aims to “learn from experience” to perform a specific task. ML describes everything in terms of a set of attributes called characteristic features. Attributes can be nominal (without order), ordinal (ordered), binary (0/1), numeric, etc. ML model learns from archived data, builds the new model based on the input and expected output, and predicts output based on new data (Fig. 2).

ML is mainly divided into three types: supervised, unsupervised, and reinforcement learning. Supervised learning consists of a training set with labeled data (inputs and their outputs are known), it learns mapping of inputs and outputs, e.g., classification, regression models. Unsupervised learning consists of a training set without labeled data; it discovers patterns, structures, etc. from the unlabeled data, e.g., clustering, dimensionality reduction models. Reinforcement learning consists of training data in the form of rewards and punishment; it is based on performing action to maximize reward in a given situation. Figure 3 depicts the different ML algorithms, along with the examples:

3 Literature Survey

Crop management is one of the most important tasks in the agricultural system. A good crop management leads to high crop production, high quality crop and disease/pest free crop. The crop management is mainly classified into four sub-categories:

- Crop Yield Prediction
- Crop Quality Prediction
- Crop Disease Detection
- Weed Detection.

Table 1 Abbreviations for machine learning algorithms/models

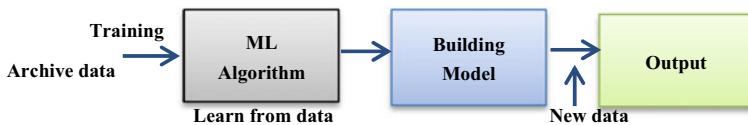
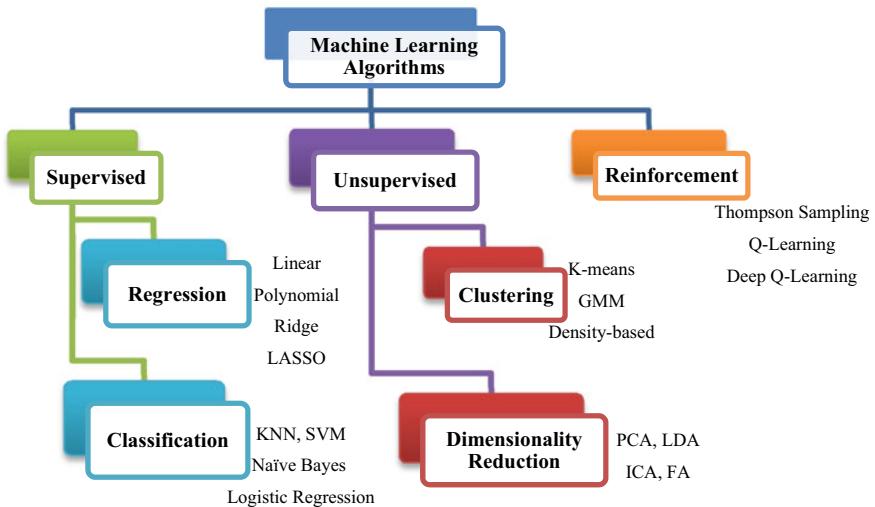
Abbreviation	Algorithm name
ANN	Artificial neural networks
BNN	Bayesian neural networks
BP	Back propagation
CNN	Convolutional neural networks
DCNN	Deep convolutional neural network
DT	Decision trees
GBM	Gradient boosting model
GMDH-logistic	Group method of data handling-logistic
GMM	Gaussian mixture model
ICA	Independent component analysis
kNN	k-nearest neighbors
LASSO regression	Least absolute shrinkage and selection operator regression
LDA	Linear discriminant analysis
LR	Linear regression
LSTM NN	Long-short term memory (LSTM) neural networks
MARS	Multivariate adaptive regression splines
MLP	Multi-layer perceptron
MLR	Multiple linear regression
MOB	Model-based recursive partitioning
PCA	Principal component analysis
PDM	Panel data model
PSO	Particle swarm optimization
QDA	Quadratic discriminant analysis
RCM	Random coefficient model
RF	Random forest
RF + AR	Random forest additive regression
RF + BAG	Random forest bagging
STSM	Separate time series model
SVM, SVMI, SVMr	Support vector machine l-linear, r-radial

3.1 Crop Yield Prediction

Crop Yield Prediction is an important task of “Precision Agriculture (PA)”. PA is a real-time management of agricultural activities using satellite technology. PA aims in maximizing the yield prediction with minimum resources usage. Schwalbert et al. [5] observed Soybean crop yield based on temperature, precipitation and concluded that the LSTM neural networks performed better than other two models. Obsie et al.

Table 2 General abbreviations

Abbreviation	Full form
AVHRR	Advanced very high resolution radiometer
CBB	Cassava bacterial blight
CBSD	Cassava brown streak virus disease
CGM	Cassava green mite
CMB	Cassava mosaic disease
MODIS	Moderate-resolution imaging spectro-radiometer
NDVI	Normalized difference vegetation index
SSC	Soluble solids concentration

**Fig. 2** Block diagram of machine learning algorithm**Fig. 3** Machine learning algorithms

[6] observed Blueberry yield based on daily air temperature and daily precipitation and concluded that XGBoost model performed better than other two models. Kamir et al. [7] observed wheat crop yield based on NDVI time series and climate time series using nine different AI models and concluded that SVMr model performed best out of nine models. Feng et al. [8] observed Wheat crop yield based on climatic features and concluded that RF is better than MLR. Cao et al. [9] observed Rice crop

yield and concluded that the LSTM model is better than RF, LASSO models. Ahmad et al. [10] observed Maize yield based on NDVI time series and eight different AI models and concluded that SVMr is the best model with accuracy 97%. Nevavuori et al. [11] observed Wheat varieties, Barley varieties yield based on thermal time, sowing date, imaging date using Deep CNN and concluded that it performs better with RGB images than NDVI images.

Gopal et al. [12] observed Maize and Wheat crop yield based on Tmax, Tmin, Tavg, potential evapotranspiration, global radiation, etc. and proposed hybrid MLR-ANN model. Khosls et al. [13] observed Bajra, Rice, Ragi, Maize crop yield based on rainfall data and proposed “Modular Artificial Neural Network—Support Vector Regression model (MANN-SVR)”. Merizig et al. [14] observed Palm tree and Date fruits yield and proposed an IoT-based model for enhancing agricultural Production by forecasting the production of fruits based on historical data. Khanal et al. [15] observed Corn plant yield based on soil-related data: Soil Organic Matter, Potassium, Magnesium, pH, etc. using seven ML models and concluded that linear regression model outperformed out of all models. Shah et al. [16] observed Corn plant yield based on humidity, yield, temperature and rainfall using MLR, RF, SVM models and concluded that SVM is better than other two models. Su et al. [17] observed Rice crop yield based on static and dynamic variables and proposed “SVM-Based Open Crop Model (SBOCM)”. Cheng et al. [18] observed Apple yield and proposed “Back Propagation Neural Networks (BPNN)” to predict early yield.

Johnson et al. [19] observed Barley, Canola, spring Wheat yield using BNN, MOB, and MLR and concluded that BNN and MOB are slightly better than MLR. Gornott et al. [20] observed Winter Wheat, Silage Wheat yield prediction based on climatic data using STSM, PDM, and RCM models and concluded that SSTM is better than other two models. Jayakumar et al. [21] observed Coffee Arabica, Coffee canephora based on Rainfall, Tmax, Tmin, mean relative humidity, etc. and proposed “Statistical forecasting model between climate and yield of Coffee arabica and Coffee canephora” for yield prediction. MATSUMURA et al. [22] observed Maize plant yield based on temperature and rainfall data and concluded that ANN model performed better than MLR model for the given dataset. Ahamed et al. [23] observed Rice varieties, Potato, Wheat using Clustering, k-means clustering, Classification: Linear Regression, k-NN, ANN models and concluded that ANN provides better prediction for some of the crops, which have more missing values than others, for example Wheat, Potato, Rice-Aus. Linear regression provides better performance for Rice-Boro, Rice-Amon. Table 3 outlines the above-mentioned papers for crop yield prediction sub-category.

3.2 *Crop Quality Prediction*

Crop management also aims for high quality crops. Crop Quality means health-related attributes like nutrients, phytochemical composition, safety, etc. related to a

Table 3 Crop management: crop yield prediction

Reference year publisher	Author name	Crop	Observed features/Input	Model/s/algorithms/functionality	Output/conclusion
[5] 2020 Elsevier	Schwalbert, R. A. et al	Soybean	Temperature and precipitation information	Multivariate ordinary least squares linear regression, random forest and LSTM neural networks	Concluded that LSTM NN performed better than other two models
[6] 2020 Elsevier	Obsie, E. Y. et al	Blueberry	Blueberry clone size, honeybee density, Bumblebee density, andrena density, osmia density, Daily air temperature, daily precipitation	MLR, boosted DT, RF, XGBoost	Concluded that XGBoost model performed better than other two models
[7] 2020 Elsevier	Kamir, E. et al	Wheat	NDVI time series (35 predictor variables), Climate time series (46 predictor variables)	RF, XGBoost, Cubist (CUB), MLP, SVM, SVMr, Gaussian Process, kNN, MARS	Concluded that SVMr model performed best out of nine models
[8] 2020 Elsevier	Feng, P. et al	Wheat	Daily climate data for the 29 sites like solar radiation, precipitation, and minimum and maximum air temperature NDVI time series	RF, MLR	Concluded that the RF model is better than MLR and can provide better forecast at earlier growth also

(continued)

Table 3 (continued)

Reference year publisher	Author name	Crop	Observed features/Input	Model/s/algorithms/functionality	Output/conclusion
[9] 2020 Elsevier	Cao, J. et al	Rice	Country yield, planting area, satellite vegetation Indexes, climate data, soil properties, irrigation ratio	RF, LASSO regression, LSTM	Concluded that LSTM model is better than other models
[10] 2020 Elsevier	Ahmad, I. et al	Maize	Maize yield, NDVI, land surface temperature	LDA, QDA, k-NN, SVM, DT, boosting DT, RF	Concluded that SVM is the best model with accuracy 97%
[11] 2019 Elsevier	Nevavuori, P. et al	Wheat varieties, barley varieties	Thermal time, sowing date, imaging date	Deep CNN	Concluded that CNN architecture performs better with RGB images than NDVI images
[12] 2019 Elsevier	Gopal, M. et al	Maize, wheat	Tmax, tmin, tavg, potential evapotranspiration, global radiation etc	SVR, kNN, RF, MLR, ANN, hybrid MLR-ANN	Hybrid MLR-ANN model is proposed
[13] 2019 Springer	Khosla E. et al	Bajra, rice, ragi, maize	Rainfall data of coastal Andhra Pradesh, bajra, rice, ragi, maize crop yield data	MANN-SVR	Proposed modular artificial neural network—“support vector regression model (MANN-SVR)” for rainfall and different crops yield prediction in Visakhapatnam

(continued)

Table 3 (continued)

Reference year publisher	Author name	Crop	Observed features/Input	Model/s/algorithms/functionality	Output/conclusion
[14] 2019, Springer Elsevier	Merizig A. et al	Palm trees and date fruits	Drone collects the number of palm trees in the forest and date fruits' information	Clustering using k-means method	Proposed an IoT based model for enhancing the agricultural production by forecasting the production of fruits based on historical data
[15] 2018 Springer	Khanal, S. et al	Corn	Soil data: soil organic matter, potassium, magnesium, pH etc., corn yield	Linear regression, RF, NN, SVM, SVMi, GBM, CUB	Concluded Linear regression model outperformed out of all models
[16] 2018 Springer	Shah, A. et al	Corn	humidity, yield, temperature and rainfall	MLR, RF, SVM	Concluded that SVM mode better than other models
[17] 2017, Saudi Journal of Biological Sciences	Su, Y. et al	Rice	Static variables: soil data: nitrogen, phosphorus, potassium, pH etc dynamic variables: daily air pressure, daily tavg, daily relative humidity, etc	SVM	Proposed “SVM-based open crop model (SBOCM)”
[18] 2017 MDPI	Cheng, H. et al	Apple	Image analysis and tree canopy features	ANN	Proposed “back propagation neural networks (BPNN)” to predict early yield

(continued)

Table 3 (continued)

Reference year publisher	Author name	Crop	Observed features/Input	Model/s/algorithms/functionality	Output/conclusion
[19] 2016 Elsevier	Johnson M. D. et al	Barley canola, spring wheat	Crop yield data, remotely sensed vegetation indices from the AVHRR and the MODIS sensor	BNN, MOB, MLR	Concluded that BNN and MOB are slightly better than MLR
[20] 2016 Elsevier	Gornott, C. et al	Winter wheat, silage wheat	Winter wheat and silage wheat yield data, climate data of Germany	STSTM, PDM, and RCM	Concluded that STSTM is better than other two models
[21] 2016 Springer	Jayakumar, M. et al	Coffee arabica, coffee canephora	Rainfall, tmax, tmin, and mean relative humidity coffee yield data	Statistical regression model	Proposed “statistical forecasting model between climate and yield of coffee arabica and coffee canephora” for yield prediction
[22] 2015, Journal of Agricultural Science	MATSUMURA K. et al	Maize	Annual Maize yield, monthly temperature and precipitation data	MLR, ANN	Concluded that ANN model performed better than MLR model for the given dataset

(continued)

Table 3 (continued)

Reference year publisher	Author name	Crop	Observed features/Input	Model/s/algorithms/functionality	Output/conclusion
[23]	Ahamed, S. et al	Rice varieties, potato, wheat	Soil data: Min pH, Max pH etc., irrigated area, cultivated area	Clustering, k-means clustering, classification: linear regression, K-NN,A. NN	Concluded that “ANN provides better prediction for some of the crops, which have more missing values than others, for example wheat, potato, Rice-Aus. Linear regression provides better performance for rice-boro, rice-anon.”

crop. Quality parameters refer to concentrations and existence of secondary metabolites, bioactivity, phytonutrients, and organoleptic features such as color, size, shape, texture, etc. as well as shelf life. Calcium, antioxidants, iron, magnesium, etc. are few nutrients that get affected by the change in climate. Nistor et al. [24] observed Grape quality and concluded that increase in temperature and rainfall levels, increases sugar content in grapes. Qu et al. [25] observed Apple quality in two different areas of China and concluded that an increase in temperature with decrease in sunlight enhances the nutrient based apple's feature in one area. Whereas in another area, there was decrease in fruit quality as temperature goes beyond the optimum range, sunlight hours were lesser than the optimum range and air moisture levels were not in between the optimum range. Sugiura et al. [26] observed Apple quality and observed that temperature is increasing with variable change in precipitation and solar radiation. It is also concluded that there is a consistent increase in SSC over the past 40 years.

Cozzolino et al. [27] observed Grape quality and concluded that increase in temperature in general increases the content of anthocyanin in grapes. Fukuoka et al. [28] observed Watermelon quality based on temperature around the watermelon during the second half development phase of watermelon and concluded that increase in temperature benefits the fruits by increasing the size but it also decreases glucose and fructose concentrations in the fruits. Rouphael et al. [29] observed Watermelon quality and concluded that with decrease in irrigation, yield decreased linearly. No relation was found between irrigation and SSC or Phosphorous/Calcium contents in fruits, but it affected the content of Potassium and Magnesium in fruits. The content of Potassium was highest at irrigation rate = 0.75. They also concluded that Magnesium content is inversely proportional to irrigation level. Idso et al. [30] observed Orange quality and concluded that an increase in atmospheric Carbon dioxide concentration generally increased the fruit production, orange's weight, and increased the content of vitamin C of the orange juice. Table 4 outlines the above-mentioned papers for crop quality sub-category.

3.3 Crop Disease Detection

Disease detection is another important task in the agricultural system. Usage of pesticides is the commonly used solution to pests and crop disease management. The solution is expensive and not environment friendly. There is a big negative impact of pesticides on groundwater quality, wildlife, and ecosystem. Detecting crop disease at an early stage is quite helpful so as to protect the crop from being useless. The main aim of crop disease management is to minimize the crop disease. Sambasivam et al. [31] observed Cassava leaves images for disease detection and found that the dataset size is small and is biased toward CMD and CBB classes. Researchers used the “SMOTE (Synthetic Minority Over-sampling Technique)” model to overcome the biasing problem and attained accuracy = 93%. Karadağ et al. [32] observed Pepper plant for disease detection using ANN, NB, k-NN and concluded that k-NN

Table 4 Crop management: crop quality prediction

Reference year publisher	Author name	Crop	Observed features/ input	Models/algorithms/functionality	Output/conclusion
[24] 2018, South African Journal of Enology and Viticulture	Nistor, E. et al	Grape	Temperature, rainfall	Grapes quality parameter (sugar content) is observed based on temperature and rainfall	Researchers concluded that increase in temperature and rainfall levels increases sugar content in grapes
[25] 2016 MDPI	Qu, Z. et al	Apple	Temperature, humidity, and solar radiation	Apple quality parameters (vitamin C content, sugar-acid ratio, anthocyanin concentration) are observed based on the climatic variables and components of the apple fruit	Concluded that an increase in temperature with decrease in sunlight enhances the nutrient based apple's feature. Whereas in another area, there was decrease in fruit quality as temperature goes beyond the optimum range, sunlight hours were lesser than the optimum range and air moisture levels were not in between the optimum range

(continued)

Table 4 (continued)

Reference year publisher	Author name	Crop	Observed features/ input	Models/algorithms/functionality	Output/conclusion
[26] 2013 Scientific Reports	Sugiura, T. et al	Apple	Temperature, precipitation, solar radiation	Apple fruit features like firmness, acid concentration, soluble-solid concentration, water core rating are observed based on the climatic features	Two locations in Japan are considered for apple quality and observed that temperature is increasing with variable change in precipitation and solar radiation. It is concluded that there is consistent increase in SSC over the past 40 years
[27] 2010 Elsevier	Cozzolino, D. et al	Grapes	Temperature, rainfall, carbon dioxide level	Grapes quality parameter (anthocyanin) is observed based on the Temperature, rainfall and carbon dioxide level	Several grapes growing regions of Australia are considered and concluded that increase in temperature in general increases content of anthocyanin in grapes
[28] 2009, JSHS—Japan society of horticultural science	Fukuoka, N. et al	Watermelon	Temperature	Watermelon quality parameter (sugar) is observed based on the temperature around the fruit	Concluded that increase in temperature benefits the fruits by increasing the size but it also decreases glucose and fructose concentrations in the fruits

(continued)

Table 4 (continued)

Reference year publisher	Author name	Crop	Observed features/ input	Models/algorithms/functionality	Output/conclusion
[29] 2008 HortScience	Rouphael, Y. et al	Watermelon	Water deficit	Watermelons were optimally given water for 3 weeks after planting. After 3 weeks, irrigation was maintained optimally or decreased to 0.75 or 0.5 of the optimal level	Researchers concluded that with Decrease in irrigation, yield decreased linearly. No relation was found between irrigation and “Soluble Solid Concentration” or Phosphorous/Calcium contents in fruits, but it affected the content of Potassium and Magnesium in fruits. The content of Potassium was highest at irrigation rate = 0.75. They also concluded that Magnesium content is inversely proportional to irrigation level
[30] 2002 Elsevier	Idso, S. B. et al	Orange	Atmospheric CO ₂ level	Orange quality parameter (Vitamin C) is observed based on atmospheric carbon dioxide level	Eight years of data is considered and concluded that an increase in atmospheric Carbon dioxide concentration generally increased the fruits production, orange's weight, and increased the content of vitamin C of the orange juice

performed the best. Jun-De et al. [33] observed Cucumber leaves for disease detection and proposed GMDH-logistic model with accuracy = 86.67%. Agarwal et al. [34] observed Tomato leaves for disease detection and proposed CNN-based model with average accuracy = 91.2%. Karthik et al. [35] observed Tomato leaf images for disease detection and proposed two architectures: First architecture applies residual learning to learn important features for classification; Second architecture applies attention mechanism on top of the residual deep network and proposed model with accuracy = 98%. Geetharamani et al. [36] observed apple, grape, cherry, potato, etc. leaves for disease detection and proposed deep CNN model with average accuracy = 96.46%.

Ferentinos et al. [37] observed leaf images of Apple, Banana, Blueberry, Cabbage, Maize, etc. using five CNN-based models: AlexNet, AlexNetOWTBn, GoogleNet, Overfeat, VGG and concluded that VGG model performed best with accuracy = 99.53%. Maniyath et al. [38] observed Papaya leaves for disease detection using Logistic regression, SVM, k-NN, Cluster and Regression Tree (CART), RF, Naïve Bayes models and concluded that RF model performed best with accuracy 70.14 percent. Lu et al. [39] observed Rice leaves for disease detection and Proposed CNN-based model. It also compared the proposed model with BP, SVM, and PSO. Proposed model performed best with accuracy = 95%. Ebrahimi et al. [40] observed Strawberry flowers for disease detection and applied SVM model. Model successfully identified the Thysanoptera (disease). Bhange et al. [41] observed Pomegranate fruits for disease detection and proposed a model based on K-means clustering and SVM classification with accuracy = 82%. Table 5 outlines the above-mentioned papers for disease detection sub-category.

3.4 Weed Detection

Weed is an unwanted plant which grows automatically with the main crop. Weed plants hinder the growth of the main crop by consuming nutrients which were supposed to be for the main crop. Weed management can be done properly only if weed detection is accurately performed. It aims at detecting and removing or

Table 5 Crop management: crop disease detection

Reference	Author Name	Crop	Observed Features/ Input	Models/Algorithms /Functionality	Output/Conclusion
[31]	Sambasivan, G. et al	Cassava	Plant images	CNN applied for four diseases detection: CMD, CGM, CBB, CRSD	Researchers found that the dataset size is small and is biased toward CMD and CBB classes. Researchers used the “SMOTE (Synthetic Minority Over-sampling Technique)” model to overcome the problem and attained accuracy = 93%
[32]	Karadağ, K. et al	Pepper	Plant grown under the climate room and observed directly	ANN, Naïve Bayes, k-NN	k-NN model performed the best with success rate = 100% in classifying only diseased and healthy plants. k-NN also achieved success rate = 99% in detecting four classes (diseased and healthy plant). Researchers identified the fusarium disease in pepper leaves
[33]	Jun-De C. et al	Cucumber	Leaf images	SVM, PCA-SVM, CNN, genetic ANN, PCA-ANN, GMDH-logistic	Proposed GMDH-Logistic model for plant disease detection (average accuracy = 86.67%)
[34]	Agarwal, M. et al	Tomato	Leaf images	CNN with 3 convolution and 3 max pooling layers followed by 2 fully connected layers	Proposed CNN-based model with average accuracy = 91.2%

(continued)

Table 5 (continued)

Reference Year Publisher	Author Name	Crop	Observed Features/ Input	Models/Algorithms /Functionality	Output/Conclusion
[35] 2019 Elsevier	Karthik, R. et al	Tomato	Leaf images	Residual CNN model, attention embedded residual CNN model	Researchers proposed two architectures: First architecture applies residual learning to learn important features for classification, Second architecture applies attention mechanism on top of the residual deep network. Proposed model performs with accuracy = 98%
[36] 2019 Elsevier	Geetharamani, G. et al	Apple, grape, cherry, potato, etc.	Leaf images	Deep CNN	Proposed Deep CNN model with average accuracy = 96.46%
[37] 2018 Elsevier	Ferentinos, K. P. et al	Apple, banana, blueberry, cabbage, maize, etc.	Leaf images	Five different CNN models: (i) AlexNet (ii) AlexNetOWTBN (iii) GoogleNet (iv) Overfeat (v) VGG(Visual Geometry Group)	VGG model performed best with accuracy = 99.53%
[38] 2018 Springer	Maniyath, S.R. et al	Papaya	Leaf images	Logistic regression, SVM, k-NN, CART, RF, Naïve Bayes	Researchers concluded that RF model performed best with accuracy = 70.14%

(continued)

Table 5 (continued)

Reference Year Publisher	Author Name	Crop	Observed Features/ Input	Models/Algorithms /Functionality	Output/Conclusion
[39] 2017, Elsevier	Lu, Y. et al	Rice	Plant images	Deep CNN, BP, SVM, PSO	Proposed CNN-based model and compared with BP, SVM, and PSO. Proposed model performed best with accuracy = 95%
[40] 2017, Elsevier	Ebrahimi, M. A. et al	Strawberry	Flower images	SVM	SVM model successfully identified the thysanoptera
[41] 2015, Elsevier	Bhang, M. et al	Pomegranate	Fruit images	K-means clustering,SVM classification	Proposed model performed well with accuracy = 82%

reducing weed. Espejo-Garcia et al. [42] observed Tomato and Cotton crop images for weed detection and proposed fine-tuned DenseNet and SVM models with accuracy = 99.29%. Yu et al. [43] observed Bermuda grass images for weed detection and applied three DCNN-based models: DetectNet, GoogLeNet, VGGNet and concluded that DetectNet model performed best with F1 scores greater than 0.99. Bakhshipour et al. [44] observed Sugar beet crop for weed detection using SVM and ANN models and concluded that ANN and SVM detected weeds in the main crop with accuracies 92.5% and 93.33%. They also concluded that ANN and SVM detected sugar beet crop with accuracies 93.33 and 96.67%, i.e., SVM is better in weed detection whereas ANN is better in main crop detection.

Dos Santos Ferreira et al. [45] observed Soybean crops using CNN, SVM, AdaBoost, and Random Forests and concluded CNN performed best with accuracy = 98% to detect broadleaf and grass weeds. Obtained Accuracy > 99% when considered 15 thousand analyzed images. Software named “Pynovisão” is built that uses “Simple Linear Iterative Clustering (SLIC) Superpixels algorithm”. Fletcher, R. S. et al. [46] observed Soybean leaf reflectance measurements and applied RF machine learning to detect two pigweeds. Cheng et al. [47] observed Rice crop for weed detection and Proposed a framework for “weed and rice identification based on image feature analysis and machine learning techniques” using Harris corner detection algorithm for feature extraction of weeds and rice crop, DT, SVM, NN models for classification after extraction, “Density-based spatial clustering of applications with noise (DBSCAN)” algorithm used for clustering to separate the rice crop from four different types of the weeds. Ahmed et al. [48] observed Chili images for weed detection and proposed a SVM-based classification model with accuracy greater than 97%. Rumpf et al. [49] observed Maize, winter Wheat, Sugar beet for weed detection and proposed SVM-based three steps classification model with accuracy = 97.7%. Table 6 outlines the above-mentioned papers for weed detection sub-category.

4 Challenges to Enhance the Agricultural System

Challenges to overcome the negative impact of climate on agriculture are as follows:

- The most important challenge is to increase in food production according to future demand based on the rate of increase in population, i.e., to meet the future demand of food production or “Food Security”.
- Properly usage of limited natural resources, i.e., to face “Ecosystem Degradation”.
- There is a huge amount of data which is available nowadays but extracting the most promising or high quality data is also an important challenging area.
- To reduce the negative impact of “Global Warming”.
- Frequency of extreme weather events like floods, drought is increasing which results in a huge amount of crop damage. Precisely prediction of these extreme weather events is a big challenge
- To predict weather patterns which are currently difficult to predict is also a very important task.
- To mitigate the effects of unpredictable weather patterns so as to avoid or reduce the sudden damage of crops.
- Climatic change affects the agricultural system directly or indirectly and adaptation is very much important so as to improve the agricultural system. Therefore, adaptation to the change in climate is required without affecting agriculture.
- Climate change impacts agriculture positively or negatively. It is important to identify the climatic features which are impacting agriculture negatively and data models are required to determine the level of impact of these features.
- It is observed that some climatic features are beneficial for one crop but impacts adversely on another crop. So it is important to identify these types of features so as to grow the crops accordingly.
- There are many machine learning models available for enhancement in the agricultural system. Enhancement of these models is another challenging area.

Table 6 Crop management: weed detection

Reference year, publisher	Author name	Crop	Observed features/ input	Models/algorithms/functionality	Output/conclusion
[42] 2020 Elsevier	Espejo-Garciaa B, et al	Tomato, cotton	Weeds images, crop images	Fine-tuned densenet and support vector machine	Proposed model performed well with accuracy = 99.29%
[43] 2019 Elsevier	Yu, J. et al	Bermudagrass	Weeds images, bermudagrass images	Three DCNN architectures including (i) DetectNet (ii) GoogLeNet (iii) VGGNet	DetectNet model performed best with F1 scores > 0.99
[44] 2018 Elsevier	Bakhshiipour, A. et al	Sugar beet	5 different weeds and Sugar beet images	SVM, ANN	Concluded that ANN and SVM detected weeds in the main crop with accuracies 92.5 and 93.33%. Also concluded that sugar beet is detected with accuracies 93.33% and 96.67% by ANN and SVM model. SVM is better in weed detection whereas ANN is better in main crop detection

(continued)

Table 6 (continued)

Reference year, publisher	Author name	Crop	Observed features/ input	Models/algorithms/functionality	Output/conclusion
[45] 2017 Elsevier	Dos Santos Ferreira, A. et al	Soybean	Soil, soybean, broadleaf and grass weeds images	CNN, SVM, AdaBoost and random forests	Concluded that CNN performed best with accuracy = 98% to detect broadleaf and grass weeds. Accuracy > 99% when considered 15 thousand analyzed images. A Software named “Pynovisão” is built that uses “Simple Linear Iterative Clustering (SLIC) Superpixels algorithm”
[46] 2016 Elsevier	Fletcher, R. S. et al	Soybean	Leaf reflectance measurements	RF machine learner and leaf multispectral reflectance data	Two pigweeds (Palmer amaranth and redroot) in soybean are identified using “RF machine learner and leaf multispectral reflectance data tools”. The RF model identifies these two pigweeds from three varieties of soybean

(continued)

Table 6 (continued)

Reference year, publisher	Author name	Crop	Observed features/ input	Models/algorithms/functionality	Output/conclusion
[47] 2015 Springer	Cheng, B. et al	Rice	Rice and four different weeds images	Harris corner detection algorithm to extract rice crop and weeds parameters, DT, SVM, NN models for classification and “Density-based spatial clustering of applications with noise (DBSCAN)” algorithm for clustering	Proposed a framework for “weed and rice” identification based on image feature analysis and machine learning techniques”
[48] 2012 Elsevier	Ahmed F. et al	Chili	Weeds images, chili images	SVM	Researchers proposed a SVM based classification model
[49] 2012 Elsevier	Rumpf, T. et al	Maize, winter wheat, sugar beet	Crop images, weed images	SVM-Weighting-T for linear classification RELIEF-F algorithm for non-linear classification tasks	Proposed SVM based model with accuracy > 97%

5 Solutions and Proposed Methodology

There are mainly two solutions to reduce the impact of climate on the agricultural system: “Mitigation” and “Adaptation” (Fig. 4). “Mitigation” deals with all the processes or methods used to reduce the climatic change which in turn helps in maintaining and enhancing the agricultural system. “Adaptation” is another solution which deals with all the processes or methods used by agriculture so as to adapt to the climatic change without harming or affecting the current agricultural system. This implies either reducing the climatic change or/and adapting to the climatic change.

Figure 5 depicts the proposed methodology to perform “Mitigation” or “Adaptation”.

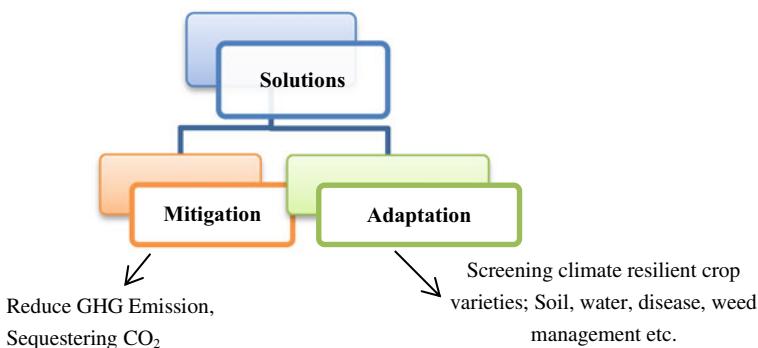


Fig. 4 Solutions to enhance agricultural system

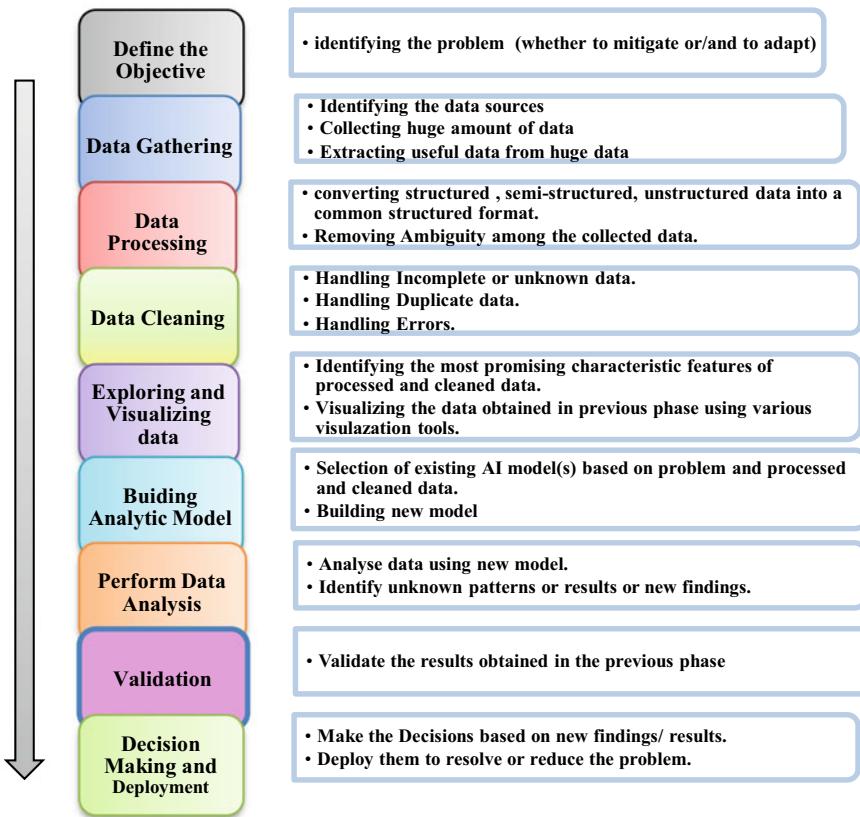


Fig. 5 Proposed methodology

6 Conclusion and Future Scope

Agriculture plays an important role in our life. We all rely on agriculture for our survival. Agricultural production improvement is a major concern to cope up with the demand supply chain. Scarcity of natural resources, Climate change, Global warming are the key factors on which agriculture depends. Data Analytics play a significant role in overcoming the adverse effects of climate change on agriculture. Crop management is one of the key areas in the field of agriculture which is studied and summarized in this paper. Crop management is mainly categorized into four sub-categories: Crop Yield Prediction, Crop Quality Prediction, Crop Disease Detection, and Weed Detection. AI/ML is the most promising decision-making tool which can be applied on the available data so as to make decisions for the enhancement of the agricultural system. It is also observed that selection of ML models depends on the available dataset. Crop production quantity and quality both can be managed by AI models.

“Ecosystem degradation”, “Global Warming”, “supply demand chain”, and “food insecurity” are the most crucial and challenging areas which we have to handle at present as well as in the future also. It is observed that lots of data models are already created for many crops but not for every crop. There are a huge number of crops available which are still unexplored due to the lack of data or due to lack of data model to handle the existing data or due to some other reason(s). Therefore, AI models are required to be applied on those crops. Many existing models lack in accuracy while performing predictions, and are required to be improved. Large amount of data is available but identifying high quality data is required for better decision making. Non-availability of data from the remote areas is another issue that needs to be addressed for enhancement in the agricultural system. Therefore, in future, lots of work has to be done to face all the challenges mentioned above and to face the upcoming challenges also.

References

1. The Future of Agriculture. <https://www.economist.com/node/21698612/help/accessibilitypolicy>, (2016)
2. Hashem I et al (2015) The rise of “big data” on cloud computing: review and open research issues. *Inform Syst* 47:98–115
3. Bastiaanssen W, Molden D, Makin I (2000) Remote sensing for irrigated agriculture: examples from research and possible applications. *Agric Water Manage* 46(2):137–155
4. Weber RH, Weber R (2017) Internet of things. Springer, New York, NY. Wolfert S, Ge L, Verdouw C, Bogaardt M
5. Schwalbert RA, Amado T, Corassa G, Pott LP, Prasad PVV, Ciampitti IA (2020) Satellite-based soybean yield forecast: Integrating machine learning and weather data for improving crop yield prediction in southern Brazil. *Agric For Meteorol* 284(107886):1–9
6. Obsie EY, Qu H, Drummond F (2020) Wild blueberry yield prediction using a combination of computer simulation and machine learning algorithms. *Comput Electron Agric* 178(105778):1–11
7. Kamir E, Waldner F, Hochman Z (2020) Estimating wheat yields in Australia using climate records, satellite image time series and machine learning methods. *ISPRS J Photogramm Remote Sens* 160:124–135
8. Feng P, Wang B, Liu DL, Waters C, Xiao D, Shi L, Yu Q (2020) Dynamic wheat yield forecasts are improved by a hybrid approach using a biophysical model and machine learning technique. *Agricult Forest Meteorol* 285–286
9. Cao J, Zhang Z, Tao F, Zhang L, Luo Y, Zhang J, Xie J et al (2021) Integrating multi-source data for rice yield prediction across china using machine learning and deep learning approaches. *Agricult Forest Meteorol* 297(108275):1–15
10. Ahmad I, Singh A, Fahad M, Waqas MM (2020) Remote sensing-based framework to predict and assess the interannual variability of maize yields in Pakistan using Landsat imagery. *Comput Electron Agricult* 178(105732):1–9
11. Neuvuori P, Narra N, Lipping T (2019) Crop yield prediction with deep convolutional neural networks. *Comput Electron Agric* 163(104859):1–9
12. Maya Gopal PS, Bhargavi R (2019) A novel approach for efficient crop yield prediction. *Comput Electron Agric* 165(104968):1–9
13. Khosla E, Dharavath R, Priya R (2019) Crop yield prediction using aggregated rainfall based modular artificial neural networks and support vector regression. *Environ Develop Sustain*

14. Merizig A, Saouli H, Zouai M, Kazar O (2019) An intelligent approach for enhancing the agricultural production in and areas using iot technology. *Multiple Myeloma* 22–36
15. Khanal S, Fulton J, Klopfenstein A, Douridas N, Shearer S (2018) Integration of high resolution remotely sensed data and machine learning techniques for spatial prediction of soil properties and corn yield. *Comput Electron Agric* 153:213–225
16. Shah A, Dubey A, Hemnani V, Gala D, Kalbande DR (2018) Smart farming system: crop yield prediction using regression techniques. *Proceedings of international conference on wireless communication* 49–56
17. Su Y, Xu H, Yan L (2017) Support vector machine-based open crop model (SBOCM): Case of rice production in China. *Saudi J Biol Sci* 24(3):537–547
18. Cheng H, Damerow L, Sun Y, Blanke M (2017) Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. *J Imaging* 3(1):6:1–13
19. Johnson MD, Hsieh WW, Cannon AJ, Davidson A, Bédard F (2016) Crop yield forecasting on the Canadian Prairies by remotely sensed vegetation indices and machine learning methods. *Agric For Meteorol* 218–219:74–84
20. Gornott C, Wechsung F (2016) Statistical regression models for assessing climate impacts on crop yields: a validation study for winter wheat and silage maize in Germany. *Agric For Meteorol* 217:89–100
21. Jayakumar M, Rajavel M, Surendran U (2016) Climate-based statistical regression models for crop yield forecasting of coffee in humid tropical Kerala India. *Int J Biometeorol* 60(12):1943–1952
22. Matsumura K, Gaitan CF, Sugimoto K, Cannon AJ, Hsieh WW (2014) Maize yield forecasting by linear regression and artificial neural networks in Jilin China. *J Agricult Sci* 153(03):399–410
23. Shakil Ahamed ATM, Mahmood NT, Hossain N, Kabir MT, Das K, Rahman F, Rahman RM (2015) Applying data mining techniques to predict annual yield of major crops and recommend planting different crops in different districts in Bangladesh. 2015 IEEE/ACIS 16th international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)
24. Nistor E, Dobrei AG, Dobrei A, Camen D (2018) Growing season climate variability and its influence on sauvignon blanc and pinot gris berries and wine quality: study case in Romania (2005–2015). *S Afr J Enol Vitic* 39:196–207
25. Qu Z, Zhou G (2016) Possible impact of climate change on the quality of apples from the major producing areas of China. *Atmosphere* 7(9), 113:1–18
26. Sugiura T, Ogawa H, Fukuda N, Moriguchi T (2013) Changes in the taste and textural attributes of apples in response to climate change. *Sci Rep* 3(2418):1–7
27. Cozzolino D, Cynkar WU, Dambergs RG, Gishen M, Smith P (2010) Grape (*Vitis vinifera*) compositional data spanning ten successive vintages in the context of abiotic growing parameters. *Agr Ecosyst Environ* 139(4):565–570
28. Fukuoka N, Masuda D, Kanamori Y (2009) Effects of temperature around the fruit on sugar accumulation in watermelon (*Citrullus lanatus*(Thunb.) Matsum. and Nakai) during the latter half of fruit developmental period. *J Japanese Soc Hort Sci* 78(1):97–102
29. Rousphael Y, Cardarelli M, Colla G, Rea E (2008) Yield, mineral composition, water relations, and water use efficiency of grafted mini-watermelon plants under deficit irrigation. *HortScience* 43(3):730–736
30. Idso SB, Kimball BA, Shaw PE, Widmer W, Vanderslice JT, Higgs DJ, Clark WD (2002) The effect of elevated atmospheric CO₂ on the vitamin C concentration of (sour) orange juice. *Agr Ecosyst Environ* 90(1):1–7
31. Sambasivam G, Opiyo GD (2020) A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. *Egyptian Inf J* 22:27–34
32. Karadağ K, Tenekeci ME, Taşaltın R, Bilgili A (2019) Detection of pepper fusarium disease using machine learning algorithms based on spectral reflectance. *Sustain Comput Inf Syst* 28:1–8

33. Jun-De C, Huayi Y, De-Fu Z (2020) A self-adaptive classification method for plant disease detection using GMDH-Logistic model. *Sustain Comput Inf Syst* 100415:1–18
34. Agarwal M, Singh A, Arjaria S, Sinha A, Gupta S (2020) ToLeD: tomato leaf disease detection using convolution neural network. *Procedia Comput Sci* 167:293–301
35. Karthik R, Hariharan M, Anand S, Mathikshara P, Johnson A, Menaka R (2019) Attention embedded residual CNN for disease detection in tomato leaves. *Appl Soft Comput* 105933:1–24
36. Geetharamani G, Pandian A (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput Electr Eng* 76:323–338
37. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145:311–318
38. Maniyath SR, PV V, MN, RP, NP B, NS, Hebbar R (2018) Plant disease detection using machine learning. International conference on design innovations for 3cs compute communicate control (ICDI3C), 41–45
39. Lu Y, Yi S, Zeng N, Liu Y, Zhang Y (2017) Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267:378–384
40. Ebrahimi MA, Khoshtaghaza MH, Minaei S, Jamshidi B (2017) Vision-based pest detection based on SVM classification method. *Comput Electron Agric* 137:52–58
41. Bhange M, Hingoliwala HA (2015) Smart farming: pomegranate disease detection using image processing. *Procedia Computer Science* 58:280–288
42. Espejo-Garciaa B, Mylonas N, Athanasakos L, Fountas S, Vasilakoglou I (2020) Towards weeds identification assistance through transfer learning. *Comput Electron Agric* 171:1–10
43. Yu J, Sharpe SM, Schumann AW, Boyd NS (2019) Deep learning for image-based weed detection in turfgrass. *Eur J Agron* 104:78–84
44. Bakhshipour A, Jafari A (2018) Evaluation of support vector machine and artificial neural networks in weed detection using shape features. *Comput Electron Agric* 145:153–160
45. Dos Santos Ferreira A, Matte Freitas D, Gonçalves da Silva G, Pistori H, Theophilo Folhes M (2017) Weed detection in soybean crops using ConvNets. *Comput Electron Agricult* 143:314–324
46. Fletcher RS, Reddy KN (2016) Random forest and leaf multispectral reflectance data to differentiate three soybean varieties from two pigweeds. *Comput Electron Agric* 128:199–206
47. Cheng B, Matson ET (2015) A feature-based machine learning agent for automatic rice and weed discrimination. *Lecture Notes Comput Sci* 517–527
48. Ahmed F, Al-Mamun HA, Bari ASMH, Hossain E, Kwan P (2012) Classification of crops and weeds from digital images: A support vector machine approach. *Crop Prot* 40:98–104
49. Rumpf T, Römer C, Weis M, Sökefeld M, Gerhards R, Plümer L (2012) Sequential support vector machine classification for small-grain weed species discrimination with special regard to *Cirsium arvense* and *Galium aparine*. *Comput Electron Agric* 80:89–96

Translate2Classify: Machine Translation for E-Commerce Product Categorization in Comparison with Machine Learning & Deep Learning Classification



Priyanshi Gupta and Shatakshi Raman

Abstract Product categorization is a necessary feature of e-commerce websites since it ensures that the websites retrieve related items from the product taxonomy tree accurately. In traditional product categorization methods, machine learning and deep learning classification algorithms are frequently applied. These cater towards product categorization by taking input and then categorising it in one of the pre-defined categories. In this paper, we propose a machine translation-based solution for e-commerce product categorization. We convert the natural language description of a product into a token sequence that reflects the root leaf taxonomy of the product category. In the experiment, three e-commerce product datasets (Flipkart, Walmart, Amazon) have been combined to substantiate the applicability of the natural language models implemented. We demonstrate that ensembling sequence-to-sequence neural networks and the transformer model outperforms state-of-the-art product categorization algorithms in terms of predicted accuracy. In addition, the accuracy comparison for machine learning classification (KNN, Random Forests, SVM), deep learning classification (LSTM, BERT) and neural machine translation models (Seq2Seq, Seq2Seq + Transformer) is shown to validate ensembling the two elements as a better method. In conclusion, we illustrate that attentional sequential models generate product category labels without supervised constraints.

Keywords Neural machine translation · Product categorization · Machine learning · Deep learning · E-commerce · Classification

P. Gupta (✉) · S. Raman

Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India

e-mail: priyanshigupta.cse1@bvp.edu.in

S. Raman

e-mail: shatakshiraman.cse1@bvp.edu.in

1 Introduction

Product categorization is a field of natural language processing research (NLP) [1], which is still one of the most difficult obstacles for e-commerce businesses to overcome. Researchers have been applying machine learning to product categorization problems as AI technology has advanced. For E-Commerce and marketing, product categorization is critical. Increase sales rates, boost your search engine, and enhance your site's Google ranking by properly categorising your items. Your search engine can retrieve items faster if you classify them correctly. As a result, you build a search engine that is both faster and more reliable. Since the search engine is often the first feature that users engage with on E-Commerce pages, a powerful search engine is critical to the overall user experience. Customer discovery is aided by product categorization. You'll be able to build relevant landing pages for your products if you have a good product taxonomy in place. As a result, Google and other search engines would have an easier time indexing the website and products. Finally, this raises the visibility of your brands on search engines, raising the likelihood that consumers may find your website.

Ontologies facilitate the recognition of comparable items and are used on e-commerce sites for product suggestion and duplicate elimination. Despite the fact that people in the organisation are urged to manually enter categories for their items when posting them. Taxonomies facilitate keyword search and provide uniformity in the classification of comparable items, allowing for product suggestion and duplication elimination.

To categorise items from general to particular classifications, e-commerce sites employ hierarchical taxonomies, as shown in Fig. 1 [2]. For instance, the product

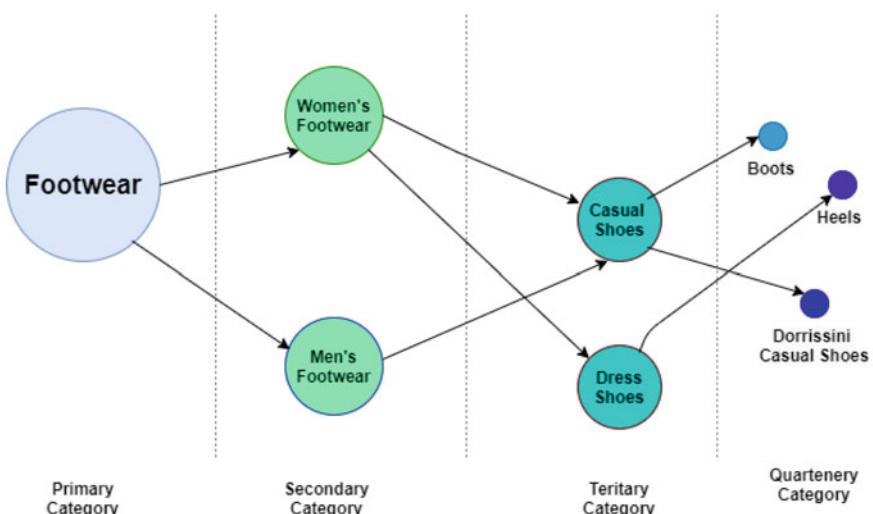


Fig. 1 Classification tree of 3 products showing categorization at 4 levels

‘Dorrisini Party Wear’ falls under the ‘Footwear → Women’s Footwear → Casual Shoes → Dorrisini Casual Shoes’ category on flipkart.com.

By far, the majority of product categorization algorithms have dealt with the situation as a standard ML classification problem, taking a product’s description as input (for example, ‘Key Features of Alisha Solid Women’s Cycling Shorts Cotton Lycra Navy, Red, Navy; Number of Contents in Sales Package is Pack of 3 Fabric Cotton Lycra’) and returning a leaf node describing the product’s designation. Because the classification is a tree, and each leaf node produces a path through one root to another, these algorithms establish an existent root-to-leaf path.

In contrast, we use neural machine translation to solve the categorization problem. A NMT system takes text in one language (predominantly identified as f) as input and produces a sequence of words in another language as its translation (denoted as e). The input f corresponds to a product’s textual specification, and the output e equates to the sequence of categories and subcategories in a root-to-leaf route (e.g., ‘Footwear Women’s Footwear Casual Shoes Dorrisini Casual Shoes’). By structuring product categorization as an NMT problem, our solution outperforms previous algorithms in terms of both technical and economic benefits.

First, big e-commerce companies usually have many languages on their websites (for example, www.amazon.com in English and www.amazon.cn in Chinese) and have invested substantially in their MT system capabilities. By utilising existing machine translation frameworks for commodity classification, we are lowering the technology debt that these businesses suffer.

Second, machine translation systems have increased their accuracy in recent years [3–5], also approaching human parity on certain language pairs [6] due to deep learning [7]. By looking at the challenge as a machine translation problem, we were able to bring the best of machine translation technology to bear on the issue of product classification in a cost-effective manner. In Sect. 5, we show how our MT technique is used on empirical data.

Third, MT systems are immune to the disparities of language by virtue of their sheer existence, making them immune to errors in a product’s description. (e.g., ‘Key Features of Alisha Solid Women’s Cycling Shorts Cotton Lycra Navy, Red, Navy; Number of Contents in Sales Package is Pack of three Fabric Cotton Lycra’ and ‘Number of Contents in Sales Package is Pack of 3 Fabric Cotton Lycra; Key Features of Alisha Solid Women’s Cycling Shorts Cotton Lycra Navy, Red, Navy’). As a result, MT systems are suitable for working with the complexities that come with natural language descriptions of products.

Fourth, the MT approach is a way to create all already existing root-to-leaf pathways in a taxonomy tree and is equipped with the ability to create novel root-to-leaf connections not found in the taxonomy that are more relevant to the user search. The configuration of a commodities catalogue is changed from a tree to a directed acyclic graph with these additional paths (DAG). This is a powerful transition since it permits a single product to have more than one root-to-leaf pathway. Unlike prior systems, which had just one path, this system had several paths. This is more in line with scientific observations that people have a tendency to look at things in different ways. Pumps, for example, are considered mainly boots because they are worn on

the feet; however, they are often referred to as heels. The different ways to pair shoes emphasise the different approaches to footwear: as a system of taxonomic categories like footwear and heels, or as situational categories like women’s footwear and dress shoes. Similarly, items in various product worlds have distinct qualities, and more than one system of categories is required to adequately capture these features. By offering multiple routes to fetch the category, the MT scheme better responds to human intuition than earlier techniques, and it has the capability to improve user product navigation. For example, a user who prefers to think of pumps as dress shoes and a user who prefers to think of them as women’s footwear will quickly find what they need. Compared to the previous statistical machine translations or SMT, that were used before, NMT uses deep learning algorithms to teach itself to produce quality results rather than being rule-based, thus overcoming one of the biggest drawbacks of SMT. We have used the description of the products provided as our first language, translating it into the primary category which is used as the target language for our NMT model [1].

Then, in Sect. 2, we briefly discuss relevant work before describing machine translation systems (Sect. 3). The experimental dataset (Sect. 4), experimental methods (Sect. 5) and assessment methods (Sect. 6) are then discussed, as well as how we utilise them to categorise products. Following that, we present a qualitative analysis of our findings (Sect. 7). Finally, we’ll talk about what we’ll be doing in the future (Sect. 8).

2 Related Work

Sun described Chimera, a WalmartLabs solution that effectively classifies items using a combination of learning, hand-crafted rules, crowdsourcing, in-house analysts, and developers, in 2014 [8]. They investigated how to improve machine learning-based classifiers as well as how to create appropriate training data quickly (e.g., using active learning). Later on, sequence-to-sequence learning with neural networks [9], a research published by Google in 2014 [10], acknowledged that while DNNs are effective classifiers for large databases if the input is labelled, they cannot map sequences. The paper then introduced the novel approach of Sequence learning as a way of generalised end-to-end learning for sequences that make minimal assumptions on its structure. Two multi-layer LSTM structures were utilised in the work, one to map the sequence into a fixed-dimensional vector and the other to decode the target from the vector.

Yang et al. [11] show how to anticipate a collection of labels for a given occurrence using multi-label classification (MLC). The sequence-to-sequence (Seq2Seq) model, which was trained using the maximum likelihood estimation approach, was successfully applied to the MLC challenge and exhibits an unusual capability to capture high-order correlations between labels before proposing a wholly unique solution. When the reward feedback is supposed to be independent of label order throughout this procedure, we may reduce the model’s dependency on label order

while still capturing high-order correlations between labels. Extensive probing indicated that technique outperformed competing baselines and successfully reduced susceptibility to label order. The suggested approach not only captured high-order correlations between labels, but it also decreased the reliance on output label order.

Pane et al. use Multinomial Naive Bayes for Holy Quran translation in their article published in 2018 [9]. The challenge of categorization of Quranic verses that will be grouped into a single subject arises as a multi-label classification problem.

To handle multi-label classification, the study created a whole new classifier model. After various data preparation processes such as case folding, tokenization, and stemming, the system was created using Multinomial Naive Bayes. The system then heavily used a bag of words as a feature extraction strategy, and Bayesian modelling techniques to text classification were demonstrated to be effective.

In 2017 [12], McCann et al. employed deep LSTM models using ensemble seq2seq to contextualise word vectors. These context vectors (CoVe) outperform unsupervised word and character vectors on a wide range of prominent NLP tasks.

Transfer learning or transferring the knowledge from an encoder to a model. The paper uses a completely unique approach for transferring information from a machine translation-trained encoder to a variety of downstream NLP applications. The research employs a novel method of transmitting data from a machine translation-trained encoder to a range of downstream NLP applications. In general, baselines that used random word vector initialisation, baselines that used pre trained word vectors from a GloVe model, and baselines that used word vectors from a GloVe model performed better than baselines that used random word vector initialisation, baselines that used pre-trained word vectors from a GloVe model, and baselines that used word vectors from a GloVe model.

In their 2016 paper, Xu et al. [13] specialise in the difficulty of translating classification models with significantly poor bilingual dictionaries. They applaud two unique strategies that combine unsupervised word embedding across languages, supervised mapping of embedded words between languages, and probabilistic classification model translation. On a benchmark corpus of Reuters news stories (RCV1/RCV2) in multiple languages, the algorithms beat equivalent baseline approaches using standard bilingual dictionaries or highly incomplete ones in CLTC.

Luong et al. [14] look at the concept of attention and how it affects the ability to learn. This study looks at two types of attentional mechanisms: a global technique that pays attention to all or any source words in the shortest amount of time, and an area technique that only looks at a selection of source words at a time.

Maggie Yundi Li [1] used an ensemble of attentional sequence-to-sequence models to supply product category labels without the necessity of supervised constraints in 2018. Such unrestricted product classification indicates potential additions to the prevailing category pyramid and uncovers unclear and repeated category leaves SIGIR eCom'18. Wu et al. [15] demonstrate that a really compact convolution can compete with the simplest reported self-attention outcomes. They then provide dynamic convolutions, which are more efficient and quicker than self-attention. The work predicts separate convolution kernels-based solely on the current time-step in order to detect the significance of context information. The number of operations in

this approach scales linearly with the length of the input, whereas self-attention scales quadratically. In large MT, language modelling, and abstractive summarization tests, dynamic convolutions beat strong self-attention models.

Linet al. [16] address MLTC with a higher-level model that includes semantic unit representations and multi-level dilated convolution, as well as a related hybrid attention mechanism that is applied both at the word-level and hence at the semantic unit level. On the datasets RCV1-V2 and Ren-CECps, their suggested model outperformed baseline methods, and is competitive with deterministic hierarchical models and more robust in categorising low-frequency labels.

In their 2018 publication, Howard et al. [17] introduce Universal Language Model Fine-tuning (ULMFiT), a replacement methodology. They proposed Universal Language Model Fine-tuning (ULMFiT), a replacement methodology that addressed these difficulties and enabled robust inductive transfer learning for any NLP task, similar to fine-tuning ImageNet models: The same 3-layer LSTM architecture outperformed highly designed models and trans <https://arXiv:1801.06146v5> [cs.CL] with the exact same hyperparameters and no modifications aside from tweaked dropout hyperparameters. On six well researched text categorization problems, they applied transfer learning methodologies.

3 Neural Machine Translation

The concept of employing machines to translate across languages dates back to 1949, when Warren Weaver originally proposed it. Since then till late 1980s, machine translations were done with the help of rule-based systems or RBMTs. Later due to development of statistical models, translations became more efficient and thus statistical machine translations influenced till 2000s. However, in 2013, Nal Kalchbrenner and Phil Blunsom developed a novel end-to-end encoder-decoder framework for machine translation that involves utilising CNNs to encode the supplied text into vector embeddings and then using RNNs to decode them into target language as shown in Fig. 2.

Nonetheless, if long text sequences were passed due to the vanishing gradient problem of RNN, it would become harder to control long distance dependencies, and thus, attention mechanisms were sought to resolve this problem. As a result, Junczys-Dowmunt conducted experiments on the ‘United Nations Parallel Corpus,’ and found that NMT was on par with or outperformed SMT in all 30 translation directions indicated by BLEU scores. When given a sequence of numbers, NMT uses an artificial neural network to predict a series of numbers. Each word in the input sentence (for example, English) is encoded as a number or linguistic vector, which the neural network then converts into a series of numbers representing the translated target phrase (e.g., German). NMT, unlike RBMT and SMT, does not evolve sentences into target language; instead, it employs a two-step approach, as seen in Fig. 3. It is made up of two parts: an encoder and a decoder.

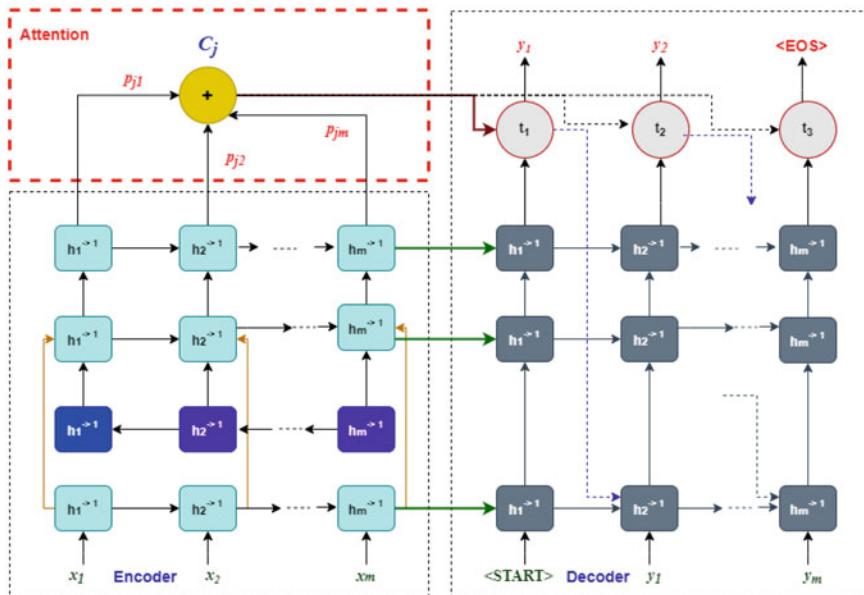


Fig. 2 Diagram shows the basic architecture of an NMT with attention

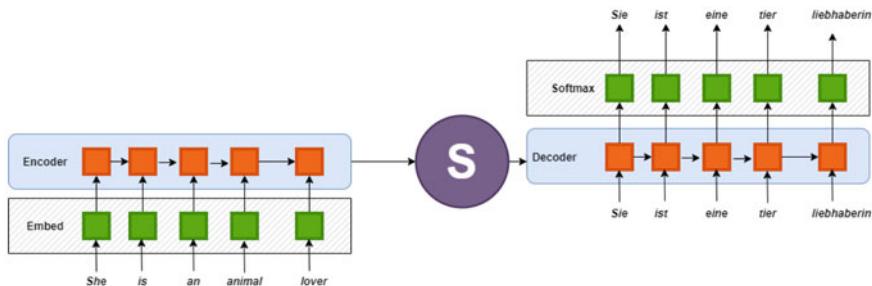


Fig. 3 Basic encoder-decoder architecture employed by NMT systems

Encoder's primary task is to transform the source text language into linguistic 'vectors' for the neural networks to work upon, while Decoder's main task is to decode these linguistic vectors into another language and then rearrange them into meaningful sentences.

4 Dataset

The datasets that we utilised to conduct our experiments and assess our findings are defined in this section. For the objective of this study, three E-commerce companies' datasets were used: Flipkart, Amazon, and Walmart. Table 1 summarises the datasets utilised from Flipkart, Walmart, and Amazon and Fig. 5, 6, 7 and 8 visualise the categories in the dataset as clusters (Fig. 4).

The features extracted from the main datasets and used are shown in Table 2.

The individual datasets were highly unbalanced and not substantial in size to draw conclusions from. Hence, to overcome this problem, we combined the individual datasets into a main dataset on which the models were trained. Figure 6 a, b represents the clusters of dataset which we got after cleaning the data:

The data as discussed present in all 3 was skewed primarily either due to the demand of certain categories like Clothes when compared to computers, or the popular categories on different platforms, which if used would have resulted in biased training. Thus, combining and randomizing all 3 datasets resulted in better less skewed results, while training the models. The treemap down below shows the division of various clusters, in different ways using the size of the tree chunks and the

Table 1 Summary of flipkart, walmart and amazon datasets

	Flipkart	Amazon	Walmart
Language	English	English	English
Source	https://www.kaggle.com	https://data.world	https://data.world/
Data size	20,000	10,000	30,000

Fig. 4 Clusters for walmart dataset

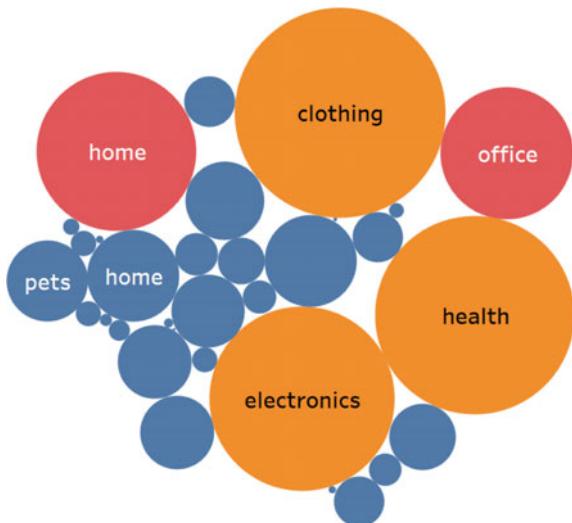


Fig. 5 Clusters for flipkart dataset

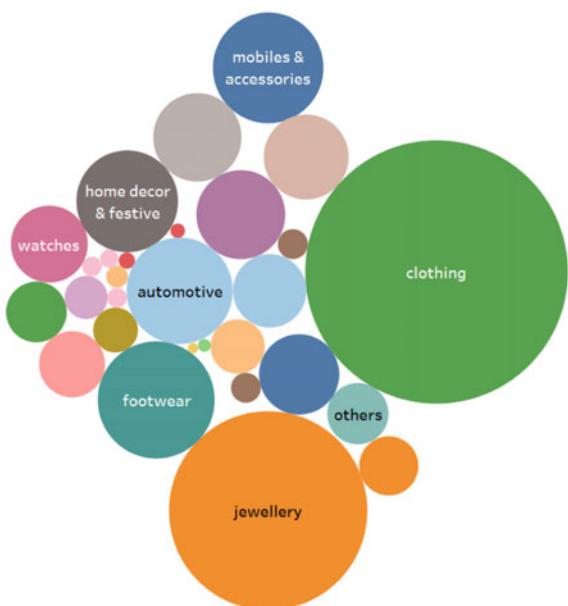


Fig. 6 Clusters for amazon dataset

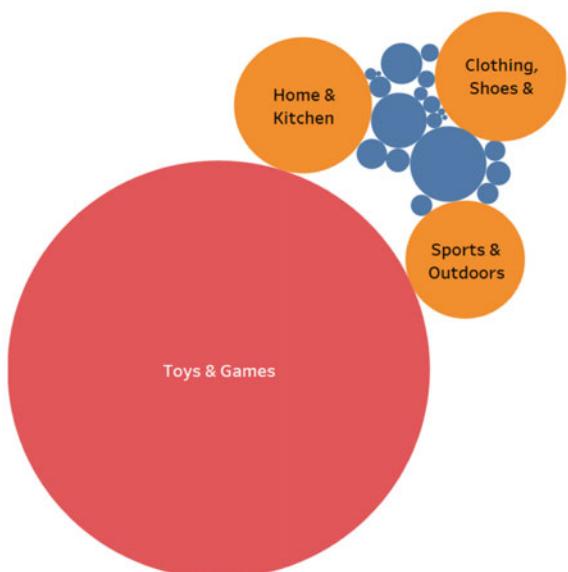




Fig. 7 **a** Visualising the clusters of categories of the combined dataset **b** visualising the clusters of categories of the combined dataset

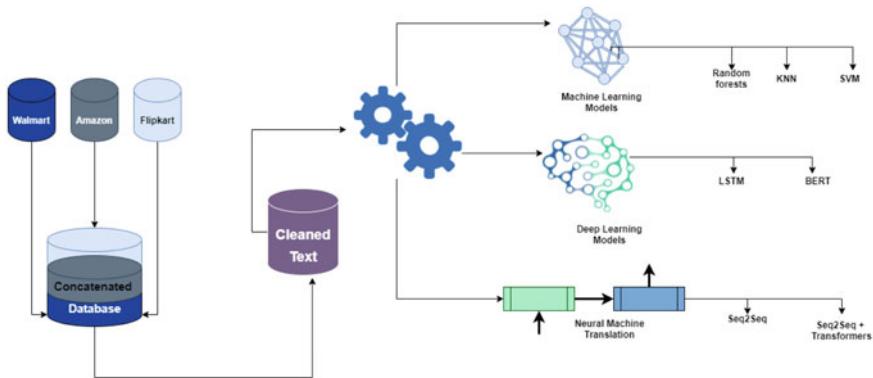


Fig. 8 The figure above represents the pipeline followed for this study

Table 2 Summary of features used

Feature name	Type	Description
Product category tree	STR	The description of the product (Primary Feature)
Description	STR	Used to extract the primary category

intensity of the colour, emphasizing that clothing, toys and games, health, electronics are the most popular categories, amongst users on all 3 platforms.

5 Experiments and Baseline Models

In this section, we describe the baseline models we use, including machine learning models, deep learning models, and neural machine translation models. Figure 7 represents the pipeline followed for this study. The settings of these models are described and the hyperparameters are mentioned in the sub sections for each approach.

5.1 Machine Learning Classification Models

Machine supervised learning is described as a type of data analysis in which the desired output is known. The learning method generates an inferred function to predict output values based on a study of a known training dataset. After adequate training, the system can offer goals for any new input. The learning algorithm may also compare its output to the exact, expected results and detect faults, allowing the model to be modified accordingly. Random Forests, Naive Bayes, SVM, and KNN

were the machine learning classification models explored for this job. The TF-IDF vectorizer is used to convert input phrases into vectors.

K-Nearest Neighbour.

K-Nearest Neighbour (KNN) [17], algorithm predicts the required number (k) of the closest neighbouring data points. Here, the pre-processing of the info is critical because it impacts the space measurements directly. Unlike others, the model doesn't have a mathematical formula, nor any descriptive ability. Here, the parameter 'k' must be chosen wisely; as a worth less than optimal results in bias, whereas a better value impacts prediction accuracy. The range limit for SelectBestK was used as $k = 1200$, and therefore, the chi-square test that measures dependence between stochastic variables for our KNN classifier.

Support Vector Machines.

SVM [18], too belongs to the supervised learning category. After training on a defined label set, and category they can categorise the incoming input, thus solving classification problem. Conventionally SVMs function by building a hyperplane from the provided data, the best fits in separating the categories. The hyperplane is built optimally by calculating the support vectors of data and maximising the margin. All that data that falls on one side is grouped together, while the others are sent into different categories. SVMs tend to be powerful for classification problems, thanks to their ability to not overfit the data. The range limit for SelectBestK was set to 1200, and so the chi-square test, which quantifies the dependency between stochastic variables, was employed for our SVM classifier.

Random Forests.

A Random Forest [19] is a consistent ensemble of several Decision Trees (or CARTs), albeit it is more commonly used for classification than regression. Individual trees are constructed using bagging (i.e., the agglomeration of bootstraps, which are nothing more than using, several train datasets formed by sampling of records with replacement) and divided using fewer features. The resultant diversified forest of uncorrelated trees has lower variance and, as a result, is more resistant to data change and carries its prediction accuracy to fresh data. However, the approach does not perform well for datasets with a high number of outliers, which must be addressed prior to model construction. For our RF classifier, we utilised $k = 1200$ as the range limit for SelectBestK and the chi-square test, which quantifies dependency between stochastic variables.

5.2 Deep Learning Classification Models

ANNs are used in deep learning to conduct complex computations on vast volumes of data. It's a form of machine learning that's focused on the human brain's structure and function. Because of its demonstrated utility in processing big data, which is also

characterised by nonlinear processes and complicated relationships, deep learning techniques have become a common trend in many science realms, including product categorization. They're commonly used for extracting high-level abstract functions, which outperform conventional models in terms of accuracy, interpretability, and understanding and processing data. The deep learning classification models that were considered for this task were LSTM and BERT.

Long Short Term Memory.

Long short-term memory (LSTM) [20] has transformed machine capabilities in diverse aspects such as, Google's speech recognition, Google Translate's automated translations, and Amazon's Alexa answers have all improved as a result of this paradigm. Prior to the development of LSTM, recurrent neural networks had a relatively discrete efficiency. The hyperparameters used for the implementation of LSTM are:

- Input Embedding Dimension: 6000
- Output Dimension: 50
- Dropout Ratio: 0.25
- Callback factor: 0.5
- Callback Patience: 2.

The Plot Loss and accuracy graphs for LSTM are shown in Figs. 9 and 10.

BERT.

The robust Transformer in BERT's design uses an attention mechanism to manipulate the input phrase. The transformer is composed of numerous attention blocks, all of which adds attention to the input sequences before transforming them employing linear layers. It just adds seq2seq mapping layers together. The disadvantage of transformers is that the input pattern is not recognised as RNNs are recognised. For example, if the first or last phrases are all the same, they would be regarded as the same tokens. By use of positional embedding, where words are in a phrase, BERT overcomes this problem. Before being sent to the forward network, the input tokens

Fig. 9 Plotting model loss for LSTM

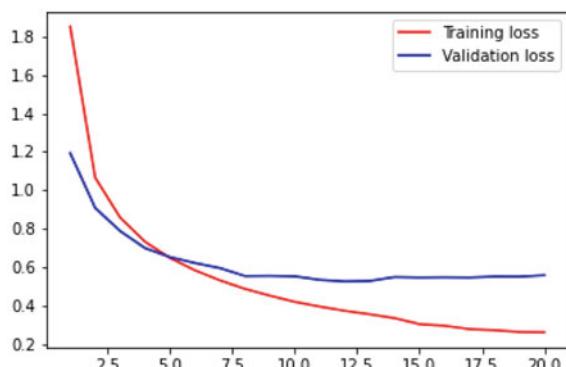
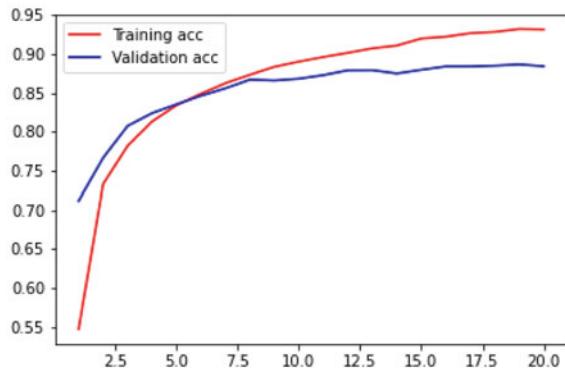


Fig. 10 Plotting accuracy curve for LSTM



are placed into the tokens. BERT is conditioned on grouped phrases that help to create the two sentences that distinguish them into a distinctive embedding. For this exact embedding, segment embedding is the name. Used are hyperparameters of the BERT Model for this study:

- BERT Model Used = ‘bert_en_uncased_L-12_H-768_A-12’
- Dropout Ratio: 0.1

The plot loss and accuracy graph for BERT is shown in Figs. 11 and 12.

When model loss and accuracy are plotted, it is clear that the BERT model achieves optimum output after only one epoch and does not overfit on training results. Rather, a steady loss is sustained.

Fig. 11 Plotting model loss for BERT

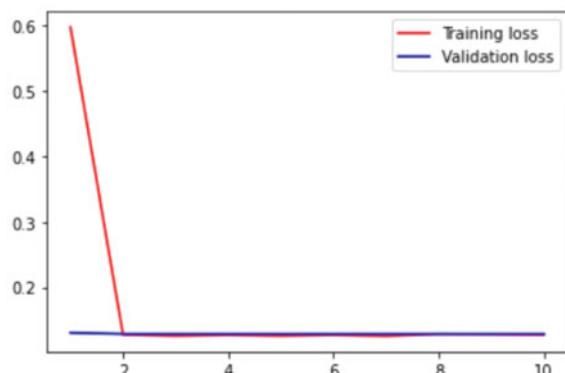


Fig. 12 Plotting accuracy curve for BERT

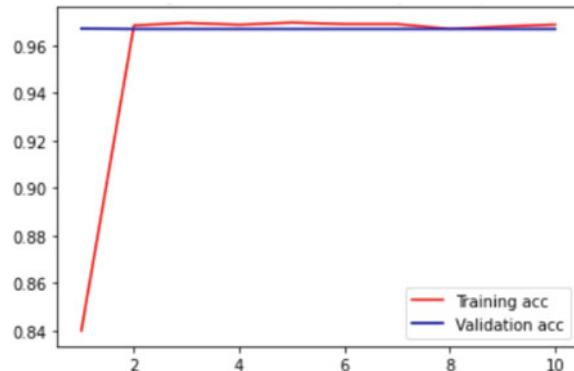
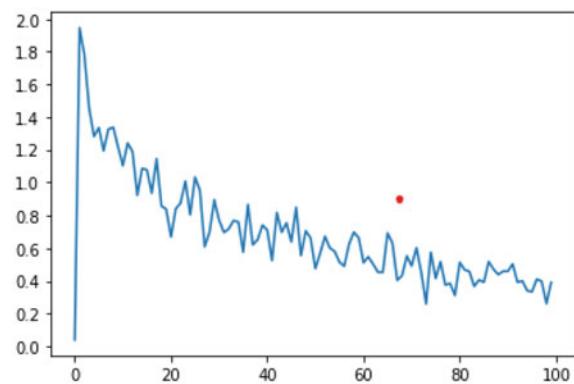


Fig. 13 The variation of negative log likely loss (NLL LOSS) with the number of epochs



5.3 Neural Machine Translation Models

The NMT models that were considered for this task were Seq2Seq and the combination of Seq2Seq and Transformer model.

Sequence-to-Sequence Learning.

Seq2Seq is an encode-decoder-based machine translation and language processing system that matches a sequence input with a label and attention-value to a sequence output. The aim is to use a certain token in combination with two RNNs to forecast the future state sequence. We employ the Seq2Seq paradigm, introduced by Luong [14], to implement it. The hyperparameters that we have used are:

- Hidden Layer Size: 256
- Dropout Ratio: 0.1
- Teacher Forcing Ratio: 0.5.

In Fig. 13, the negative logs that are lost are displayed. For machine learning models, it is a loss function which shows us how well they perform; smaller the loss, better the model.

Ensemble Level Sequence-to-Sequence + Transformer Learning.

In Seq2Seq learning, we discovered that the model seems unable to properly translate the category until it reaches the desired section of the attribute. To get around this, we encoded the location with a sinusoid, as shown in Fig. 14, then ensembled the Transformer embedding to the right of the Seq2Seq output. Embedding + positional encoding + dropout constitute the Transformer embedding. We employ the Seq2Seq model of Luong [14] and the Transformer model presented by Vaswani [21] for Ensembling. We combine the attentional Seq2Seq and Transformer models by aggregating their decoder results and the learner versus loss graph is shown in Fig. 15.

The hyperparameters used for the Ensembled Seq2Seq and Transformer model are shown in Table 3:

Fig. 14 Positional encoding

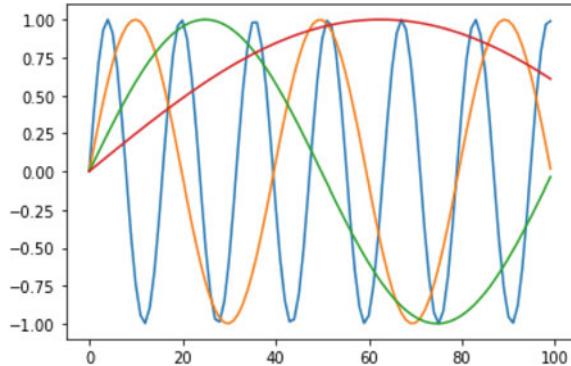


Fig. 15 Learner versus loss graph

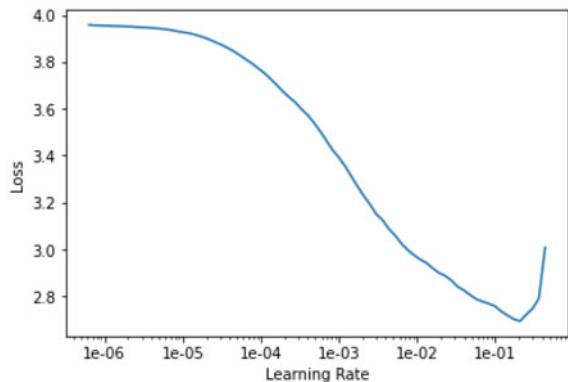


Table 3 Hyperparameters of ensembled attentional Seq2Seq and transformer model

	Multihead Seq2Seq attention	Transformer
Input/output embedding dimension	256	256
Number of layers	6	6
Number of attention heads	8	8
Size of intermediate layer	–	1024
Dropout ratio	0.5	0.5

6 Results

The models are overfitting on training data which will be a major issue when the size of data is increased. The results comparison for our machine learning versus deep learning versus machine translation models are shown in Table 4 and Fig. 16

Table 4 Results of our NMT models versus ML classification versus deep learning classification

		Precision	Recall	F1 Score	BLEU
Machine learning models	KNN classifier	0.61	0.53	0.56	–
	SVM classifier	0.61	0.41	0.46	–
	Random forests classifier	0.62	0.54	0.56	–
Deep learning models	LSTM classifier	0.88	0.88	0.88	–
	BERT classification	0.90	0.90	0.90	–
Machine translation models	Seq2Seq attention + teacher forcing	0.80	0.81	0.78	–
	Seq2Seq + transformer ensemble	–	–	0.895	0.74
	Seq2Seq + transformer Ensemble + label smoothing			0.885	0.68

**Fig. 16** The graph above compares the outcomes of all the models

As we can observe from the results Table, our machine translation performs better than the state-of-the-art machine learning and deep learning classification models and as good as Google's BERT classifier. For our simple encoder-decoder Seq2Seq model, we found out that adding Teacher Forcing improves the accuracy for this MT model. We also observed that ensembling the outputs of Transformer and Seq2Seq models and adding positional encoding, we get better results. However, while adding label smoothing, we observe sufficient decrease in accuracy and an increased training and validation loss. In machine learning models, Random Forests classifier has the highest accuracy and in deep learning models, state-of-the-art BERT model performs best. Overall, our ensembled model has the best accuracy and a high BLEU score value for this E-Commerce Product dataset. In machine translation models, we also observe the advantage of the product description being translated to a more relevant primary category which is beneficial for the user search and product retrieval.

7 Conclusion

Seq2seq approach and using machine translation accomplished a major improvement in the field of NLP in several tasks and has made leaps in the field of NLP. Another advantage of machine translation in product categorization is that the translator model translates the description into unique primary categories that are more beneficial to the user search, thus improving the relevance for the products. The model is considered provisional autoregressive models, implying that they are capable of generating a sequential element by accustoming another sequential element, thus being superior to LSTMs that cannot be used for general transduction undertakings. With proper tuning of our parameters, more concise results were obtained. We have devised our study to perform comparisons between various classification methods that are used for product categorization. We analysed different classification procedures from basic Random Forests, nearest neighbours, support vectors, and then moved to deep learning models involving LSTM and BERT, and lastly Seq2Seq model, with and without attention, achieving that Seq2seq with attention serves the purpose best according to our study. Another advantage of machine translation in product categorization is that the translator model translates the description into unique primary categories that are more beneficial to the user search, thus improving the relevance for the products. To epitomise, this study by no means undermines the success attainable by BERT or fine tuning LSTMs over other architectures. It is therefore critical to scrutinise the dataset and task at hand to research first prior to deciding on the model.

8 Future Work

For machine learning models, we can further use boosted models. Boosting algorithms connect weak learners, also known as base learners, to create a strong rule. Some of the common boosting algorithms are AdaBoost, XGBoost, CatBoost. Each time the base learning method is used, a new weak prediction rule is generated. This is a step-by-step procedure. The boosting method combines numerous weak rules into a single powerful prediction rule after many rounds. For deep learning models, the models in concern perform decently in predicting the product categories. The model does not overfit, indicated by concurrent loss curves. Deep learning tends to perform better with more and more data, since the available data is scarce and categories are severely imbalanced and the combined data is less imbalanced. Use of graphical neural networks can also be implemented for building knowledge graphs.

References

1. Yundi Li M, Tan L, Kok S (2018) Don't classify, translate: multi-level e-commerce product categorization via machine translation. Workshop on information technologies and systems 2018 (WITS2018) [arXiv:1812.05774v1](https://arxiv.org/abs/1812.05774v1) [cs.CL]
2. Yundi Li M, Tan L, Kok S, Szymanska E (2018) Unconstrained production categorization with sequence-to-sequence models. Rakuten data challenge at the 2018 SIGIR workshop on ecommerce
3. Kalchbrenner N, Blunsom P (2013) Recurrent continuous translation models. In Proceedings of 2013 conference on empirical methods in natural language processing, 1700–1709
4. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In Proceedings of the 27th international conference on neural information processing systems, 3104–3112
5. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In Proceedings of the third international conference on learning representations
6. Hassan H, Aue A, Chen C, Chowdhary V, Clark J, Federmann C, Huang X, JunczysDowmunt M, Lewis W, Li M, Liu S, Liu TY, Luo R, Menezes A, Qin T, Seide F, Tan X, Tian F, Wu L, Wu S, Xia Y, Zhang D, Zhang Z, Zhou M (2018) Achieving human parity on automatic chinese to english news translation. Comput Res Repository, abs/1803.05567
7. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. <http://www.deeplearningbook.org>
8. Sun C, Rampalli N, Yang F, Doan A (2014) Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. Proc VLDB Endowment 7:1529–1540
9. Pane RA, Mubarok MS, Huda NS, Adiwijaya (2018) “A multi-label classification on topics of quranic verses in english translation using multinomial Naive Bayes,” 2018 6th international conference on information and communication technology (ICoICT) pp 481–484, <https://doi.org/10.1109/ICoICT.2018.8528777>
10. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. [arXiv:1409.3215v3](https://arxiv.org/abs/1409.3215v3) [cs.CL]
11. Yang P, Ma S, Zhang Y, Lin J, Su Q, Sun X (2018) A deep reinforced sequence-to-set model for multi-label text classification. [arXiv:1809.03118v1](https://arxiv.org/abs/1809.03118v1) [cs.CL]
12. McCann B, Bradbury J, Xiong C, Socher R (2017) Learned in translation: contextualized word vectors. v1

13. Xu R, Yang Y, Liu H, Hsi A (2016) Cross-lingual text classification via model translation with limited dictionaries
14. Luong T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In Proceedings of the 2015 Conference on empirical methods in natural language processing, 1412–1421
15. Wu F, Fan A, Baevski A, Dauphin YN, Auli M (2019) Pay less attention with lightweight and dynamic convolutions. [arXiv:1901.10430v2](https://arxiv.org/abs/1901.10430v2) [cs.CL]
16. Lin J, Su Q, Yang P, Ma S, Sun X (2018) Semantic-unit-based dilated convolution for multi-label text classification. [arXiv:1808.08561v2](https://arxiv.org/abs/1808.08561v2) [cs.CL]
17. Altman NS (1992) “An introduction to kernel and nearest-neighbor nonparametric regression”. *Am Statis* 46(3):175–185
18. Cortes C, Vapnik VN (1995) “Support-vector networks”. *Machine Learning*. 20(3):273–297. CiteSeerX 10.1.1.15.9362. <https://doi.org/10.1007/BF00994018>. S2CID 206787478
19. Ho TK (1995) Random decision forests. Proceedings of the 3rd international conference on document analysis and recognition, Montreal, QC, 14–16 August 1995. pp 278–282
20. Van Houdt G, Mosquera C, Nápoles G (2020) A review on the long short-term memory model. *Artif Intell Rev* 53:5929–5955. <https://doi.org/10.1007/s10462-020-09838-11>
21. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser LU, Polosukhin I (2017) Attention is all you need. In Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems 30, Curran Associates, Inc., 5998–6008
22. Howard J, Ruder S (2018) Universal Language Model Fine-tuning for Text Classification. [arXiv:1801.06146v5](https://arxiv.org/abs/1801.06146v5) [cs.CL]
23. Yu H-F, Ho C-H, Arunachalam P, Somaia M, Lin C-J (2013) Product title classification versus text classification. Technical report, Department of Computer Science, National Taiwan University, Taipei, Taiwan
24. Xia Y, Levine A, Das P, Di Fabbrizio G, Shinzato K, Datta A (2017) Large-scale categorization of japanese product titles using neural attention models. In Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Vol 2, Short Papers. Association for Computational Linguistics
25. Shen D, Ruvini JD, Sarwar B (2012) Large-scale item categorization for e-commerce. In Proceedings of the 21st ACM international conference on information and knowledge management, CIKM ’12, New York, NY, USA. ACM, 595–604
26. Shafto P, Coley JD (2003) Development of categorization and reasoning in the natural world: novices to experts, naive similarity to ecological knowledge. *J Exp Psychol Learn Mem Cogn* 29(4):641–649
27. Ross BH, Murphy GL (1999) Food for thought: cross-classification and category organization in a complex real-world domain. *Cogn Psychol* 38(4):495–553
28. McAuley JJ, Pandey R, Leskovec J (2015) Inferring networks of substitutable and complementary products. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 785–794. ACM
29. Kozareva Z (2015) Everyone likes shopping! multi-class product categorization for e-commerce. In NAACL HLT 2015, The 2015 conference of the North American Chapter of the association for computational linguistics, Human Language Technologies

Fake Feedback Detection to Enhance Trust in Cloud Using Supervised Machine Learning Techniques



Harsh Taneja and Supreet Kaur

Abstract Lots of enterprises have commenced to move toward a cloud-based infrastructure for their services. Organizations of distinct types, sizes, and industry are adopting cloud services for a broad range of use cases, such as software development, product deployment, backup of data, email services, and numerous tasks. Although there are various advantages of using cloud computing as the ability to scale rapidly, store data remotely, reducing the overall cost. Cloud computing has some major trust concerns due to privacy issues such as lack of transparency, lack of user control over user's data. Thus, lack of trust is a big issue for cloud service providers and cloud infrastructure users. This paper centers on how to achieve a trust management system around cloud services by finding fake feedbacks on CloudArmour's Trust Feedback Dataset. Post preprocessing and data cleaning, multiple models like Support vector machine, Naive Bayes, Random Forest, and Decision Tree were trained on our dataset to find the most accurate model. An ensemble model has also been proposed which has proved to be best algorithm for identifying fake feedbacks and has shown accuracy, precision & recall value of 93.55%, 92.90%, and 92.82%, respectively.

Keywords Trust management · Fake feedback detection · Cloud service provider · Cloud customer

1 Introduction

Cloud computing allows on-demand network access to a shared pool of configurable computing resources comprising of servers, networks, storage and so on which can

H. Taneja (✉)

Department of Computer Science and Engineering, Punjabi University, Patiala, India

Bharati Vidyapeeth's College of Engineering, New Delhi, India

S. Kaur

Department of Computer Science, Punjabi University Regional Centre for Information Technology and Management, Mohali, India

be used by paying as per usage only. Advantages of using cloud computing includes the ability to scale rapidly, store data remotely, reduce expenses incurred by using computing and storage resources only on-demand basis provisioning mechanisms relying on the pay-per-use model. [1] Cloud computing also has some disadvantages such as: becoming network-dependent for accessing own data and services; some unresolved trust issues; and security concerns regarding cloud computing [2]. To overcome these drawbacks, cloud customer (CC) relies on the cloud service provider (CSP) who becomes responsible for the storage and handling of users' data, thus reducing overall control by the user to a large extent [3, 4]. While ensuring high availability, there is also a chance of flow of data between jurisdictions because CSPs make use of replicating data in multiple datacenters [5–7]. In this process, data may be moved to a datacenter in different legal jurisdictions, thus jeopardizing the data, increasing the risk factors, and enhancing the legal complexity. Various Security Issues also exist for cloud computing such as Cloud services may lead to easier access to private and confidential information of the user whereas the user has limited command over the data lifecycle. Some concerns of CCs are there for CSPs regarding availability and backup, because guaranteeing adequate availability and regular backups is not easy [8–10]. Trust Issues for Cloud Computing are there as customers of cloud computing lack control and are not provided with the demanded level of transparency [11]. Because of low trust levels, organizations make use of various contracts and other trust mechanisms such as Service level agreements (SLA), auditing, ratings, measurements, etc. [12–15]. To overcome these problems, reputation-based trust management systems have been enhanced from time to time by providing the capability to CC to compute trust with respect to CSPs. This work focuses on identifying fake feedbacks given by a CC to a CSP using CloudArmor dataset [16] by proposing an ensemble classifier model. This model has depicted higher performance as compared to other classification models.

2 Methodology

The methodology section has been categorized into these two parts: datasets and implementation which has been elaborated in Sects. 2.1 and 2.2, respectively.

2.1 Dataset

Cloud Armor dataset was collected by [16] by developing a cloud service crawler engine that collected cloud services available on the Web. After parsing a lot of relevant links, Cloud Armor was able to gather meta-data for almost 6000 cloud services. From the collected information, [Ref] prepared numerous datasets of cloud services. CloudArmour's Trust Feedback Dataset was created by collecting cloud service consumers' feedback from leading review Web sites such as Cloud Hosting

Reviews, Best Cloud Computing Providers, Cloud Storage Reviews, and Ratings. Cloud Armor collected more than 10,000 feedbacks furnished by approximately 7000 consumers to 113 cloud services. The feedbacks are based on various Quality of Service (QoS) attributes. This dataset has been taken in our work and has been pre-processed to optimize the outcome. Apart from preprocessing the data, labels have been incorporated in the dataset to enable the dataset to be applied on supervised machine learning methods.

2.2 Implementation

The first step is to perform preprocessing and data cleaning, so we can achieve uniformity across the dataset. Also the data type of values are checked in the “timestamp” column are in string or date-time format. All the tuples which have “timestamp” in the date-time format were converted to the string format and saved in a new column named “Date”. “Timestamp” values of tuples which had the correct format already saved as in “Date” column. On the other hand, non-uniform “timestamp” column was dropped.

After analysis, unnamed columns present in the dataset were dropped. More columns that had “NULL” Value in most of the rows were removed from considerations as well. These Columns were “Response Time”, “Speed”, “Storage Space”, “Feature”, “Customer Service”, “Level of Expertise”, “Accessibility”, “Ease of use” and “Security”. Further, any row which had either “NaN” or “NULL” value in the “Trust Result” column had to be deleted. Any row which had “NULL” value in any of three columns of “Availability”, “Price”, or “Technical Support” was deleted. Next, the “Trust Result” column was analyzed. A new Boolean column “Target” is created comprising of value ‘1’ for all the tuples which had “Trust Result” greater than a pre-determined threshold value, and value “0” otherwise. Similarly, the number of unique values in the “Cloud Service Name” column was calculated. A unique number was inserted in the “Cloud Service Number” column corresponding to the “Cloud Service Name”, this was done for ease of use and can be compared to the enumeration of various “Cloud Service Name”.

To train the machines so as to perform supervised machine learning, following pre-existing machine learning techniques have been applied on the dataset: SVM; Naive Bayes; Random Forest; and Decision Tree. To further optimize the results, simple voting ensemble technique has been proposed and machine has been trained using ensemble of above-mentioned supervised machine learning techniques.

Soft Voting is computed as Eq. 1 [17].

$$X = \max \sum_{j=1}^m \text{CM}_j \quad (1)$$

Table 1 Performance of various classification models

Model	Accuracy	Precision	Recall
SVM radial bias	85.066	87.8	87.898
SVM linear bias	85.454	88.406	88.104
Naïve Bayes	81.823	86.669	86.877
Random forest	90.994	91.165	90.802
Decision tree	91.545	91.528	91.465
Voting ensemble	93.552	92.908	92.824

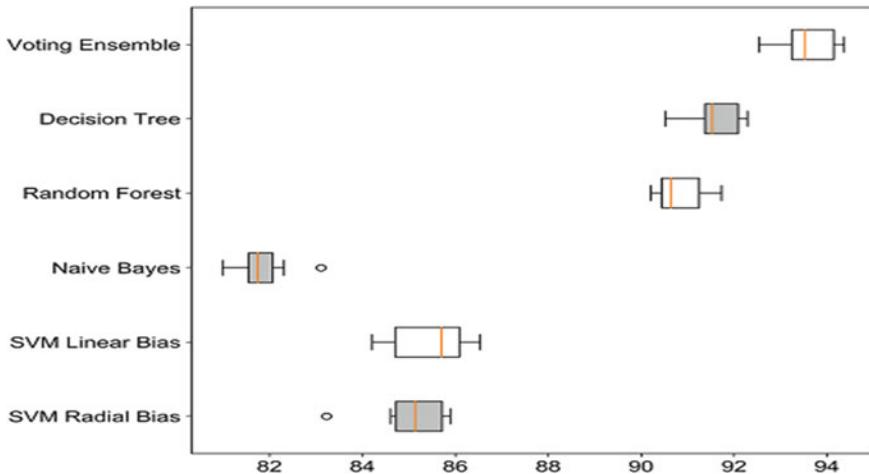


Fig. 1 Accuracy box plot for various classifiers

where X is value for ensemble model, CM is classification model and m is maximum number of CM used to classify the data. Algorithm for proposed ensemble model is:

1. Classify the data into two classes “1” and “0” using following algorithms: SVM_linear, SVM_Radial, Naïve Bayes, Random Forest and Decision Tree.
2. Compile classified values of above-mentioned models and form ensemble model using Eq. 1.
3. Compute the values for each case using simple ensemble technique.
4. Compare the results of ensemble model with above-mentioned classification models.

3 Results

Results have been demonstrated through Table 1 containing values of various classifiers with respect to accuracy, precision, and recall; and Figs. 1, 2, 3 and 4 showing box plot comparison of classifiers on the basis of following performance parameters:

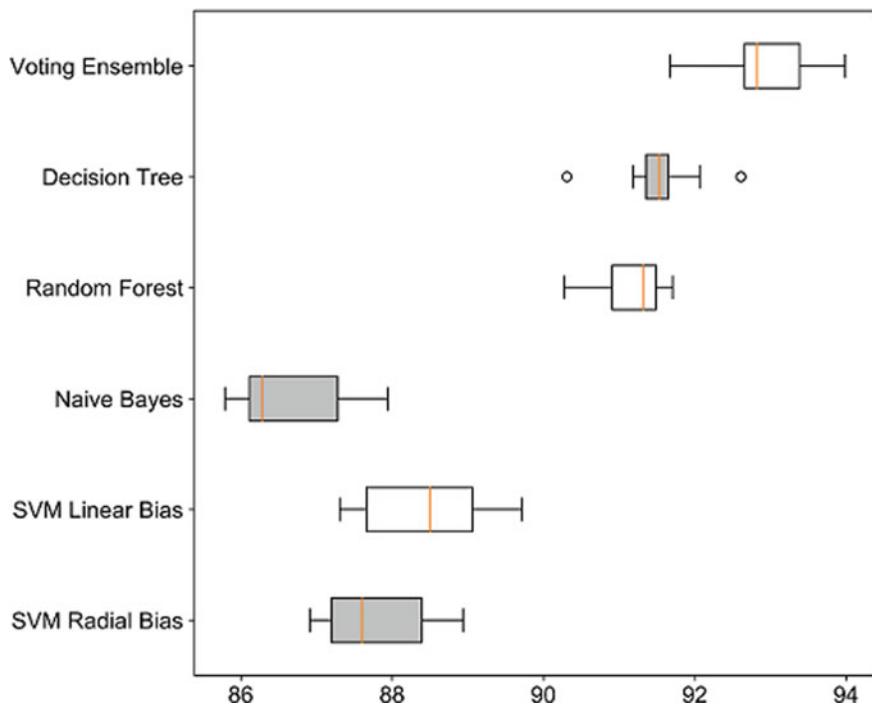


Fig. 2 Precision box plot for various classifiers

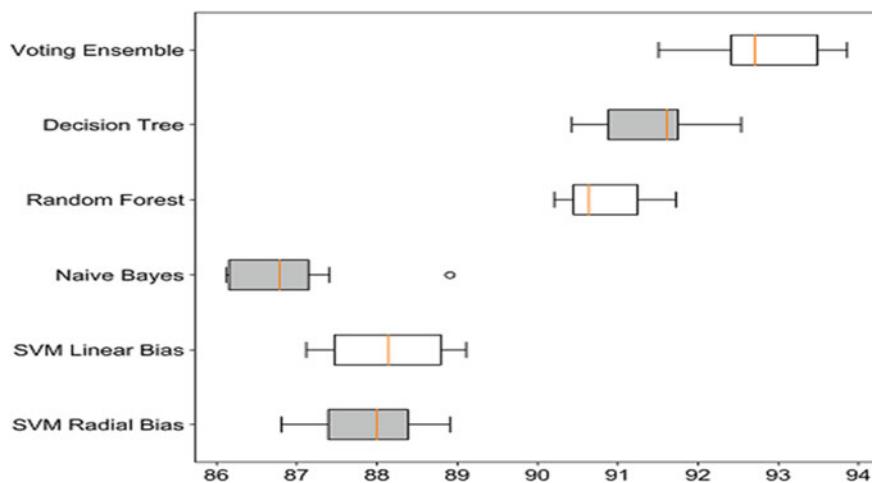


Fig. 3 Recall box plot for various classifiers

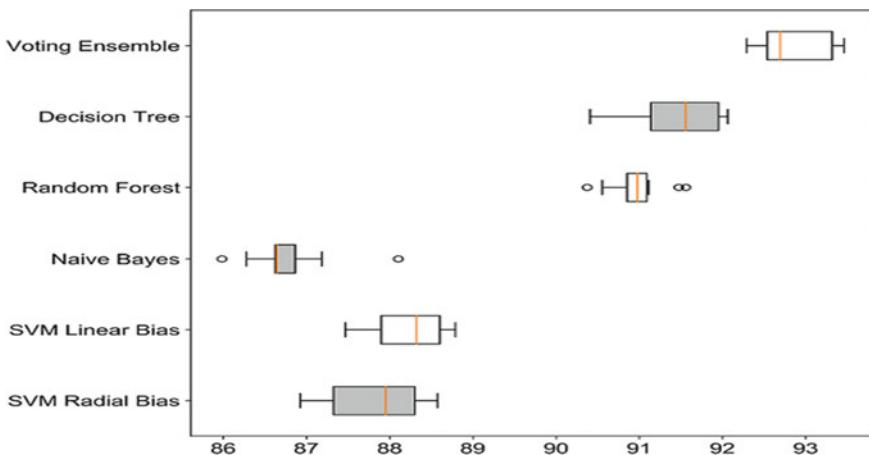


Fig. 4 F1-score box plot for various classifiers

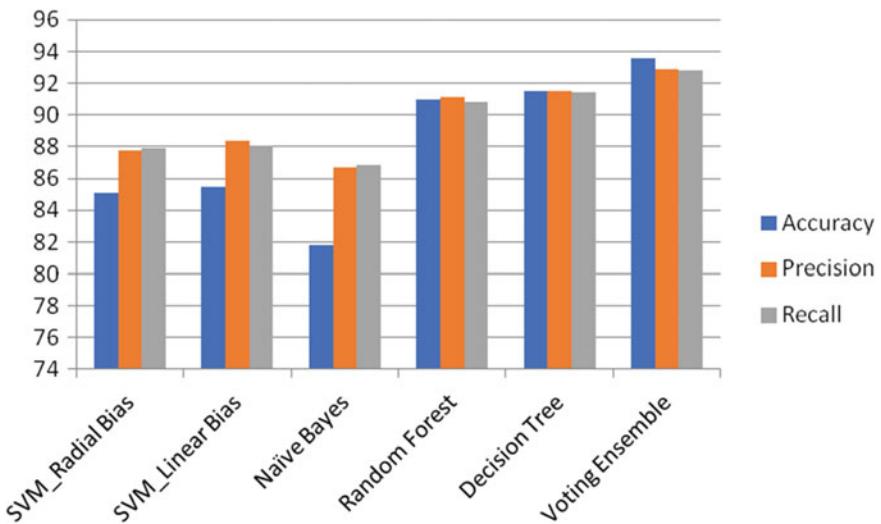


Fig. 5 Performance evaluation on the basis of accuracy, precision, and recall performance parameters

accuracy; precision; recall; and F1-score. Figures 1, 2, 3, 4, and 5 and Table 1 clearly depict that ensemble model has outperformed with respect to all the above-mentioned performance parameters.

Figure 5 is graphical representation of the performance shown by numerous classifiers and bars clearly demonstrate that ensemble has outperformed other models in terms of accuracy, precision, and recall performance parameters.

4 Conclusion

There are various advantages of using cloud computing as the ability to scale rapidly, store data remotely, reducing the overall cost. But cloud computing has some major trust concerns due to privacy issues such as lack of transparency and reduced user control over their data. Thus, Trust management and its enhancement is big issue for cloud service providers and cloud infrastructure users. CloudArmour's Trust Feedback Dataset is used for research purposes so as to study the impact of malicious users and fake feedbacks on trust of cloud service provider.

Initially, dataset had a number of attributes that were not of our use, after removal of those attributes and deletion of tuples which did not satisfy certain criteria, numerous classification models like support vector machine, Naive Bayes, random forest, and decision tree were trained on our dataset to find the most accurate model. Further, an ensemble model has been proposed which has depicted best results in comparison with all other above-mentioned pre-existing classification models.

Future scope relevant to this research comprises of the possible application of machine learning techniques on various similar datasets, improving and expanding the dataset underuse, taking more attributes under our consideration for model training.

References

1. Srivastava P, Khan R (2018) "A review paper on cloud computing". *Int J Adv Res Comput Sci Softw Eng*, ISSN: 2277–128X 8(6)
2. Nazir M (2012) "Cloud computing: overview and current research challenges". *IOSR J Comput Eng (IOSR-JCE)* ISSN: 2278–0661, ISBN: 2278–8727, 8(1)
3. Pearson S, Benameur A (2010) "Privacy, security and trust issues arising from cloud computing". 2nd IEEE international conference on cloud computing technology and science
4. Liu YC, Ma YT, Zhang HS, Li DY, Chen GS (2011) "A method for trust management in cloud computing: data coloring by cloud watermarking". *Int J Autom Comput*
5. Ko RKL, Jagadpramana P, Mowbray M, Pearson S, Kirchberg M, Liang Q, Lee BS (2011) "TrustCloud: a framework for accountability and trust in cloud computing"
6. Akram RN, Ko RKL (2014) "Digital trust—trusted computing and beyond: a position paper" IEEE 13th international conference on trust, security, and privacy in computing and communications, pp 884–92
7. Gu L, Wang C, Zhang Y, Zhong J, Ni Z (2014) Trust model in cloud computing environment based on fuzzy theory. *Int J Comput Commun Control* 9(5):570–83
8. Lynn T, van der Werff L, Hunt G, Healy P (2016) Development of a cloud trust label: a delphi approach. *J Comput Inf Syst* 56(3):185–193
9. Mao C, Lin R, Xu C, Qiang He (2016) "Towards a Trust prediction framework for cloud services based on pso-driven neural network" 4th international conference on advanced cloud and big data (CBD'16)
10. Ruan A, Martin A (2016) "RepCloud: Attesting to cloud service dependency" IEEE conference
11. Ruan A, Martin A (2015) NeuronVisor: defining a fine-grained cloud rootof- trust. In Proceedings of the sixth international conference on trusted systems (INTRUST), pp 184–200
12. Alhanahnah M, Bertok P, Tari Z, Alouneh S (2017) "Context-aware multifaceted trust framework for evaluating trustworthiness of cloud providers". *Future Gener Comput Syst*

13. Ruan Y, Durresi A (2019) A trust management framework for clouds. *Comput Commun* 144:124–131
14. Papadakis-Vlachos Papadopoulos K, González RS, Dimolitsas I, Dechouniotis D, Ferrer AJ, Papavassiliou S (2019) “Collaborative SLA and reputation-based trust management in cloud federations”, *Future Gener Comput Syst* 100:498–512
15. Sharma K, Shrivastava G (2014) Public key infrastructure and trust of web based knowledge discovery. *Int J Eng Sci Manage* 4(1):56–60
16. Noor TH, Sheng QZ, Yao L, Dustdar S, Ngu AH (2015 Mar 4) CloudArmor: supporting reputationbased trust management for cloud services. *IEEE Trans Parallel Distrib Syst* 27(2):367–380
17. Cao J, Kwong S, Wang R, Li X, Li K, Kong X (2015 Feb) Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing* 3(149):275–284

Face Recognition Using Artificially Intelligent Methodologies on FERET and FEI Datasets



Nilay Pant, Devanshu Rathee, and Rahul Gupta

Abstract Artificial intelligence and computational science have been developing at a phenomenal rate in the past decade. Adapting to increasing amounts of data in the world, scientists shifted a lot of their focus towards data science in the recent past. One area of AI is facial recognition from raw image data. In modern times it would be a boon to be able to accurately and quickly predict information from image data, especially when it comes to identifying individuals. This work utilizes a double pronged approach to analyze and improve the work done in the field, by testing validity of previously hailed models on large publicly available controlled face image datasets FERET and FEI. The goal of this paper is to fit implementation of the best-suited subspace reduction technique and support vector classifier along with CNN and test their ability to scale to large datasets. This work shows the difference in efficient scalability between CNN and SVM. Additionally, it achieves an improvement upon past results obtained on the FEI dataset using convolutional neural networks and the PCA + SVM technique.

Keywords Facial recognition · Principal component analysis (PCA) · FERET · FEI · Support vector machines (SVM) · Deep learning · CNN

1 Introduction

The past decade brought with it immense leaps in digital imaging technology and also in the computational power of personal and professional analytical machines (computer). Today, in our surroundings, we can very well observe the importance of facial recognition technologies. It has had an impact on the various scientific and non-scientific fields such as neuroscience, document management and restoration, law and order, computer vision and even psychology. It facilitates instantaneous recognition of a human face and we have become increasingly reliant on this field owing to the major leaps and bounds it has taken. Especially the application of this

N. Pant (✉) · D. Rathee · R. Gupta

Delhi Technological University (Formerly Delhi College of Engineering), Delhi, India

technology in the areas of cyber security, where the face is used as a biometric and in our day-to-day mode of communication viz. social media apps. Real world applications are still growing further; both in performance and popularity, owing to increasing amount of research being done to increase both its reliability as well as accuracy [1]. Facial recognition using artificial intelligence has indeed evolved into a modern cornerstone in the pursuit of scientific grandeur. By using computer hardware to analyze various features of the human face we can train various recognition and prediction models (machines) to identify human faces.

Here we use the FERET database (1996) [2] and the FEI face database [3] for implementing various facial recognition systems and analyzing them to arrive at our own unique model. Therefore, here we implement various algorithms on the FERET dataset to observe its behaviour, which in turn will help us in deducing various different insights for further research.

The FERET dataset took birth in the year 1993 when the United States Army commissioned the pertaining project George Mason University and Army Research Laboratory in Adelphi in Maryland, US.

Since then, many additions, modifications and improvements have been made to the structure of the FERET dataset (which today, is facilitated and managed by the NIST, US). Also, many new face image datasets with better image type distribution and more diverse applications have been developed. Due to lack of diversity and very large size of FERET dataset, we have also used a newer, smaller and more controlled FEI dataset for our model evaluation.

Before arriving at the crux of this research, it would be good to understand why facial recognition is so advantageously placed among modern data science research fields. An analysis of some existing applications of facial recognition technologies in the real world will help paint a better picture.

1.1 Applications of Facial Recognition

1.1.1 Three-Dimensional Recognition

This field takes facial recognition up a notch. Here a person's visage is mapped using 3-D sensors for capturing the information about the face in a 3-D space. In turn, this information further is used for identification of different features of the face such as jaw size, distance between eye sockets, etc. One of the many advantages of using this technique is that the results do not vary with varying degrees of face illumination and it can analyze and recognize different faces from different viewpoints and angles as it gets an additional distance metric to base predictions upon. This technique vastly improves the precision of analyzing the faces. This feat is achieved by development of sophisticated sensors that project light on the face for its analysis. Three cameras are used embedded with sensors for the mapping of the face and which are pointed at different angles. The 3-D matrices obtained are explained in Fig. 1.

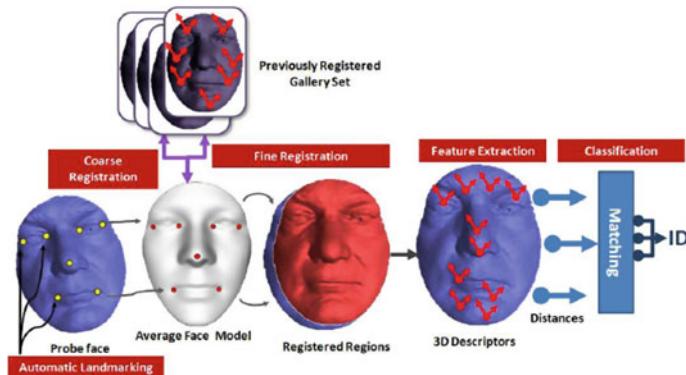


Fig. 1 Three-dimensional recognition

1.1.2 Thermal Cameras

This is also one of the many techniques for facial recognition. Figure 2 shows real-time implementation of thermal cameras for analysis of the faces by marking and differentiating the facial nodes. In this case, the camera exclusively captures the face without taking into consideration of various accessories such as sunglasses, hats or makeup. It analyzes the features of the faces by measuring the temperature of face. This technique can be used for verifying and analyzing faces in even low light conditions or night time environments [4]. But one of the many challenges of this technique is the lack of availability of data on the thermally enhanced images. Although at certain institutes, such as IITD-PSE and Notre Dame, thermal face databases have been created in 2016 are continuous work is underway for augmenting thermal face recognition implementations.

Fig. 2 Use of thermal cameras for face detection



1.2 Problem Statement

This research work was undertaken with a goal to:

- Analyze current research work on facial recognition and FERET dataset to establish best practices.
- Apply different machine learning techniques on the FERET and FEI data set and compare the results.
- Explore the room for improvement for identifying faces in the facial recognition technology system.
- Compare performance of different algorithms for two separate datasets of varying sizes and analyze performance at scale.
- Analyze viability of software based, facial recognition as a biometric identification device.
- Develop knowledge of machine learning algorithm implementation.

2 Literature Review

2.1 Concepts Behind the Simulation

As research on facial recognition kept on progressing, the process was classified, categorized and broken down into steps in many different ways by many different researchers. Upon a brief analysis of research papers [5, 6] it was seen that both of them classified entirety of face recognition approaches in three categories:

- (A) **Holistic/global approach** - entire face taken as a single feature vector by creating feature matrix with pixel values.
- (B) **Feature-based/component-based approach** - localized features such as nose, lips, eye-cheek distance, etc. are used as feature vectors.
- (C) **Hybrid approach** - type of approach that usually combines the two above categories in one or another way.

Another paper [7] also established three broad groups but switched out the term “hybrid method” with “soft computing methods”. Some Papers [8], suggested two main groups viz. global approaches and component-based approaches; but then papers like [9] again found a third category in the amalgamation of these two. Ultimately it was revealed that there is overall consensus and even though each scientist uses differing terminology for defining the concepts and principles, a closer and informed look at their work reveals that all converge at a common understanding of the classification of facial recognition approaches. Which are the same three categories as given above.

Similarly, papers [5, 7] suggest a 3-step process which is common between varying facial recognition methods. Similarity to this three-step process can also be found in

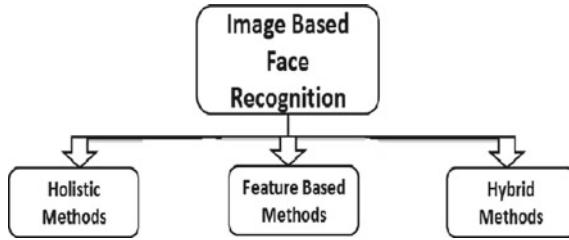


Fig. 3 Categorisation of approaches to face recognition

the works of [8]. Thus, the previous research again seems to converge upon the three steps shown in Fig. 3, which are common to all facial recognition approaches.

- (A) **Face Detection\Localization:** identifying and separating a given region of an image as a face is the process of localisation or detection. This can be useful in implementations dealing with face tracking, emotion mapping, etc.
- (B) **Face Normalization\Feature Extraction:** This step deals with standardization and normalization of the feature vectors that are extracted in the first step. Vectors must not be affected by or be reflecting information about position, scale and face rotation angle.
- (C) **Face Recognition\Matching:** as can be seen from the name this is the final step and it deals with superimposing or matching the vector of each face to a known face database and ultimately attempting to correctly identify (classify) an input face.

Figure 4 that synchronous to the three approaches there are common steps behind every facial recognition implementation as well. Having developed a solid understanding of approaches we come to the dataset required for facial recognition. In the paper [10], it has been said that there is an acute requirement of comprehensive facial images dataset and a recognized test for evaluating the accuracy of models framed on the said dataset. In numerous works [5, 10] the FERET dataset and the FERET tests have been proposed as a viable solution to this seemingly daunting problem.

3 Methodology

As shown in Fig. 5 the scope of work done in this research concerns itself with broadly four steps, three steps in synchronization with the generalized steps of facial recognition as discussed in the literature survey and one step of the unique output and conclusions drawn from the result of the methodologies explained here. The entirety of this work has been implemented in Python's programming environment with its excellent library support for machine learning implementations. FERET dataset was acquired with proper permissions [2] and the FEI face dataset is publicly

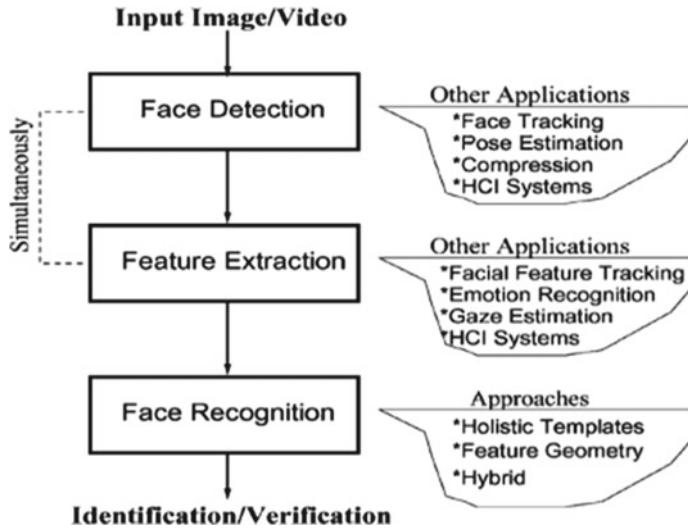


Fig. 4 General steps of face recognition mechanism

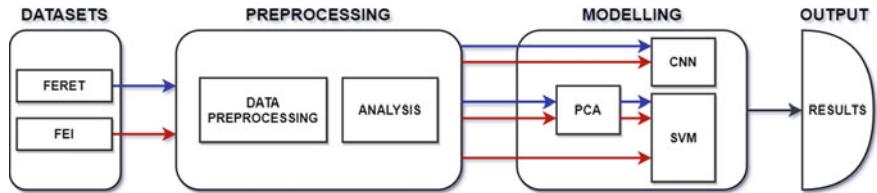


Fig. 5 Process flowchart

available for download at FEI's website [3]. Preprocessing implementations included scale reduction, channel reduction and manipulation of image matrices, all of which were achieved with the regular expression(re), cv2, glob, pickle and pandas library [11]. The third step, classification and identification models, was implemented using numpy for basic calculation and sklearn, tensorflow and keras library for implementing the support vector machine classifiers and the convolutional neural networks. Principal component analysis was used as the subspace reduction technique after the authors discovered it to be the most accurate for the problem statement mentioned above, based on insights drawn from study of previous literature in the field of facial recognition. The results and plots were generated using the metrics and matplotlib libraries. Microsoft's Visual Studio Code was utilized at the primary text editor and code compiler while Google's Colab (free) environment was used for training, evaluating and validating the performance of the models. Colab platform indeed turned out to be a blessing since the dataset, especially FERET, was immensely huge for any reasonably accurate models to be trained successfully on the local machines available. Our local system was running a Windows 10 with 8 gigabytes of RAM,

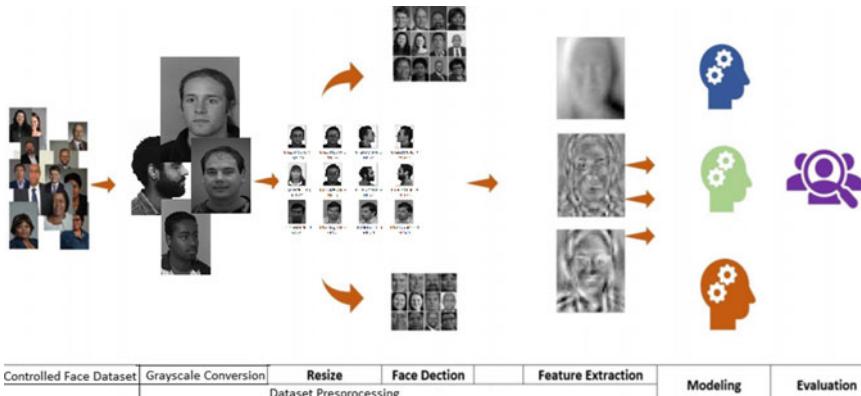


Fig. 6 Life of image in model

2.4 GHz processor and 2 GB dedicated graphics RAM. For a clearer understanding of the pipeline of data Fig. 6 shows the stages an image went through in the scope of this work.

3.1 Dataset

See Table 1.

The raw images, if used directly had a dimension of 640X480 pixels which would have led to a feature matrix with 307,200 features each, for 2800 images. Implementing neural networks and even support vector classifiers on such a huge amount of data would be highly inefficient and might even be a problem beyond the scope of normal computing devices because, in the case of support vector classifiers where support vectors for each data point are calculated the computation might run into billions or maybe even trillions of separate relations each, again, for at least 2800 images [12]. Before this could be tackled with dimensionality reduction methods like PCA or ICA, the raw data had to be reduced to a metric where it could be easily processed and a smooth flow of data through the classifier architecture could be

Table 1 Datasets

S.No	Attributes	FEI	FERET
1	Image size	640×480	256×384
2	No. of images	2800	14,126
3	No. of subjects	200	1564
4	Colour/Grey	Colour	Colour
5	Image conditions	Controlled	Controlled



Fig. 7 Gallery view of FEI

established. The following steps list out the preprocessing methods implemented by the author on the raw images obtained from FERET and FEI datasets (Fig. 7):

1. **Conversion of the face images to grayscale** in order to reduce the dimensions of the data being fed to the machine learning model by reducing the images from RGB to a single grayscale channel. This was done because in exploration of related literature the authors happened upon a book [13], a professor of neurology at the Albert Einstein College of Medicine, which contained real world and computer vision related evidence pertaining to how colour values can mask or make it difficult for an individual and an intelligent algorithm to identify hidden patterns in the image that are presented before them.
2. **Converted grayscale images were resized** to 64X64 pixels. The resulting reduced image(right) is compared to original input image (left) as follows:
3. **The establishment of a target variable**, which is basically the identity of the subject whose picture is being discussed here (“Person 1” and “Person 10” in the above figure), was done by using the regular expression (re) library of Python to split out the “person_id” from the path of the image, for which the OS library was utilized.
4. **The 2-dimensional pixel data from the 64X64 pixel images were flattened** to create a tuple of 4096 features for each image. (Figure 8 shows the raw image on left and the preprocessed image right before flattening on the right.)
5. **Each row was concatenated** with its corresponding “person_id” by using concat function of dataframe object to combine feature vector columns with target variable columns.
6. **Thus in the final dataframe object representing the feature vector** to be input into the machine learning models, each image was represented by 4096 features (extracted variables) and its corresponding class(labels), which is the “person_id” of the subject whose image it is.

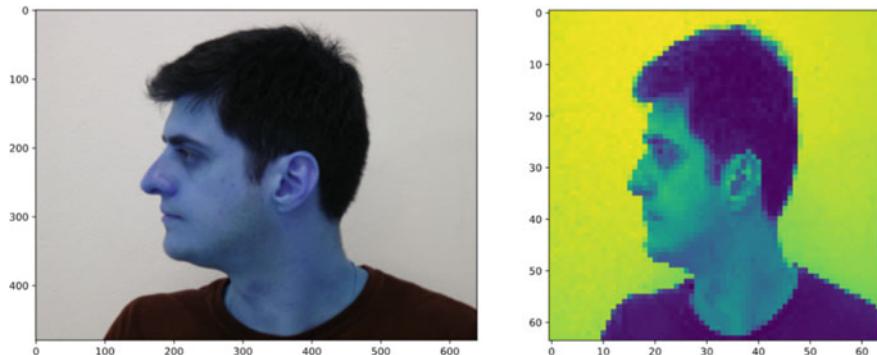


Fig. 8 Image before and after preprocessing

The final feature vectors for the FEI and FERET datasets were as shown in Fig. 9 and 10:

The resulting datasets were used as input to find the best-suited support vector machine classifier for the purpose of performing accurate facial recognition upon images from this dataset.

	pixel 0	pixel 1	pixel 2	pixel 3	pixel 4	...	pixel 4092	pixel 4093	pixel 4094	pixel 4095	person_id
0	204	203	205	205	203	...	19	18	17	15	1
1	191	192	193	192	195	...	16	16	17	21	1
2	187	187	190	189	188	...	14	15	16	21	1
3	186	187	189	189	192	...	14	15	16	21	1
4	186	184	186	182	187	...	15	17	19	119	1

5 rows x 4097 columns

Fig. 9 FEI feature vector

	pixel 0	pixel 1	pixel 2	pixel 3	pixel 4	...	pixel 4092	pixel 4093	pixel 4094	pixel 4095	person_id
0	141	141	138	135	133	...	114	76	77	95	1119
1	133	129	127	127	123	...	229	229	234	233	1118
2	53	53	51	51	53	...	109	105	102	102	1118
3	129	129	127	127	124	...	68	74	63	60	1118
4	140	137	133	133	132	...	121	105	68	60	1119

5 rows x 4097 columns

Fig. 10 FERET feature vector

3.2 SVM for Facial Recognition

An SVM classifier works with the goal to find the best decision boundary to separate the classes of the problem statement. In problems where the decision space is one dimensional, it finds the best point, the best line in two dimensions, the best plane in three dimensions and the best hyperplane for all problems which deal with decision spaces of more than three dimensions [14]. This paper only concerns itself with the best hyperplane as most modern problems addressed using support vector classification deal with greater than three-dimensional spaces.

The authors have provided Fig. 11 for better understanding of the functioning of the complex mathematical parameters and hyperplanes that play a huge role in tuning support vector classifier models.

The datasets used in this research presented a unique problem, in the sense that there were a large number of classes (1200 and 200 unique subjects participated in the collection of FERET and FEI datasets, respectively) and only 4–14 images per class to facilitate a successful classification. The authors used a grid search method to train the classifier with 12 different combinations of “C” and “gamma”.

By recursively fitting the FEI and FERET dataset to all these 12 models [15] the authors were able to identify the best parametric values that suited the scope of this research. The parametric values providing best accuracy with respect to both datasets are shown in Table 2:

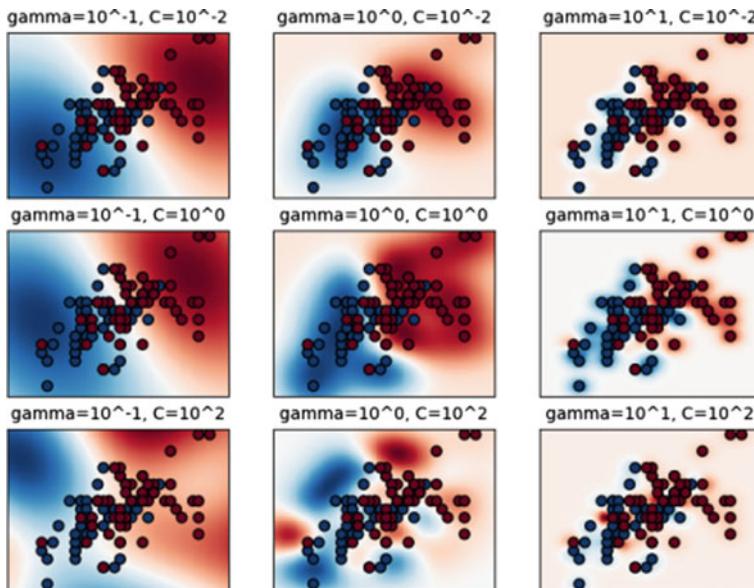


Fig. 11 Understand SVM parameters—C and gamma

Table 2 Parameter passed and obtained

Datasets/parameters	Passed		Obtained	
	C	Gamma	C	Gamma
FEI	[0.1, 1.0, 10, 100]	[auto, scale, 0.001, 0.0001, 0.00001]	10.0	0.001
FERET	[0.1, 1.0, 10, 100]	[auto, scale, 0.001, 0.0001, 0.00001]	1.0	auto

3.3 Subspace Reduction Techniques

As discussed in the literature review section of this work it has been established in numerous previous researchers works that Principal Component Analysis (PCA) method is the best method to implement subspace reduction on image data. The next step in the analysis that this research deals with, was to compare the performance of Support Vector Machine classifier on a reduced subspace obtained after application of PCA on the FERET and FEI dataset.

The PCA method implemented by the authors returned a reduced dataset with 722 instead of 4096 features or components (subspace reduction of 80%) while maintaining a data loss of less than 1%.

It is imperative to understand that such a component, when projected will look like an image but the underlying data that we are projecting is in no way close to being an image in itself. They can be understood as mathematical relations between values of each feature vector with all other feature vectors [16]. The underlying logic behind PCA algorithm dictates that not all of these relations are required to accurately classify one image from another and hence its works to reduce the number of these projections and then make the feature vector from that reduced set all the while maintaining informational integrity of the data.

For a better understanding see the first thirty components' projection with weights in decreasing order in Fig. 12:

These 30 and 72 more such components are basically enough for the computer to accurately identify one image as distinct from any other in the dataset [17]. Together these 102 parameters for 1960 training images of the FEI dataset were fit on an SVM classifier with the same parameters as before.

3.4 Deep Learning for Facial Recognition

In the work preceding this step the authors have implemented manual feature extraction or no feature extraction on the dataset and used the same to train the SVM classifier model to make accurate recognition (predictions) of human faces from the given dataset. Manual feature extraction is not a resource intensive method and allows us to define the tolerance for data loss based on our own requirements. Deep learning methods on the other hand provide easier classification in complex classification tasks like facial recognition as they perform both feature extraction and



Fig. 12 First 30 components after PCA

classification by themselves [18]. Figure 13 diagrammatically explains the structure and functioning of a convolutional neural network.

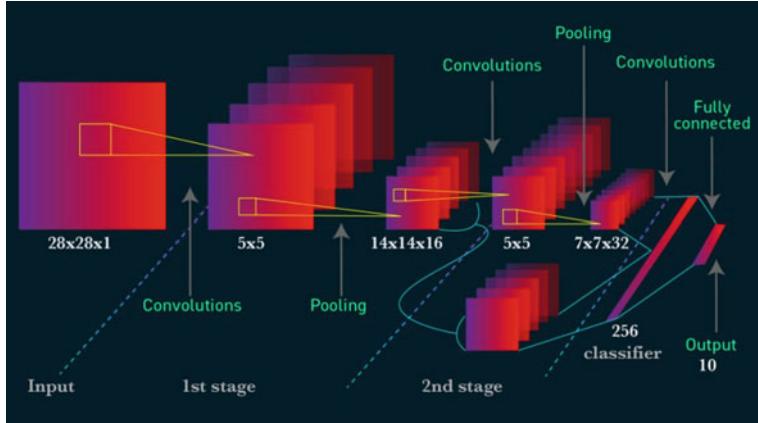


Fig. 13 Representation of convolutional NN layers

In this research Convolutional Neural Network (CNN or ConvNet) has been implemented to achieve the task of facial recognition after analyzing various other research work which advocates the efficacy of CNN over other deep learning methods for all image classification problems [19]. The authors have built a fine-tuned, multi-layered custom CNN, to be trained on the feature vectors for both datasets.

An important point to note here would be that flattened images are not passed into a deep learning model because flattening the images means manually taking away potential meaning from the data that the neural network might be able to utilize while training itself [20]. Thus, the feature vector that was earlier used in this research to train the Support Vector Machine classifier cannot be fed directly into the CNN model that the authors have created.

This issue is addressed by storing the 64X64 pixel map for each image in its original numpy 2d-array format and dumping information for all the images in a pickle file. A pickle file can be easily created by utilizing the pickle library in Python and its unique standout feature is that it saves the information dumped into it without making any changes at all to the innate structure of the data object being dumped. Thus, the structure of the multidimensional arrays stored in it isn't altered at all and can be quickly read and fed into the CNN model that we discussed above.

The CNN models that were implemented in this research identified 8 million trainable parameters on the FEI dataset, from which CNN will select most accurate and efficient parameters by its own dynamic feature reduction.

Similarly, almost 130 million trainable features that the CNN had identified, show that the FERET dataset is indeed a very large face dataset and with this, the authors would like to share other results of their probings with the reader of this research work.

4 Results

The two datasets were simultaneously put through the process of this research and the respective results obtained by the above discussed algorithms were:

- (A) **FERET DATASET**—Two different classification models were implemented on this dataset. PCA achieves subspace reduction by eliminating features with lesser weight and mapping two or more features' information into a combined single feature [21]. Based on analysis of previous literature in the field, the size of the FERET dataset w.r.t number of features, size of dataset and lack of uniformity in amount of trainable information available for each individual class, the authors arrived upon the conclusion that implementing SVM classification on a dataset as large as FERET would be an exercise that is as futile as is wasteful. Before interpreting the following results, please note that the training of the CNN model had to be stopped mid-way (at training accuracy of 84%) due to lack of GPU resources and maxing out of Google Colab free

compute time and RAM allotment. Following are the evaluation metric and training graph report of the PCA + SVM CNN for this dataset (Fig. 14).

(B) FEI DATASET

FEI was put through the same process but with better hypertuning due to pliability of a smaller dataset. Figure 15 shows performance of various iterations of PCA with varying degrees of information retention when tested against the common SVM classifier with same parameters for each PCA iteration.

Following are the results achieved by applying SVM, PCA + SVM(max. Acc.) and CNN classification models on the FEI dataset (Fig. 16):

Finally Table 5 is shown here to give a cumulative essence of the models implemented in this research work and their respective results. Error values are calculated using Mean Average Error and Root Mean Square Error. Loos function has also been shown for the CNN models to provide a better understanding. Results from the work of this thesis can be easily compared in the given format and then conclusions that follow ahead will be easier to follow (Table 3).

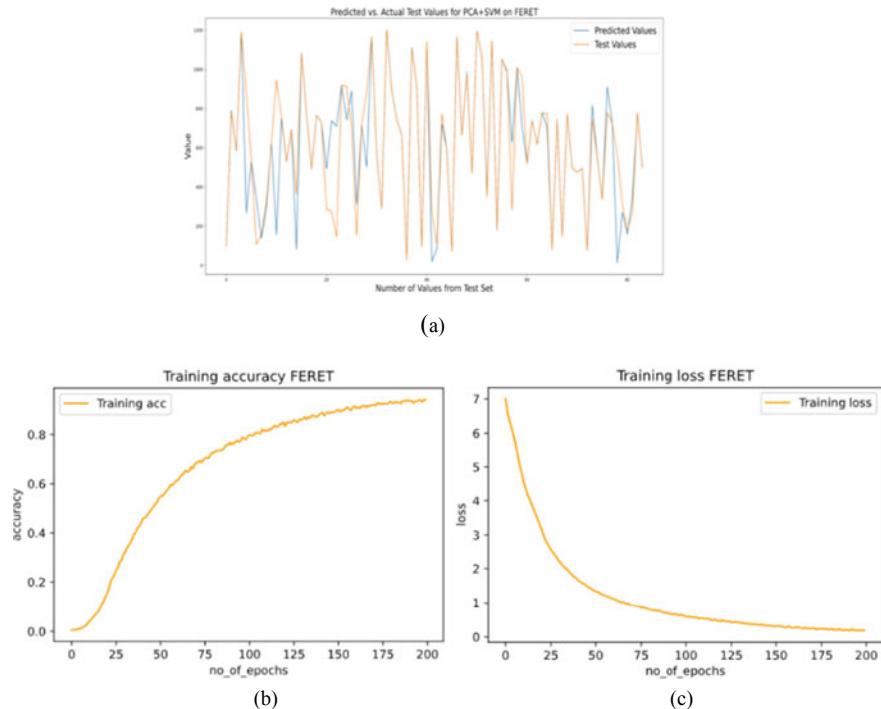


Fig. 14 Results for FERET **a**—Predicted values and test values for PCA + SVM, **b**—Training accuracy CNN, **c**—Training loss

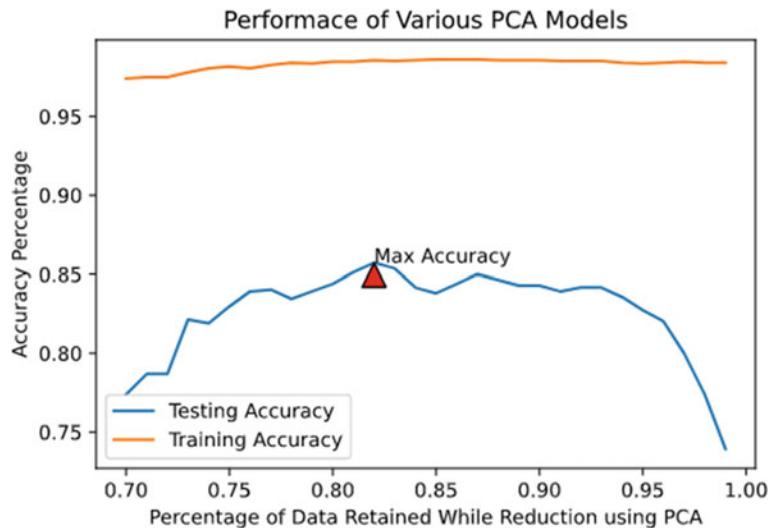


Fig. 15 Performance of PCA iterations

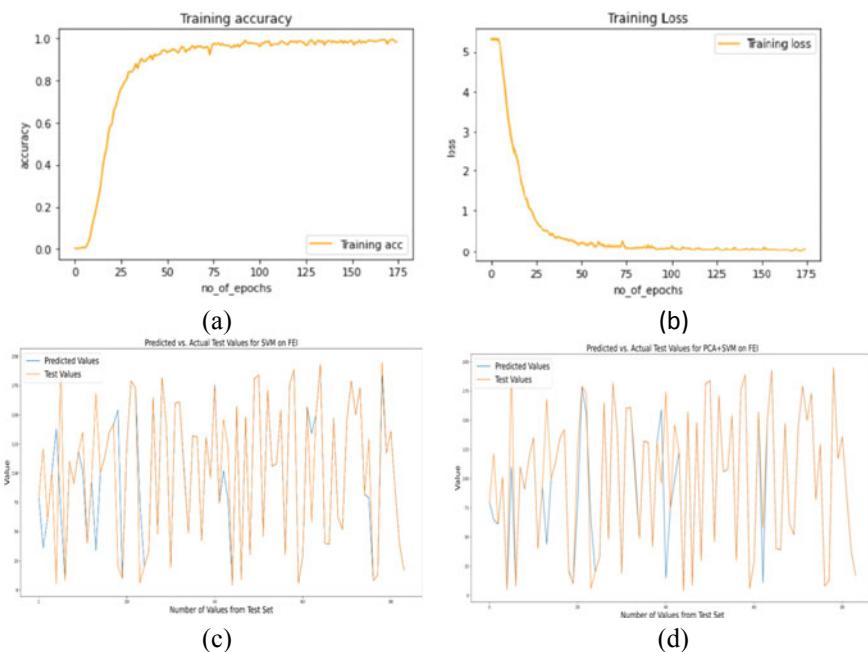


Fig. 16 FEI results: **a** Training accuracy on CNN, **b** Training loss on CNN, **c** Predicted and test values by SVM **d** Predicted and test values by PCA + SVM

Table 3 Compilation of results from 5 models

Model/Dataset	FERET			FEI		
	Training accuracy	Error	Testing accuracy	Training accuracy (%)	Error	Testing accuracy
SVM	NA	NA	NA	97.95	MAE: 6.866 RMSE: 24.801	85.47%
PCA + SVM	99.84%	MAE: 67.633 RMSE: 167.369	60.1%	96.61	MAE: 7.717 RMSE: 26.151	86.54%
CNN	84%	–	74.75	97		93.00

5 Conclusion and Future Scope

Work on this research started with an aim to analyze, derive insights from and improve the current facial recognition technologies by performing comparative analysis on two varying datasets. The authors were able to successfully implement the three different classification models on both the datasets and achieve exceedingly favourable and statistically significant results from SVM + PCA and CNN implementations on the FEI dataset. Yet, this improvement in accuracy scores would be futile until mitigation of the limitations of this technology that were observed during the course of this research.

The FERET and FEI face datasets both are localized datasets with less diversity and their scope is limited to the region where they were created (USA and Brazil). Further, as posited earlier, the datasets aren't exactly uniform when it comes to amount of trainable information available for each individual class. Analysis of classification reports rendered for the CNN model on FERET dataset revealed instances of "Support" value being 0 for a considerable amount of classes which signifies their complete absence from the testing set.

Dealing with raw image data is very computationally expensive and the current methods of subspace reduction aren't as efficient and robust at feature selection as would be required to make digital facial recognition a convenient and reliable application. Considering the discussion around biometric applications of facial recognition that were discussed at the beginning of this work it can be concluded that without the elimination of above mentioned limitations, dependable and accurate facial recognition using artificially intelligent machines would continue to be a costly exercise in vain unless assisted by hardware technologies like infrared or laser (as in the case of Apple's successful Face ID system).

Even in the face of this knowledge, the authors found significant future scope in this field of research. Beginning with the creation of comprehensive face datasets with diverse images from all over the world there turned out to be significant room for

improvement and augmentation of feature extraction methods. These changes might be in conjunction with PCA methodology or adopt an approach entirely different from it. But it can be said with surety that there is much requirement and scope to facilitate accurate and efficient feature extraction and reduce the noise in the image dataset. Such an improvement in feature extraction methods would not only further the interests of future face recognition research but would also aid all kinds of computer vision applications that deal with classification of any type of raw image data. Lastly, it is concluded that while subspace rejection methods combined with support vector classification can provide promising results in case of a smaller and highly controlled face dataset like FEI. But without significant improvement in the technology behind them, it is practically impossible for this approach to match or supersede the deep learning methods (like CNN) when it comes to large, semi-controlled datasets like FERET.

References

1. Bruce V, Young A (1986) Understanding face recognition. *Br J Psychol* 77:305–327. <https://doi.org/10.1111/j.2044-8295.1986.tb02199.x>
2. Phillips PJ, Wechsler H, Huang J, Rauss P (1998) The FERET database and evaluation procedure for face recognition algorithms. *Image Vision Comput* J 16(5):295–306
3. <https://fei.edu.br/~cet/facedatabase.html>
4. Mudunuri SP, Venkataraman S, Biswas S (April 2019) Dictionary alignment with re-ranking for low-resolution nir-vis face recognition. *IEEE Trans Inf Forensics Secur* 14(4):886–896. <https://doi.org/10.1109/TIFS.2018.2868173>
5. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: a literature survey. *ACM Comput Surv* 35(4) (December 2003):399–458. <https://doi.org/10.1145/954339.954342>
6. Parisa Beham M, Mohamed Mansoor Roomi S (2013) A review of face recognition methods. *Int J Pattern Recogn Artific Intell*
7. Hassaballah M, Aly S (2015) Face recognition: challenges, achievements and future directions. *IET Comput Vis* 9:614–626. <https://doi.org/10.1049/iet-cvi.2014.0084>
8. Tolba AS, El-baz AH, El-harby AA (2006) Face recognition: a literature review. *Int J Signal Process*
9. Mohamad D, Meethongjan K (2007) A summary of literature review: face recognition. In: Postgraduate annual research seminar (PARS' 07). 3–4 July, 2007, UTM, Johor Bahru 34
10. Phillips PJ, Moon H, Rizvi SA, Rauss PJ (2000) The FERET evaluation methodology for face recognition algorithms. *IEEE Trans Pattern Anal Machine Intell* 22(2000):1090–1104
11. Navarrete P, Ruiz-del-Solar J (2002) Analysis and comparison of Eigenspacebased face recognition approaches. *Int J Pattern Recogn Artif Intell* 16:817–830
12. Draper B, Baek K, Bartlett MS, Beveridge JR (2003) Recognizing faces with PCA and ICA, *Comput Vis Image Understand* (Special Issue on Face Recognition) 91, 115–137. Jambor WS, Draper BA, JR. B
13. Sacks O (Feb. 1996) An anthropologist on Mars. First Vintage books edition, USA
14. Komura D, Nakamura H, Tsutsumi S, Aburatani H, Ihara S (2005) Multidimensional support vector machines for visualization of gene expression data. *Bioinformatics* 21(4):439–444. <https://doi.org/10.1093/bioinformatics/bti188>
15. Beveridge R, She K, Draper B, Givens G (Dec 2002b) Parametric and non-parametric methods for the statistical evaluation of human ID algorithms. IEEE third workshop on empirical evaluation methods in computer vision, Kauai, HI

16. Moghaddam B (2002) Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Trans Pattern Anal Machine Intell* 24:780–788
17. Moon H, Phillips PJ (2001) Computational and performance aspects of pca-based face-recognition algorithms. *Perception* 30(3):303–321. <https://doi.org/10.1088/p2896>
18. Lawrence S, Giles CL, Tsoi AC, Back AD (Jan 1997) Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw* 8(1):98–113. <https://doi.org/10.1109/72.554195>
19. Hu G, Yang Y, Yi D, Kittler J, Christmas W, Li SZ, Hospedales T (2015) “When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition”. Proceedings of the IEEE international conference on computer vision (ICCV) Workshops, pp 142–150
20. Agarwal MV (2021) A study on image analysis and recognition using learning methods: cnn as the best image learner. In Khanna A, Gupta D, Pólkowski Z, Bhattacharyya S, Castillo O (eds) Data analytics and management. Lecture Notes on Data Engineering and Communications Technologies, vol 54. Springer, Singapore. https://doi.org/10.1007/978-981-15-8335-3_3
21. Thomaz CE, Giraldi GA (June 2010) A new ranking method for principal components analysis and its application to face image analysis. *Image Vis Comput* 28(6):902–913

Early Prognosis of Acute Myocardial Infarction Using Machine Learning Techniques



Abhisht Joshi, Harsh Gunwant, Moolchand Sharma, and Vikas Chaudhary

Abstract A cardiovascular complication, such as a heart attack, is among the most serious contemporary health issues to confront. People diagnosed with acute myocardial infarction (AMI) have a considerably higher risk of dying in the first year after their diagnosis. People in more and more parts of the globe are being diagnosed with myocardial infarction (MI). A report by the CDC states that statistically, someone dies every 36 s from a heart attack, and the cause of three out of four of those fatalities is a heart attack. Accurate prediction of long-term negative outcomes after an AMI may assist decide the amount of care delivered and form part of informed patient choice-making and emerging approaches that hold up the possibility of extracting new information from the present data. In this paper, we have applied different machine learning algorithms on the myocardial infarction complications database which can process and incorporate an exponentially greater number of variables and identify the intricate correlations between risk factors and ultimate outcomes. Ridge classifier and SVC have the best F1 score of 90.29% on test data which would help in the early diagnosis of acute MI. These advancements will make risk factor identification far more valuable for myocardial infarction prediction, and medical experts will concentrate on the key variables selected by the machine learning model.

Keywords Acute myocardial infarction (AMI) · Myocardial infarction (MI) · Machine learning · Mortality · Machine learning classifier

A. Joshi

Information Technology, Maharaja Agrasen Institute of Technology, GGSIPU, Delhi, India

H. Gunwant · M. Sharma (✉)

Computer Science and Engineering, Maharaja Agrasen Institute of Technology, GGSIPU, Delhi, India

e-mail: moolchand@mait.ac.in

V. Chaudhary

Computer Science and Engineering Department, JIMS Engineering Management Technical Campus (JEMTEC), Greater Noida, India

1 Introduction

A myocardial infarction (MI), often known as a heart attack, happens when blood supply to a heart region is reduced or stopped, resulting in damage to the heart muscle. All cardiovascular diseases (CVDs) are linked to the greatest fatality rate: myocardial infarction [1]. Myocardial infarction occurs when blood flow is reduced, causing damage to the heart muscle. Chest pain or discomfort is the most prevalent symptom, affecting the shoulder, arm, back, neck, or jaw. Cardiovascular illnesses are the leading cause of mortality worldwide, with heart attacks (myocardial infarction) and strokes accounting for four out of every five CVDs fatalities [2]. As novel treatment options have been used over the previous several decades, we now have a better knowledge of the etiology of myocardial infarction. So, we can use novel treatments to assist the patient while also helping to increase survival. Through time, the healthcare system has seen a significant transformation, from passive healing of the infarction via weeks of bed rest to immediate release generally within two to three days after the infarction has occurred. Nonetheless, though, obstacles persist. Even though cardiogenic shock can result in a patient's death within a month, mortality is still rather high, with around a 40% chance of dying in 30 days [4].

There are currently no studies looking at predicting MI incidence time (i.e., the age of a MI), which is important for preoperative risk assessment [3]. As a result, predicting when a MI may develop is nearly difficult. As a result, early detection of MI and information on its occurrence time (such as a longer treatment time) would allow for more prompt treatment, improve patient outcomes, and lower the global rise in CVD mortality and MI deaths. The course of the illness in individuals with MI differs from the other individuals. Therefore, a benign indication for medical imaging should not be equated with an absence of complications, nor should a worsening problem be seen as a precursor to a MI. Acute and subacute complications of illness occur in half of the patients, and those problems are responsible for the deterioration of the disease. Those problems are also responsible for the patient's mortality [14]. Even the most seasoned professionals are not always able to anticipate these issues developing. To perform this work, it is essential to know the risk of myocardial infarction complications to detect it and then take all of the required procedures to rescue the patient.

In this paper, multiple machine learning algorithms and libraries are used to extract the important features from the dataset (i) at the time of admission of the patient and (ii) on the third day of the hospital period of the patient. This will help determine the important factors on day one and day three that contribute most toward predicting the cause of the death. Various machine learning models like random forest, decision trees have been used to check the credibility of the selected features. To check the credibility, we have used the F1 score. We used F1 score instead of accuracy to compare the model used when the false negatives and false positives are crucial. In our case, false negative, and false positive play a more important role than true positive and true negative. In medical-related issues, false negative and false positive are important as telling people the wrong result might scare the patient, thus worsening

the project. In most real-world classification tasks, there is an imbalanced class distribution. Hence, F1 score is a superior statistic to use for evaluating this sort of distribution.

The critical aspects of the technique proposed are as follows:

- Machine learning classifiers such as random forest, decision tree, K-nearest neighbor, ridge classifier, and SVC were utilized.
- Ridge classifier and decision tree earned the best F1 score for the training dataset among the various classifiers; both classifiers could attain a perfect 100 F1 score.
- Ridge classifier and SVC both earned the highest F1 score of 90.29 in the test dataset.
- The following are important characteristics that aid in the diagnosis of MI:
 - (i) Diastolic blood pressure ('D_AD_ORIT')
 - (ii) Systolic blood pressure ('S_AD_ORIT')
 - (iii) Cardiogenic shock at the time of admission ('K_SH_POST')

Section 2 of the paper consists of the current study followed by the proposed methodology, data collection, and analysis. Finally, the fourth section contains the results and discussions, followed by the conclusion and future scope.

2 Literature Review

Since the death rate from myocardial infarction (heart attack) is growing, this issue has become a serious worry for both emerging and developed nations, especially persons in their younger years. An electrocardiogram has been used in almost all cases of hearts found to have had a heart attack (ECGs). CVDs such as MI are an area where researchers recognize the usefulness of ECG signals, particularly because of the significance of ECG findings in determining if an individual has a CVD [6].

Most of the strategies to identify MI have been devoted to an abnormality in the morphology of EKG signals. The approach employed in these papers utilizes several preprocessing processes, such as identifying ECG complexes and hand-crafting characteristics to extract learning opportunities from it [6, 7]. The creation of these models has shown to be quite time-consuming; it also requires model development. Comparison of how deep learning and machine learning models compares to a logistic regression baseline model with just risk variables that are known are applied to harmonized EHR data for predicting incident myocardial infarction (MI) [5, 8]. Deep learning models eliminate the feature engineering stages like preprocessing, saving time by removing them. They automatically save time in the long run since they work without the need for feature engineering. Many researchers have looked at using deep networks to diagnose cardiovascular problems based on ECG [9, 18]. The most widely used and reliable and, most of the time, best performing deep learning architectures are considered CNNs, e.g. [10, 13, 17]. Not much research has used recurrent neural network, but most have had little effect [21].

Further development of the model is needed. The combination of CNN and LSTM has proven to be better than either CNN or LSTM in isolation [15, 18]. However, with all of the tremendous promise of deep learning, results are only currently within reach for state-of-the-art performance as training of the model takes tremendous time. An important factor when trying to build a new type of architecture from scratch is that each of these structures has several parameters that must be learned. For that, many researchers are working on transfer learning techniques that have been used in [12, 19].

Employing Google's inception image recognition algorithm revealed the usefulness of transfer learning for CVD diagnosis. [20]. Heart ECG signals were processed as pictures to be analyzed by the network, resulting in an equivalent to the type of analysis performed by cardiologists. ECG-based MI detection has been studied recently using end-to-end deep learning techniques [11]. The majority of these cases treated MI detection as a binary classification problem (non-MI or MI). In contrast, MI localization was the primary goal in others, using a multiclass problem formulation. The latter was concerned with detecting various MI types based on the location of the occluded artery [16]. In addition, the current electrocardiogram [ECG] was given a deep MI technique. Deep MI uses fusion methods to distinguish MI from normal situations and determine when it occurs. The longitudinal information in ECGs is encoded using RNN. Deep MI also does feature extraction using transfer learning and offers the flexibility of multi-lead ECG fusion [1].

All major work regarding MI was done using ECGs. In contrast, the dataset that we have used has additional information in the inclusion of ECG parameters, making it important as additional factors would contribute to the prediction's betterment, which would improve the model performance. It will also help the specialist doctor to save the life of a patient.

3 Proposed Methodology

In this paper, Fig. 1 shows the methodology adopted by us for finding out the accurate detection of MI. The below points show an explanation of every step taken.

- (i) Initial steps involve preprocessing the data, which includes filling the missing values with the median value of each column values and removing the complete columns whose missing values were above 400 as they might alter the result's accuracy.
- (ii) After cleaning the data bar graph, the 'LET_IS' column was plotted to analyze the number of deaths, and the cause of the death as the last column tells whether the patient is alive or dead, and if dead, then what was the cause of the death.
- (iii) After analyzing the columns, we dropped the columns mentioned in the dataset description. For example, the columns 2–112 can be used as input data for prediction, but at the time of admission on day 1, all input columns (2–112)

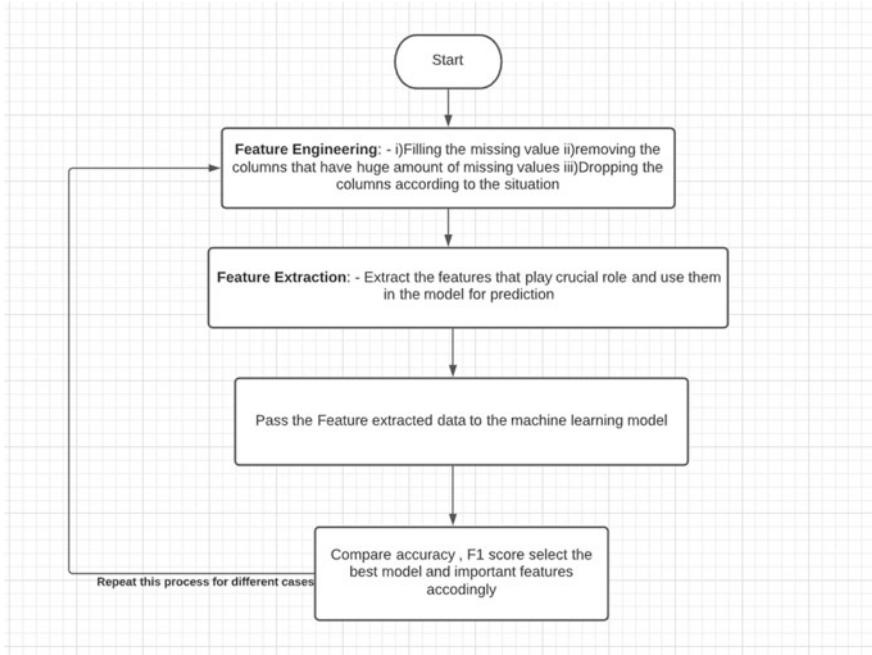


Fig. 1 Flowchart of the proposed system

- except (93, 94, 95, 100, 101, 102, 103, 104,105) can be used for prediction, so we accordingly deleted the columns as per the given instructions.
- (iv) After removing the columns, the data was split into training and testing so that the problem of overfitting can be avoided. In our case, we have divided the ratio of the train-test into 80:20.
 - (v) After that, 20 most important features were selected using mutual information compared to the output variable, which would contribute toward the final result. Mutual information estimate mutual information for a discrete target variable. A non-negative value that measures the interdependence between two random variables called mutual information. It is zero if two random variables are independent, while larger values indicate greater interdependence. Non-parametric approaches based on entropy estimates from K-nearest neighbor distances are used in this function.
 - (vi) This paper uses five different machine learning model that performs feature selection on our data, and we compare their efficiency by calculating the accuracy as well as by calculating the F1score, which is defined as $2*((\text{precision}*\text{recall})/(\text{precision} + \text{Recall}))$.

Precision is defined as $\text{TP}/(\text{TP}+\text{FP})$

And recall is defined as $\text{TP}/(\text{TP}+\text{FN})$

FP = False Positive

FN= False Negative

TN= True Negative

TP= True Positive

3.1 Machine Learning Classifier Used

The different machine learning classifier used was mentioned below:

3.1.1 K-Nearest Neighbor

K-nearest neighbor is a machine learning technique that may be utilized for Regression and classification. Another useful feature of this tool is that it uses predefined learned labels for the new dataset and classifies using the new sample. The first step in finding the distance between the new sample, and all the predefined trained points calculates the Euclidean distance. Next, we choose the value of K for which the data is closest to the training points. Finally, we use the most common sample as the solution to our classification issue. Choosing the proper value of K was done by attempting multiple K values and selected the most effective one in our situation, and it turned out to be 8.

3.1.2 Random Forest Classifier

Using a random forest is a versatile, easy-to-use machine learning approach that gives, even without hyper-parameter adjustment, a fantastic result most of the time. It is also one of the most commonly used algorithms due to its simplicity and broad application. A random forest classifier is a meta estimator that fits many decision tree classifiers on different sub-samples of the dataset and utilizes an average strategy to increase predicted accuracy while also controlling overfitting. The size of the sub-sample is regulated by a parameter called max samples. Random forest is an ideal machine learning tool for most classification and regression tasks. Similar to a decision tree or bagging classifier, the random forest also contains virtually the same hyperparameters. However, there is no need to utilize a bagging classifier, as the random forest is sufficient on its own. Random forest is useful for regression and classification problems since you may use the algorithm's regression or classification regressor. Growing the trees introduces extra unpredictability into the model, so a random forest is a form of ensemble learning. Rather than sifting through many features to find the most significant feature, the feature finder sorts through a random selection of features and finds the greatest feature. Thus, it encourages the production of a broader range of findings, which yields a better model.

3.1.3 Decision Trees Classifier

A decision tree is one of several supervised learning methods. Regression and classification issues may also be solved using the decision tree approach. The goal of using a decision tree is to create a training model that will use the discovered fundamental decision rules to identify the class or value of the target variable (training data). When using decision trees to forecast a record's class label, we start at the top of the tree. The value of the root property is compared to the value of the record's attribute. We make a decision based on comparison and then go to the next node. A decision tree classifier operates by dividing the dataset into smaller datasets until the dataset cannot be divided further or the target variable is equal. It first allocates all the training examples to the root of the tree. Then it uses the criteria to partition the values based on that feature (In the decision tree algorithm, Gini index and information gain methods are used to calculate these nodes). Nodes are partitioned on criteria and a threshold, after which the partitioned values are termed nodes. Those partitioned values are then split again, and the process is continued until the tree achieves full height or is aborted.

3.1.4 Ridge Classifier

Ridge classifier implements ridge regression as a classifier-making strategy, and Ridge regression methodology is distinct for binary and multiclass classification. The objective of binary classification is to make predictions on our target variable using a ridge model. For example, this function produces the following values: MSE + l2 penalty. If the ridge regression predicts that the class is positive, it predicts that it is positive. If, however, the prediction is that class is negative, it predicts that class is negative. The problem faced in multiclass classification is that Label Binarizer generates a multi-output regression. Then the separate ridge regression models are trained to classify each class (One-Vs-Rest modeling). After that, it offers predicted values for each class in ridge regression model form (and presents them in the form of an actual number for each class) and uses argmax to calculate class predictions.

3.1.5 SVC (Support Vector Classifier)

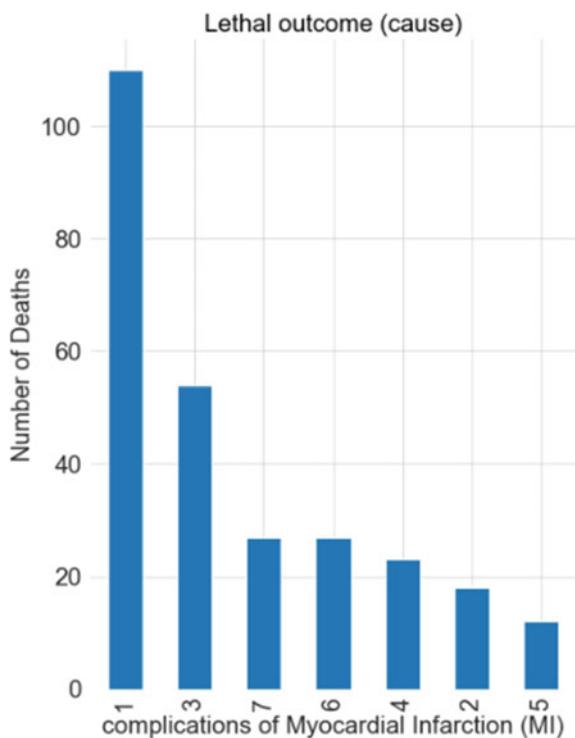
The goal of a linear SVC (support vector classifier) is to classify or split your data according to your specifications, which allows you to select the best-fitting hyperplane for your requirements. You will use the hyperplane that you just created to get there, and then you can feed the resulting features to your classifier to see what the projected class is. SVC is a non-parametric clustering technique that makes no assumptions about the number of clusters in the data ahead of time. Therefore, it is best for data with a low number of dimensions, as our case values are less than 3000. A linear SVC aims to fit the train data you have provided and produce the best fit according to the data hyperplane that divides or categorizes your data accordingly.

3.2 Dataset and Its Analysis

The UCI repository's Myocardial Infarction Complications Database was used. (<https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications>).

This dataset was provided on 2020-12-09. The dataset contains 1700 records (rows) and 124 columns, of which 110 columns are used as input data for prediction, and possible complications have been listed in columns from 113 to 124. The 2nd through 112th columns of the dataset are utilized as input. The potential for difficulty may occur at any one of four possible time moments: the foundation of the facts and information already known. Patients who have been admitted to the hospital before coming to the ER can utilize all the input columns (2–112) except 93, 94, 95, 100, 101, 102, 103, 104, and 105 for prediction. Except for 94, 95, 101, 102, 104, and 105, the data in all input columns can make predictions at the end of the first day after admission to the hospital. The ability to utilize 2–112 for prediction will have been increased to 95, 102, and 105 after two days (after admission to the hospital), except 95 and 102, which could not be used as input for prediction during the second day the hospital. Prediction can commence once all input columns (2–112) have been used for prediction at the end of the third day (72 h after admission to the hospital). Figure 2 shows the no. of deaths due to different complications of MI.

Fig. 2 Graph for number of deaths due to different myocardial complication



As shown in Fig. 2, the total myocardial complication that has caused most deaths are in order:

- (i) Cardiogenic shock
- (ii) Myocardial rupture
- (iii) Asystole
- (iv) Ventricular fibrillation
- (v) Third-degree AV block
- (vi) Pulmonary Edema
- (vii) Dressler syndrome.

Faced with hard challenges, one must do a thorough evaluation and comparison of various data mining and pattern recognition algorithms. Two practically important problems that could be solved with the use of the database are:

- (i) On the first day of admission and after the third day in the hospital, it is quite tough to accurately predict the likelihood of complications in patients with myocardial infarction.
- (ii) They are developing data that healthcare providers may use to provide a plan of care for these issues. In addition to many basic categories of activities, several fundamental actions must be done for an outcome to be measurable (cluster analysis). Concerning all of the above problems, the dataset was employed to tackle them.

3.3 *Experimental Setup*

The system was built on a computing environment employing mobile computers. The system on which our laptop is based is an Eighth-generation Intel Core i5 CPU that has a 2.3 GHz clock speed and 2304 MHz, 4 Core(s), 8 Logical Processor 4 MB cache memory, and 8 GB of DDR4 RAM installed on a 1 TB hard drive and an additional 128 GB of SSD (Ver. 10.0.19041 build 19,041, 64 bit). Python 3.2 with packages such as NumPy, Scikit-learn, and Pandas was used on the Jupyter notebook provided by the Anaconda.

4 Result and Discussion

At the time of admission of the patient, we can see in Figure 3, five most significant complication that decides the patient's survival are in the order: -

- (i) **Myocardial Rupture (RAZRIV):** Lacerations of the heart's ventricles or atria, the interatrial or interventricular septum, or the papillary muscles are all considered lacerations.

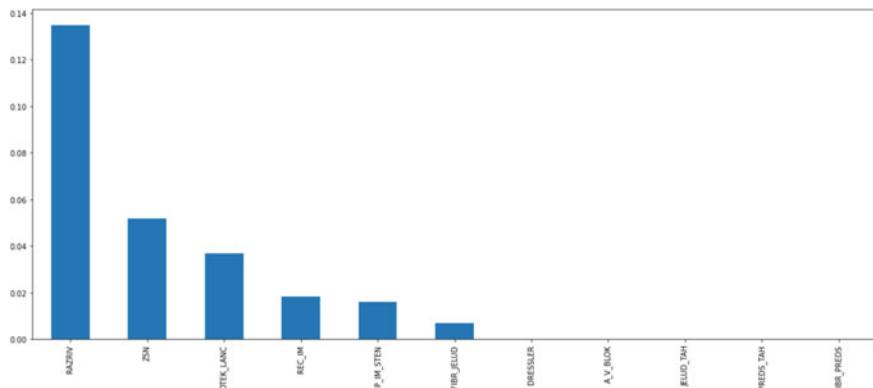


Fig. 3 Complications of myocardial infarction (MI) at the time of admission (Day 1)

- (ii) **Chronic heart failure (ZSN):** When your heart muscle is not pumping blood as efficiently as it should, this happens. Certain conditions, such as narrowed arteries in the heart (coronary artery disease) or high blood pressure, lead your heart to weaken or stiffen over time, making it difficult to fill and pump.
- (iii) **Pulmonary Edema (OTEK_LANC):** Excess fluid in the lungs causes this illness. This fluid gathers in the lungs' many air sacs, making breathing harder. Heart issues are the most common cause of pulmonary edema.
- (iv) **Relapse of the myocardial infarction (REC_IM):** It is deterioration in someone's state of health after a temporary improvement by recurring of the myocardial infarction. Recurrent myocardial infarction is common after a first MI and is linked to a higher risk of morbidity and mortality.
- (v) **Post-infarction angina (P_IM_STEN):** Post-myocardial infarction syndrome is a condition that occurs after a heart attack. Repeated or protracted symptoms characterize it. Fever, chest discomfort, and clinical and laboratory tests are all common symptoms. In addition, pericarditis, pleurisy, and pneumonitis are all present.

On the 3rd day after being admitted to the hospital, Figure 4 shows important complication that determines the diagnosis of MI are in order:

- a. **Ventricular tachycardia (JELUD_TAH):** A fast, irregular heart rhythm is known as ventricular tachycardia. It begins in the ventricles, the lower chambers of your heart. Three or more heartbeats in a row, at a pace of more than 100 beats per minute, are considered VT. If VT persists for more than a few seconds at a time, it can be fatal.
- b. **Post-infarction angina (P_IM_STEN):** The post-myocardial infarction syndrome. It is characterized by prolonged or recurrent. Fever, chest pain, and clinical and laboratory. Evidence of pericarditis, pleurisy, and pneumonitis.

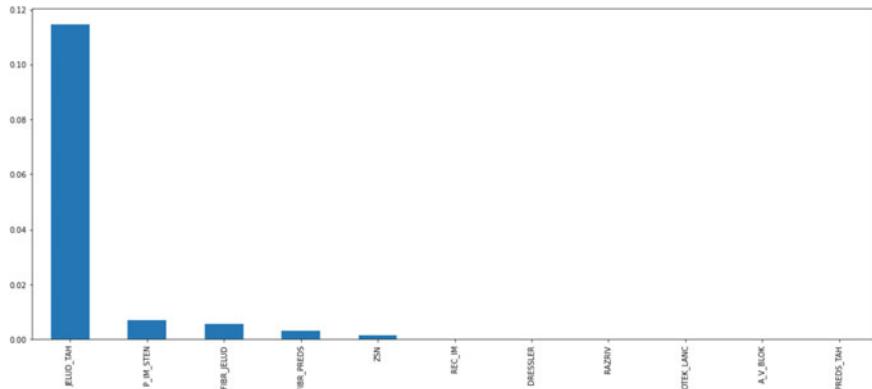


Fig. 4 Complications of myocardial infarction (MI) at the end of the 3rd day

- c. **Ventricular fibrillation (FIBR_JELUD):** Ventricular fibrillation (V-fib or VF) is a kind of irregular cardiac rhythm in which the heart's ventricles quiver rather than pump regularly. A jumble of electrical activity causes it. Ventricular fibrillation causes cardiac arrest, characterized by loss of consciousness and the absence of a pulse.
- d. **Atrial fibrillation (FIBR_PREDS):** Atrial fibrillation is an irregular and often rapid heart rate that occurs when the two upper chambers of your heart experience chaotic electrical signals. The result is a fast and irregular heart rhythm
- e. **Chronic heart failure (ZSN):** When your heart muscle is not pumping blood as efficiently as it should, this happens. Certain conditions, such as narrowed arteries in the heart (coronary artery disease) or high blood pressure, lead your heart to weaken or stiffen over time, making it difficult to fill and pump.

4.1 Important Feature from the Input Column

To select the most important feature, we plotted a graph as shown in Fig. 5. From this, we have selected 20 most important features that might predict the final outcome of the patient and they are:

'AGE', 'INF_ANAM', 'ZSN_A', 'np_09', 'zab_leg_01', 'S_AD_ORIT', 'D_AD_ORIT', 'O_L_POST', 'K_SH_POST', 'ant_im', 'inf_im', 'ritm_ecg_p_01', 'ritm_ecg_p_08', 'n_r_ecg_p_08', 'fibr_ter_05', 'fibr_ter_08', 'GIPO_K', 'ROE', 'TIME_B_S', 'NITR_S'.

From these three most important features as we see from the graph are:

- (i) Systolic blood pressure on the report by ICU ('S_AD_ORIT')
- (ii) Cardiogenic shock on the report by ICU ('K_SH_POST')
- (iii) Diastolic blood pressure on the report by the ICU ('D_AD_ORIT').

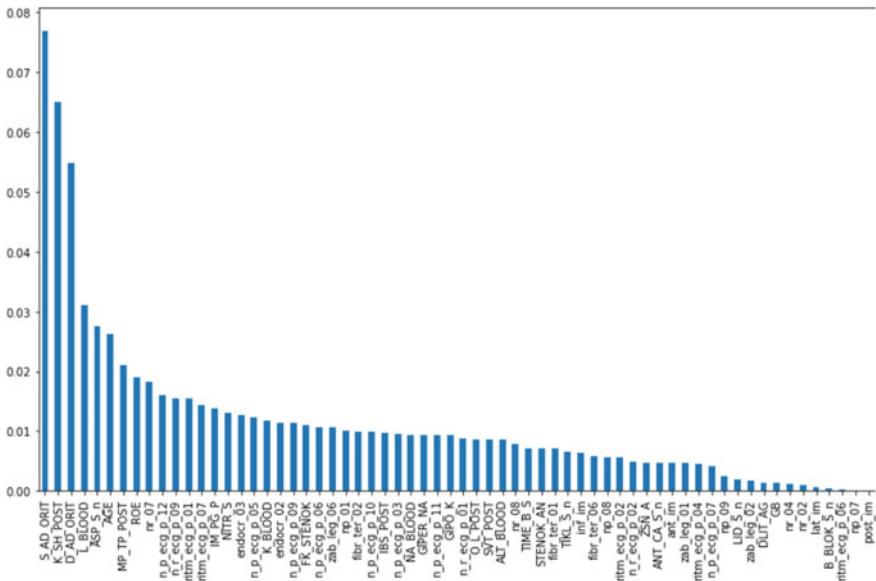


Fig. 5 Important features of MI at the time of admission of the patient (Day 1)

Most 20 important feature were selected at the end of the 3rd day after myocardial infarction, these features as shown in Fig. 6 are:

'AGE', 'SEX', 'STENOK_AN', 'np_01', 'np_04', 'S_AD_ORIT',
'D_AD_ORIT', 'K_SH_POST', 'MP_TP_POST', 'IM_PG_P', 'ritm_ecg_p_01',
'ritm_ecg_p_02', 'n_r_ecg_p_08', 'n_p_ecg_p_11', 'fibr_ter_01',
'fibr_ter_06', 'K_BLOOD', 'ALT_BLOOD', 'NITR_S', 'ASP_S_n'.

From these three most important features as we see from the graph are:

- (i) Diastolic blood pressure on the report by the ICU ('D_AD_ORIT')
 - (ii) Systolic blood pressure on the report by ICU ('S_AD_ORIT')
 - (iii) Cardiogenic shock on the report by ICU ('K_SH_POST').

here the three features are same as that on 1st day but the order is different
'S AD ORIT' is replaced by **'K SH POST'**.

In Figs. 7 and 8, for easy understanding, all the F1 scores were multiplied by 100. From Fig. 7, from the F1 score on training data, we can conclude that Decision Tree and Ridge Classifier have outperformed well. However, in Fig. 8, which compares the F1 score of test data done during testing for different machine learning classifiers, ridge classifier and SVC have the best F1 score. It means that during testing and validation, ridge classifier outperformed all the other classifiers, which would help further in the early diagnosis of MI.

F1 score of the test Data:

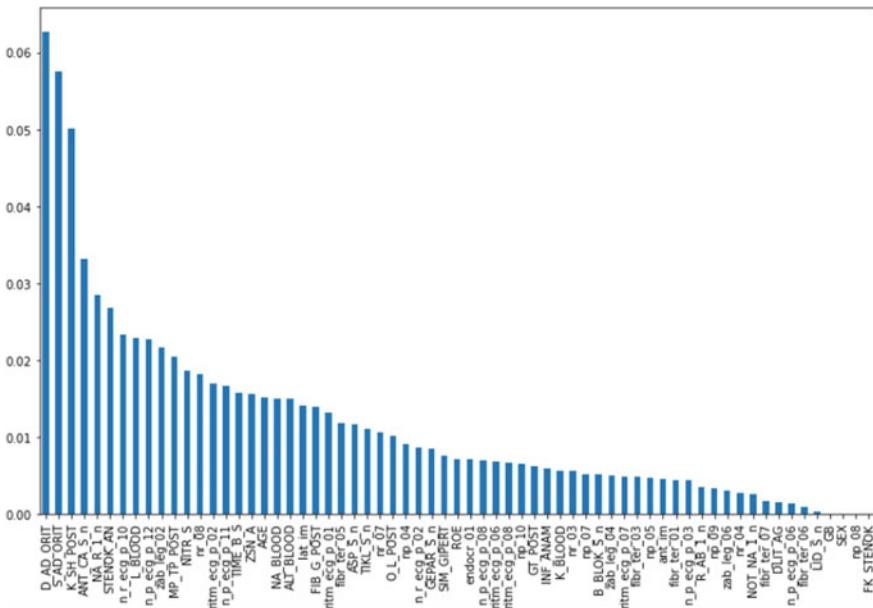


Fig. 6 Important features of MI at the end of the third day



Fig. 7 F1 score comparison during testing for different machine learning algorithms

5 Conclusion and Future Scope

The research and comparison show that the most important complication to consider is myocardial rupture (RAZRIV). In addition, diastolic blood pressure is the most important feature on the report by the ICU ('D_AD_ORIT'). On the third day, the most important feature remains the same as that of day one, but we saw a change in

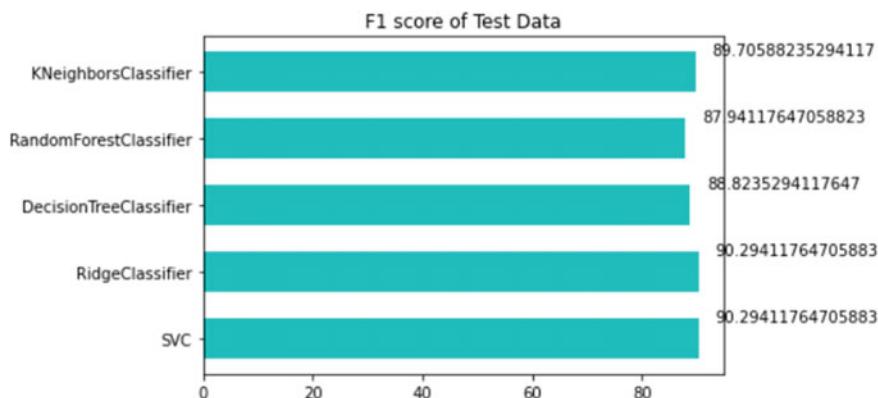


Fig. 8 F1 score comparison during testing for different machine learning algorithms

complication. Instead of RAZRIV, it is ventricular tachycardia (JELUD_TAH). So, when a patient is admitted, these characteristics should be considered since they are crucial in the prognosis of acute MI and its complications. Among the expert systems used, K-nearest neighbor, random forest, decision trees, ridge classifier and SVC, **ridge classifier** and **SVC** have the best F1 score. They both were able to achieve an outstanding F1 score of 90.29.

In the last several decades, we have made considerable advances in our knowledge and treatment of coronary atherosclerosis and associated consequences. An article from the centers for disease control and prevention mentioned that for 2014–2015, an estimated \$216 billion were spent in the US alone for heart-related issues on new medication and other treatment techniques. Early intervention is the greatest way to address the issues above, lower the incidence, and enhance results for individuals who get the latest treatments outlined above.

References

1. Tadesse G, Javed H, Liu Y, Liu J, Chen J, Weldemariam K, Zhu T (2021) DeepMI: deep multi-lead ecg fusion for identifying myocardial infarction and its occurrence-time. ArXiv, abs/2104.02054.
2. WH. Organization (2018) “Cardiovascular diseases (CVDs),” URL link: [www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](http://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)), Last accessed on 08 June 2020
3. Fleisher LA, Fleischmann KE, Auerbach AD, Barnason SA, Beckman JA, Bozkurt B, Wijesundera DN et al (2014) 2014 ACC/AHA guideline on perioperative cardiovascular evaluation and management of patients undergoing noncardiac surgery: a report of the american college of cardiology/american heart association task force on practice guidelines. Circulation 130(24):e278–e333. <https://doi.org/10.1161/cir.0000000000000106>
4. Saleh M, Ambrose JA (2018) Understanding myocardial infarction. F1000Research, 7:1378. <https://doi.org/10.12688/f1000research.15096.1>
5. Rossiev DA, Golovenkin SE, Shulman VA, Matjushin GV (1995) “Neural networks for forecasting myocardial infarction complications,” The second international symposium on

- neuroinformatics and neurocomputers, pp 292–298. <https://doi.org/10.1109/ISNINC.1995.480871>
- 6. Jahmunah V, Oh SL, Wei JKE, Ciaccio EJ, Chua K, San TR, Acharya UR (2019) Computer-aided diagnosis of congestive heart failure using ECG signals—a review. *Physica Med* 62:95–104. <https://doi.org/10.1016/j.ejmp.2019.05.004>
 - 7. Adam M, Oh SL, Sudarshan VK, Koh JE, Hagiwara Y, Tan JH, Acharya UR et al (2018) Automated characterization of cardiovascular diseases using relative wavelet nonlinear features extracted from ECG signals. *Comput Methods Programs Biomed* 161:133–143. <https://doi.org/10.1016/j.cmpb.2018.04.018>
 - 8. Mandair D, Tiwari P, Simon S et al (2020) Prediction of incident myocardial infarction using machine learning applied to harmonized electronic health record data. *BMC Med Inform Decis Mak* 20:252. <https://doi.org/10.1186/s12911-020-01268-x>
 - 9. Rahhal MMA, Bazi Y, AlHichri H, Alajlan N, Melgani F, Yager RR (2016) Deep learning approach for functional classification of electrocardiogram signals. *Inf Sci* 345:340–354. <https://doi.org/10.1016/j.ins.2016.01.082>
 - 10. Acharya UR, Fujita H, Oh SL, Hagiwara Y, Tan JH, Adam M (2017) Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. *Inf Sci* 415–416:190–198. <https://doi.org/10.1016/j.ins.2017.06.027>
 - 11. Reasat T, Shahnaz C (2017) Detection of inferior myocardial infarction using shallow convolutional neural networks. 2017 IEEE region 10 humanitarian technology conference (R10-HTC). <https://doi.org/10.1109/r10-htc.2017.8289058>
 - 12. Xiao R, Xu Y, Pelter MM, Mortara DW, Hu X (18 May 2017) A deep learning approach to examine ischemic st changes in ambulatory ecg recordings. *AMIA Jt Summits Transl Sci Proc*. 2018:256–262. PMID: 29888083; PMCID: PMC5961830
 - 13. Strodtthoff N, Strodtthoff C (2019) “Detecting and interpreting myocardial infarctions using fully convolutional neural networks.” *Physiol Measure* 40(1):015001. <https://doi.org/10.1088/1361-6579/aaf34d>
 - 14. Raghunath S et al (2019) “Deep neural networks can predict mortality from 12-lead electrocardiogram voltage data.” *arXiv preprint arXiv:1904.07032*
 - 15. Goto S, Kimura M, Katsumata Y, Goto S, Kamatani T, Ichihara G, Sano M et al (2019) Artificial intelligence to predict needs for urgent revascularization from 12-leads electrocardiography in emergency patients. *PLoS ONE* 14(1):e0210103. <https://doi.org/10.1371/journal.pone.0210103>
 - 16. Baloglu U, Muhammed T, Yıldırım Özal, Tan San R, Acharya UR (2019) Classification of myocardial infarction with multi-lead ecg signals and deep cnn. *Pattern Recog Lett* 122. <https://doi.org/10.1016/j.patrec.2019.02.016>
 - 17. Han C, Shi L (2019) ML-ResNet: a novel network to detect and locate myocardial infarction using 12 leads ECG. *Comput Methods Programs Biomed* 105138. <https://doi.org/10.1016/j.cmpb.2019.105138>
 - 18. Feng K, Pi X, Liu H, Sun K (2019) Myocardial infarction classification based on convolutional neural network and recurrent neural network. *Appl Sci* 9(9):1879. <https://doi.org/10.3390/app9091879>
 - 19. Xiao R, Xu Y, Pelter MM, Fidler R, Badilini F, Mortara DW, Hu X (2018) Monitoring significant ST changes through deep learning. *J Electrocardiol*. <https://doi.org/10.1016/j.jelectrocard.2018.07.026>
 - 20. Szegedy C et al (2015) Going deeper with convolutions. *IEEE Conf Comput Vision Pattern Recog (CVPR)* 2015:1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
 - 21. Darmawahyuni A, Nurmaini S, Sukemi Caesarendra W, Bhayyu V, Rachmatullah MN, Firdaus (2019) Deep learning with a recurrent network structure in the sequence modeling of imbalanced data for ecg-rhythm classifier. *Algorithms* 12(6):118. <https://doi.org/10.3390/a12060118>

Multimodal Biometric Authentication by Slap Swarm-Based Score Level Fusion



G. Elavarasi and M. Vanitha

Abstract According to the physical or behavioral characteristics, the identification techniques are referred using the term ‘biometrics’. One of the growing concerns is replay attacks, obfuscation and the fear of circumvention as biometric recognition becomes increasingly popular. When contrasted to unimodal biometric security systems, the lower false alarms and higher recognition rates are exhibited in which the fusion of biometrics leads to security systems. From a well-known benchmark biometrics database, the number of patterns with supervised learning is performed. From the same database, the patterns with validation/testing took place that never present in the training dataset. This chapter discussed the biometric recognition by Score Level Fusion (SLF) with distance minimization by Slap Swarm Optimization (SSO) and finally recognizing through Support Vector machine (SVM). Finally, the higher accuracy with greater efficiency is recognized in terms of fingerprint recognition.

Keywords Biometric · Optimization · Image recognition · Fusion and modeling

1 Introduction

Biometric innovation holds out the confirmation of a difficulty free, safe method to make extremely exact validations of people. It outfits a got strategy for acknowledgment that cannot be taken, lost or neglected, which is by and large continuously more needed in security environments and applications, for example, access control and electronic exchanges [1]. A huge measure of interest and broad examination has prompted the improvement of biometric information therefore. Biometrics, in the least complex definition, is the estimation of an individual utilizing the physical and social qualities. It empowers a human to be distinguished and validated through a bunch of unmistakable and obvious biometric information, like face, unique mark, iris, and voice information [2]. Every iris is novel and does not change over the

G. Elavarasi (✉) · M. Vanitha

Department of Computer Applications, Alagappa University, Karaikudi, Tamil Nadu, India

individual's lifetime, thusly making their utilization to recognize individuals works far better than fingerprinting and natural eye retina. Iris is a physiological biometric highlight and it contains an extraordinary surface which is sufficiently intricate to be utilized as a biometric signature [3]. This sort of situation may happen, for example, on the off chance that one plans to utilize biometric validation to get to advantaged assets over the Internet. Circulated biometric validation requires cross-breed conventions incorporating cryptographic procedures and example acknowledgment devices [4, 5]. Far-off client confirmation component is valuable in conveyed area to distinguish legitimacy of far-off clients. There exist a few procedures for far-off verification. Biometrics is more usable than remembrance and actual token-based techniques for validation, albeit the utilization of biometrics can present dangers as serious as for all time trading off a person's security [6]. The possibility to incorporate security and ease of use viably is more noteworthy with biometrics than with other validation strategies [7]. Biometric advances for security incorporates acknowledgment of faces, fingerprints, iris, retina, voice, signature strokes and so on Cryptography is a significant security highlight of PCs. Data in PC can be gotten by utilizing large numbers of the accessible cryptographic calculations [8]. The proficiency of proposed framework about FAR and FRR is determined by utilizing Multimodal Biometric blend programming. The main issue with these methodologies is that the discriminative ability is not significantly better in light of the fact that the proposed techniques have not used the traded data of the pre-owned pictures productively [9]. The creators have introduced two-level score level combination strategy for the consolidation of scores [10] being gotten from the drop capable layout of different biometric qualities. The structure has been run on two virtual information bases. The outcomes have shown that the proposed combination has upgraded preferred execution over the nonbiometric framework [11].

2 Literature Review

Specifically, the periocular biometrics uses authentication procedure and biometric system that was proposed by Mason et al. [12]. The patients are identified using method thereby providing healthcare system with this approach. In healthcare information systems, the electronic master patient and a new technique combine a use of particular biometrics. In our research study, various periocular biometric recognition approaches are compared. The deep learning-based methods and various traditional models are assessed.

For biometric information protection, Payal Garga and Ajit Kumar Jain in 2020 [13] proposed invisible watermarking scheme. The 2D-DWT transformations provide appropriate band coefficients and the edge entropy segments the host image into blocks. The watermarking image properly embedded by an adaptive histogram technique. Ultimately, the safety and security to biometric images are provided with visual cryptography technique (VC) and a reversible data hiding approach (RDH) to retrieve the original watermark.

For mobile devices, the continuous authentication and behavioral biometrics technologies classification are suggested by Stylios et al. in 2020 [14] in which the feature extraction techniques and behavioral biometrics collection methodologies are used. For continuous authentication, the seven types of behavioral biometrics performances are focused on machine learning models with a state-of-the-art literature review. Moreover, highlight relevant counter measures are utilized vulnerability of machine learning models against well-designed adversarial attack vectors also showed.

Padmanabhan and Radhika [15] proposed any one of the enrolled emotions given as input with the successful decryption at receiving end validates key management with correctness and efficiency are used. From facial emotions, the optimal features are selected by merging the Chicken Swarm Optimization and Deer Hunting Optimization Algorithm. The Rooster Update with Fitness Sorted Deer Hunting Optimization Algorithm is derived algorithm. From Japanese Female Facial Expression, the optimal features is extracted with neutral, surprise, sadness, happiness, fear, disgust and anger. The neural network is trained using a Yale Facial datasets.

The fingerprint texture patterns and integration of face are utilized by feature extraction and classification with the help of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) that was proposed by Kumari et al. [16]. The grayscale converts the set of training images. For each number of images, generate multiple samples applies a crossover operator. At the time of verification, if a biometric modality does not exist that ensured. For each weight of each biometric modality, work proposed here is pre-planned. The weights the threshold value calculates person can be certified to provide. Higher effectiveness security is given fast feature selection in biometric image authentication is selected.

Elavarasi et al. [17] were proposed an effective Multiple Share Creation (MSC) with Light Weight Cryptography (LWC) technique and Elephant Herd Optimization(EHO) algorithm to achieve security for biometric images. In order to examine the performance of the presented model, a set of simulations take place on iris images. Originally, MSC process is applied to produce a multiple set of shares for every applied image. Afterward, the shares are encrypted by LWC technique. For raising the effectiveness of the LWC, stream cipher is applied and the optimal key selection process takes place by EHO algorithm. The experimental results depicted that the projected technique has reached maximum security compared to other methods.

3 Methodology

From a specific physiological or behavioral characteristic, a person according to the feature vector derived a pattern-recognition system as a biometric system. The individual's biometric template with captured biometric characteristics is compared by identifying the system validates a person's in verification mode. The Support Vector Machine (SVM) performs recognition.

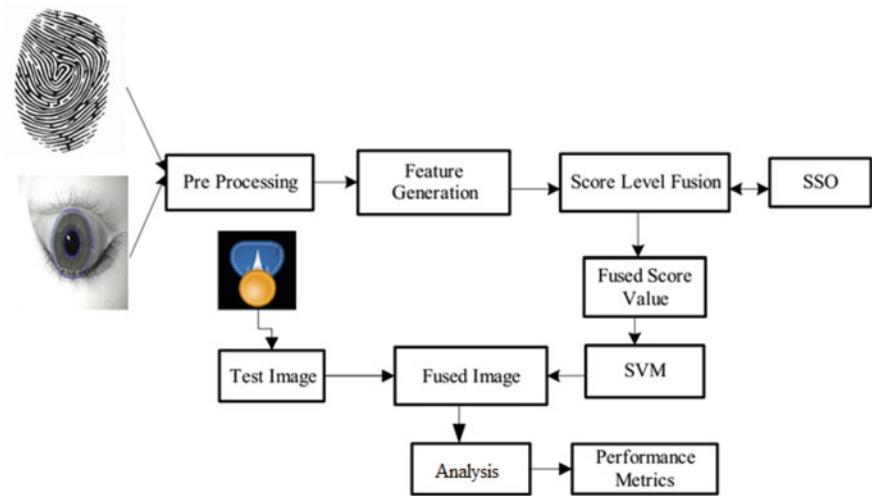


Fig. 1 Block diagram for proposed biometric method

For the purpose of the score level fusion, proficiently utilizes the correlation metrics to biometric recognition score level fusion are utilized. From the diverse classifiers, the optimization process integrates diverse classifiers as of scores achieved. For the purpose of identification, the trained system furnishes the test images in the testing phases. Figure 1 shows the comprehensive procedure of the novel technique. A minimization of the structural risk is SVM approach. The sum of the training error delimits a generalization error based on the feature set by fusion.

3.1 Basic Pre-Processing for Biometrics

For the succeeding two procedures, one of them effectively adapts the input image as ideal including recognition and feature extraction. The color image into gray image is adopted by necessary. According to the results of image analysis, the quality of feature extraction is affected by image impressive positive present in an image pre-processing. The mathematical normalization of a fingerprint is similar to biometric image pre-processing. The contrast adaption performs an effective method. The successive procedure furnishes processed image subsequent to the pre-processing function.

3.2 Image Feature Selection

Very leading part that plays a feature extraction is a task of the input image verification and identification. Certain properties or features evaluate the scale down original dataset to final motive of the feature extraction. The input pattern from the other is distinguished by an ability. Score level fusion process is the next step in which texture and wavelet features are extracted.

Score Level Fusion (SLF-SSO).

The input biometric data than the output decision of a matcher or matching score as of feature set contains more information. According to the fusion at the feature level, recognition results expect a fusion at the feature level. But in the proposed work, feature-based score level fusion is adopted, the strategies including minimum and maximum of a score, sum and simple average to fuse face and fingerprint at score level. The Slap Swarm Optimization (SSO) selects the optimal rules-based fuzzy system. The best numbers of fuzzy terms are determined using the SWO algorithm based on each data. Recalculate the fitness function with new fuzzy term number. A fitness function of SSO represents the classification accuracy of medical data. Next, the SSO process is continued at the end of termination condition met. The following section explains the SSO algorithm.

(i) *Representation of the solution:*

The medical data classification limits zero and one in each solution. For the slap swarm optimization algorithm, the feature classification work utilizes binary should be developed. The solution is defined via one-dimensional vector. The vector length is represented using the real dataset based on the number of features. All vector cells hold one and zero value. One is selected as the relevant medical data and zero for irrelevant medical data class. Equation (1) denotes the continuous value into binary value mapping.

$$X_{nm} = \begin{cases} 1 & \text{if } Y_{nm} > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Therefore, a discrete form X_{nm} is represented by the solution vector Y . Where Y_{nm} is the continuous position of the search agent n at dimension m . Figure 2 explains the sample features to the dataset of 7 attributes. The remaining one value is discarded and four chosen features (1, 4, 6 and 7) for classification task.

1	0	0	1	0	1	1
---	---	---	---	---	---	---

Fig. 2 Slap solution representation for classification

(ii) ***Fitness function Evaluation:***

The multi-objective optimization problem represents the medical data classification in this section. The two conflicting objectives including maximal amount of classification accuracy and minimal number of selected features accomplishes. The optimal solution is presented in small number of features. The fitness function improves the classification performance according to all the search agents. Chosen features are balanced between selected features in each solution and classification accuracy. The fitness function in SSA is assessed by fitness function.

$$F = \delta \text{Error}(d) + \varphi \frac{|f|}{|t|} \quad (2)$$

where $\text{Error}(d)$ is identified subset classifier error rate from the above equation. The parameters φ are and δ controls both feature reduction and the classification. The total numbers of features are $|t|$ and identified feature subset is $|f|$. In this study, value of $\delta = 0.9$ and if δ tends to 0 and 1 then $\varphi = (1 - \delta)$.

(iii) ***The output of classification results:***

Initialize Slap swarm and decision tree with maximal number of iterations parameters. The multi-objective function is efficiency and classification accuracy. Equation (4) updates the decision tree position. We obtain the optimal and most relevant disease class if stopping criterion met or repeat the process. Figure 3 delineates best medical data classification result using SSO.

3.3 Biometrics Recognition

Biometrics acknowledgment is one of the most established and most esteemed examination regions in the field of example acknowledgment, however a few scientists have confronted bunches of issues because of unmannered information assortment. The ideal objective of this methodology is to expand the edge between the classes and to decrease the distance between the hyperplane core interests. To play out the non-direct cycle, the piece capacities are begun in the SVM grouping. In this investigation utilizes Radial Bias Function (RBF) part capacities to arrangement measure [11].

RBF Kernel Function

The (Gaussian) radial basis function kernel is machine learning. The SVM classification utilizes RBF kernel work. The results of external summations ascertained disconnected may utilize a classification for a test vector.

$$\text{RBF}(a, b) = \exp(-\delta||a - b||^2), \quad \delta > 0 \quad (4)$$

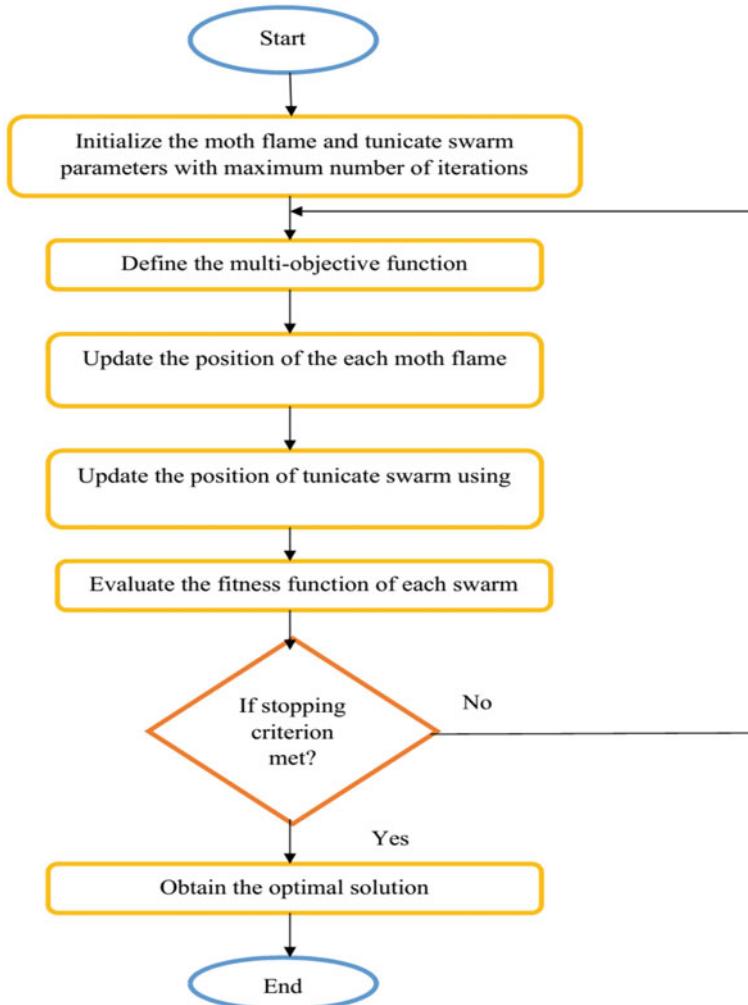


Fig. 3 Medical data classification using DT-SWO algorithm

Classifying the hyperplane is created and proficient the standard of the result by merging of two results. The data is classified via all the classifiers having training and testing levels [2].

Training phase: Based on above classification process, the input of the training procedure gives an output of attribute selection. The hyperplane knows every one of the process detachment of the position places. Improve the training specificity from the optimal features.

Testing Phase: The test the trained SVM utilizes a test data is a different dataset in which SVM has summed up the training dataset is decided. The thyroid database is

tested with training dataset precisely. The pattern net performs training of the system. A pattern net is used with the optimized feature set to train the proposed system.

4 Results Analysis

The high configuration system with MATLAB 2018a with i5 processor implements the proposed model. When compared to other security techniques, the False Rejection Rate, accuracy and False Acceptance Rate (FAR) are used to evaluate the iris images and fingerprint process.

The time-consuming tasks are a most critical task in data collection in any biometric recognition system. Return the closest match decision within the database that recorded as a template is compared to all records. The individual and authenticated deem the closest match within the allowed specific threshold. Figure 4 illustrates sample images. The analysis considers CASIA database.

As contrasted to the real technique, outperforms proposed technique that has been concluded. The SSO algorithm to authenticate the users improves the input image quality. The optimal feature set is delineated in Table 1. The output decision of a matcher or the matching score that the input biometric data by the feature set contains more information. According to pre-processing of input data, the better recognition results are provided with the help of feature level.

Where FRR is the false rejection rate and FAR is the False Acceptance Rate. For iris and fingerprint image, the FAR and FRR value are delineated in Table 2. When compared with the unimodal images, the fused image is minimum. Where simulation rounds are the number of simulations, simulation rounds are concerned by proposed model efficiency compared to the other optimization techniques. The accuracy plays a vital role for research work effectiveness validation. The reliability of the system modality is calculated using a criterion called accuracy. Due to its

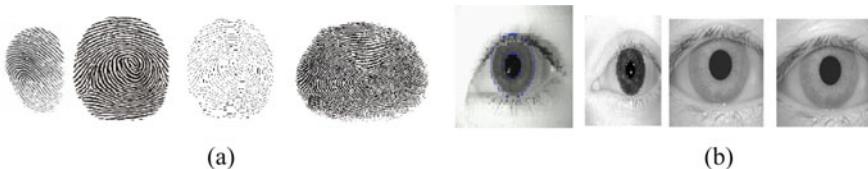


Fig. 4 Sample images **a** fingerprint **b** iris

Table 1 Optimal SLF-based feature

Image	Feature value		
Iris	350	356	189
Fingerprint	425	167	110

Table 2 Biometric security results (Optimal SLF with SVM)

Images	FRR	FAR	Accuracy
F1	0.67	0.88	0.97
F2	0.87	0.90	0.96
F3	0.76	0.78	0.88
F4	0.69	0.69	0.89
I1	0.92	0.89	0.75
I2	0.69	0.84	0.85
I3	0.73	0.65	0.69
I4	0.71	0.75	0.77

natural computation, the feature selection time minimized that is a merit of SSO with proposed work. Easily train the new environment.

Few other techniques like SLF with SLF, SVM and the above noted figures are the comparison chart of our proposed model. The FAR and FRR values are shown in Fig. 5 and 6. Over relative improvement, SVM fusion method is 93.4% by a classification absolute improvement. Significantly, the performances of a biometric system are improved as of different experts in whom effective combination of information is corroborated. Furthermore, the existing methods delineates difference is 6.7% and 8.9% but SVM produce 96.6%, with optimal SLF. When compared with others, 97.99% accuracy it's maximum. The smartphones accessed by image verification process become suitable. A greater accuracy, good performance with enabled high security (Fig. 7).

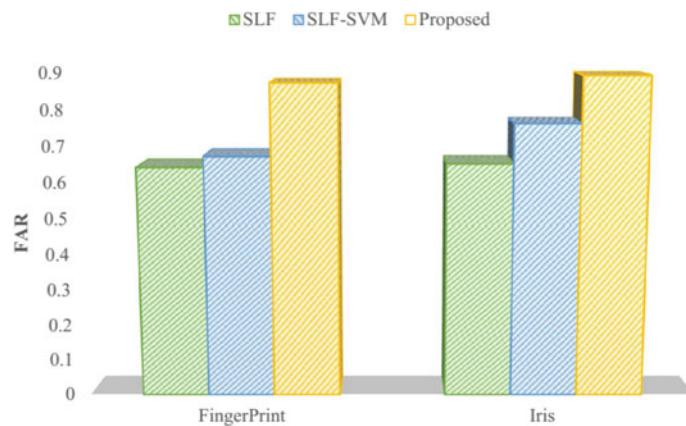
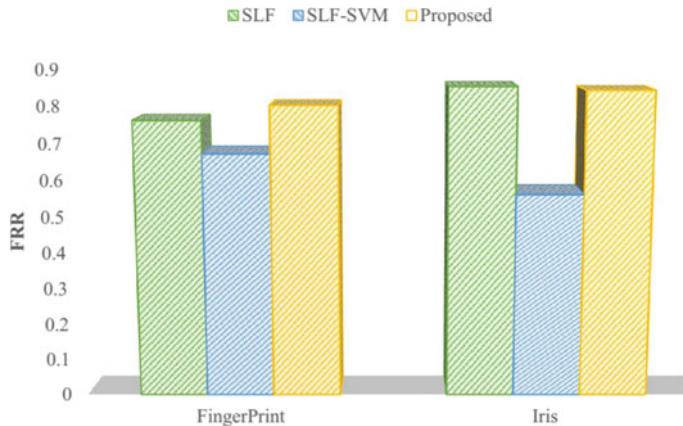
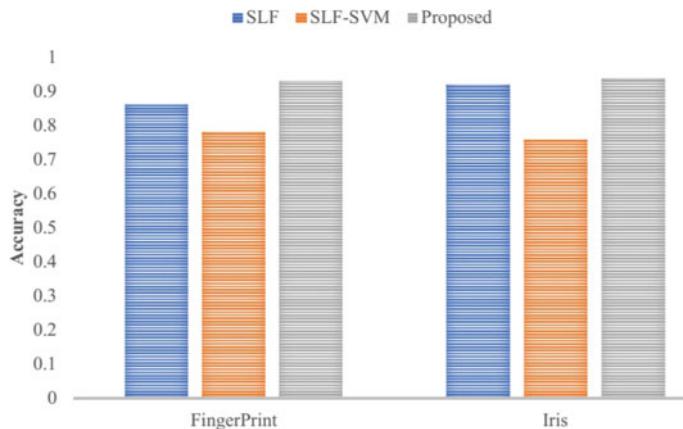


Fig. 5 FAR evaluation

**Fig. 6** FRR evaluation**Fig. 7** Accuracy evaluation

5 Conclusion

The iris and fingerprint recognition system is proposed in this chapter. In order to verify the authorized iris recognition technology, the grayscale eye images databases are tested. The effectiveness of the proposed system is evaluated by performing experimental study via biometric database. For both open and close-set condition, an excellent FAR value is produced using SVM classifier depending upon the obtained results. A good level of security is seemed via proposed system. The value of FRR is reduced to improve level of usability in further study. When contrasted to parallel modern methods, evaluate the new-fangled efficiency in performance. The proposed

procedure of par-excellence performance is ample credentials that appear in epoch-making technique yields charismatic outcomes. The recognition and identification results are excellent. The retinal identified and recognized easily using enhanced strategies.

Acknowledgements This research work has been supported by RUSA PHASE 2.0, Alagappa University, Karaikudi.

References

1. Ahamad D, Hameed SA, Akhtar M (2020) A multi-objective privacy preservation model for cloud security using hybrid Jaya-based shark smell optimization. *J King Saud Univ-Comput Inf Sci*
2. Rani BMS, Rani AJ (Mar 2017) A survey on classification techniques in biometric retinal system. In 2017 international conference on innovations in green energy and healthcare technologies (IGEHT), IEEE, pp 1–7
3. Deshpande PD, Mukherji P, Tavildar AS (2019) Accuracy enhancement of biometric recognition using iterative weights optimization algorithm. *EURASIP J Inf Secur* 2019(1):1–16
4. Jayaram MA, Fleyeh H (2013) Soft computing in biometrics: a pragmatic appraisal. *Am J Intell Syst* 3(3):105–112
5. Teodoro FGS, Peres SM, Lima CA (May 2017) Feature selection for biometric recognition based on electrocardiogram signals. In 2017 international joint conference on neural networks (IJCNN), IEEE, pp 2911–2920
6. Patro KK, Kumar PR (2017) Machine learning classification approaches for biometric recognition system using ECG signals. *J Eng Sci Technol Rev* 10(6)
7. Kumar T, Bhushan S, Jangra S (2019) An improved biometric fusion system based on fingerprint and face using optimized artificial neural network. *Int J Innovative Technol Explor Eng (IJITEE)* 8:1568–1575
8. Roy K, Bhattacharya P (Jan 2006) Iris recognition with support vector machines. In International conference on biometrics. Springer, Berlin, Heidelberg, pp 486–492
9. Barbosa M, Brouard T, Cauchie S, De Sousa SM (Jul 2008) Secure biometric authentication with improved accuracy. In Australasian conference on information security and privacy. Springer, Berlin, Heidelberg, pp 21–36
10. Mayron LM, Hausawi Y, Bahr GS (Jul 2013) Secure, usable biometric authentication systems. In International conference on universal access in human-computer interaction. Springer, Berlin, Heidelberg, pp 195–204
11. Yuan C, Sun X, Lv R (2016) Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Commun* 13(7):60–65
12. Mason J, Dave R, Chatterjee P, Graham-Allen I, Esterline A, Roy K (2020) An investigation of biometric authentication in the healthcare environment. *Array* 8:100042
13. Garg P, Jain AK (2020) “An invisible based watermarking technique for biometric image authentication.” *Materials Today: Proceedings*
14. Stylios I, Kokolakis S, Thanou O, Chatzis S (2021) Behavioral biometrics and continuous user authentication on mobile devices: a survey. *Information Fusion* 66:76–99
15. Padmanabhan S, Radhika KR (2021) Optimal feature selection-based biometric key management for identity management system: emotion oriented facial biometric system. *J Visual Commun Image Representation* 74:103002

16. Chhikara R, Kumari AC (2020) Feature selection optimization of HealthCare software product line using BBO. *Procedia Comput Sci* 167:1696–1704
17. Elavarasi G, Vanitha M (2021) Multiple secret share creation scheme with elephant herd optimization algorithm for biometric image security. *Adv Math: Sci J* 10(1):443–451

Hybrid Metaheuristic Algorithm-Based Clustering with Multi-Hop Routing Protocol for Wireless Sensor Networks



S. Jagadeesh and I. Muthulakshmi

Abstract Advancements in sensing and communication technologies resulted to the design of wireless sensor network (WSN) for low-cost distributed monitoring systems. Energy dissipation is a major problem in WSN. Clustering and routing are the familiar energy-efficient techniques offering several merits such as energy efficiency, network longevity, scalability, and less latency. Appropriate cluster heads (CHs) and optimal route selection processes can be assumed as the NP hard optimization problem and can be resolved by the use of metaheuristic algorithms. This paper presents a hybrid metaheuristic algorithm-based clustering with multi-hop routing (HMA-CMHR) protocol for WSN. The presented model incorporates different phases such as node initialization, clustering, routing, and data transmission. Firstly, the HMA-CMHR technique uses quantum harmony search algorithm (QHSA) based clustering process to elect an optimal subset of CHs. Secondly, the improved cuckoo search (ICS) algorithm based route technique is employed for an optimal selection of routes. The experimental results of the HMA-CMHR model are validated under different scenarios based on the number of nodes. An extensive experimental analysis is carried out to ensure the betterment of the HMA-CMHR method in terms of different measures. The experimental results showcased the superior performance of the HMA-CMHR technique over the compared methods in terms of distinct aspects.

Keywords Wireless sensor network · Metaheuristic algorithms · Harmony search · Cuckoo search · Quantum computing

1 Introduction

In general, wireless sensor network (WSN) is composed of sensor nodes subjected for sensing, computation, and wireless transmission between the nodes. The network is generally arranged for predicting event-based data and forward to base Station (BS) or

S. Jagadeesh (✉) · I. Muthulakshmi

Department of Computer Science and Engineering, V V College of Engineering, Tuticorin, Tamil Nadu, India

sink node for further examination [1]. WSN is a well-known and reputed system with massive benefits and applied in various domains like climatic forecasting, military service, healthcare, smart home, target observing, traffic observation, free management, farming verification, industrial damage prediction, and power management. Moreover, WSN is developed with maximum number of nodes in remote regions which could not be retrieved by human being. Hence, the deployment of independent and power effective network from sensor nodes is essential for extending the network lifespan and balanced power dispersion [2]. Also, energy efficiency is associated with effectual data routing where collective nodes are developed for reducing the power utilization and manage the overload, while interference among the sensor nodes is reduced [3].

Basically, most of the power consumption takes place in data transmission, processing, and so on. Hence, data transmission applies to maximum power when compared with all other processes [4]. Therefore, effective data communication and processing methods should be deployed for expanding the system duration. In order to implement typical process, nodes inside the network have to interact with one another [5]. The data aggregation approach is initialized by three major constituents of data collection that has to be assumed. Initially, aggregation is applied by the protocol. Finally, routing approach describes that the routing protocol applied for sending the gathered data to BS by network structure.

The main aim of this model is to reduce the transmission distance so that the power consumption is limited and eliminate the routing whole issues also named hot spot problems. According to the logical structure, routing protocols are classified into two classes. The initial class is flat routing, where the responsibility of a node in a system is similar and no special nodes are required. Hence, merits of these protocols are efficiency. Followed by second class is referred to be hierarchical based routing.

The traditional concepts of hierarchical based routing are called clustering. Also, it contributes to dividing the system into minimum set of nodes termed clusters. In a cluster, hierarchy is categorized as cluster head (CH) and cluster member (CM) [2]. Mostly, the CH collects the data from CM nodes. Afterward, data is collected and transmitted to upstream nodes. Low-energy adaptive clustering hierarchy (LEACH) and hybrid, energy-efficient, distributed (HEED) are two traditional clustering techniques. It is varied in CH election process. First, LEACH is developed on the concern of node power which is symmetric in CH selection, whereas HEED assumes the asymmetric node power for optimizing the network duration.

For establishing a stable election protocol (SEP) [6] presented the hierarchical routing in which two types of nodes have the selection probability. A clustering protocol is presented in [7] named energy effective heterogeneous clustered (EEHC) approach in the heterogeneous way. Also, the nodes in a network are classified into three classes namely, normal, advanced, and super nodes depend upon the basic energy. Eventually, normal nodes are filled with minimum energy, while advanced nodes have considerable energy when compared with normal nodes, and super nodes have supreme level of power. Followed by EEHC depends upon SEP, and three kinds of nodes available in EEHC has corresponding election probability to become a CH within a limited time period. A reactive clustering approach is presented in [8]

called distance-based residual energy-efficient SEP (DRESEP) for dissimilar WSN. DRESEP concerns residual energy (RE) of nodes and the distances from BS (DBS) which is referred as attributes for CH election. Additionally, a stable edition of DRESEP called stable EE clustering protocol (SECP) is presented in [9] where the CH selection is carried out in deterministic manner which depends upon the RE of nodes for distributing the load for all nodes. Developers have integrated the clustering approach with routing framework for expanding the network lifetime.

This paper introduces a novel hybridization of metaheuristic algorithm based clustering with multi-hop routing (HMA-CMHR) protocol for WSN. The HMA-CMHR technique performs clustering process to elect CHs using quantum harmony search algorithm (QHSA) by deriving a fitness function based on four parameters. In addition, the improved cuckoo search (ICS) algorithm is employed for routing process. The experimental outcome of the HMA-CMHR technique has been validated and the results are determined under diverse aspects.

2 Literature Survey

Evolutionary algorithm (EA) is subjected to manage the cluster-based issues for optimizing power consumption and expand the system period with heterogeneity. The energy-aware evolutionary routing protocol (EAERP), evolutionary-based clustered routing protocol (ERP), stable-aware ERP (SAERP) as well as stable threshold-sensitive EE routing protocols (STERP) by applying differential evolution (DE), HAS, and spider monkey optimization (SMO) are some of currently deployed EA based clustering approaches. EAERP has redeveloped the important features of EA, which ensures extended stable period and lifespan with significant power dispersion. ERP is applicable in resolving the limitation of hierarchical cluster-based routing (HCR) method [10] by integrating the factors like cohesion as well as separation error. SAERP integrated the merits of SEP and EAs for the purpose of enhancing stability. Moreover, energy-aware heuristics are used for CH election and enhance stability. Moreover, GA-based protocol [11] is applied to resolve the load distribution issues of CHs.

A DE relied on clustering technique is presented in [12] to expand the network duration. Shokouhifar and Jalali [13] projected application-based RP to maximize the duration of WSN based on the application applied. Rao et al. [14] implied novel chemical reaction optimization (nCRO) framework based EE clustering model for accomplishing optimal power efficiency. In order to eliminate the energy hole issues, Rao et al. [15] presented unequal clustering and routing protocol according to nCRO model. Moreover, Rao et al. [16] introduced an EE particle swarm optimization (PSO) based CH election framework. An optimal CH election is achieved by limiting the average intra-cluster and BS distance, and maximizes the energy of CH nodes. This study deals with clustering protocols which are employed for extending the system duration and reduce the power consumption of effective clustering models. But, in cluster that relied on WSN, CHs tolerate excess overhead subsidized by CM. Here,

multi-hop inter-cluster data transmission is applied for resolving the load distribution constraints of WSN, whereas the intra-cluster data transmission involves CM to apply threshold selection relied reactive procedure for data communication to CH. Finally, developing energy effective clustering approach is referred to as NP-hard issues.

3 The HMA-CMHR Protocol

The working principle involved in the presented HMA-CMHR technique is presented in Fig. 1. The figure states that the nodes are arbitrarily placed in the target region. Then, the information exchange among the nodes is carried out at the time of initialization, which finds useful during clustering and routing process. Next, the CHs are elected among the nodes by the QHSA algorithm and the rest of the nodes are declared as CMs. Followed by the optimal paths between CHs and BS are determined by the use of ICS protocol. At last, the communication between CHs and BS takes place via the chosen routes effectively.

3.1 QHSA Based Clustering Protocol

Generally, when musicians create harmony, numerous music pitches are originated with better significance. The harmony search (HS) is well-known and effective in identifying best solution for engineering crisis. Actually, HS is evolved from the performance of harmony maximization. In the fundamental HS model, where 4 principal steps are applied.

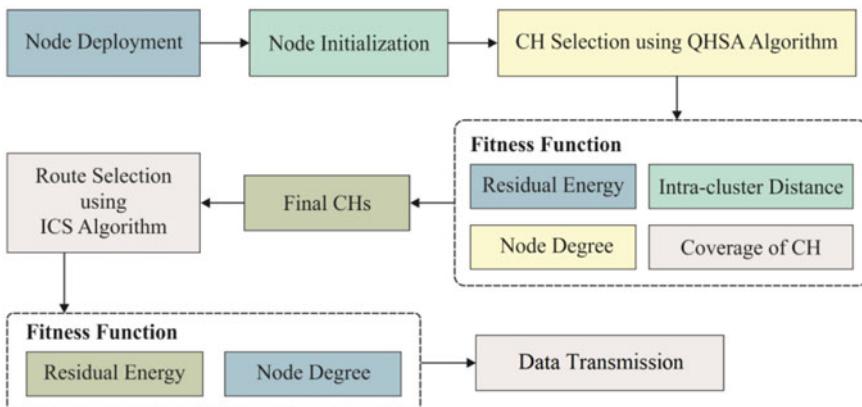


Fig. 1 Working process of HMA-CMHR model

- Step 1.** Initiate HS Memory (HM). The basic HM is composed of specific solutions for optimization issues which are emulated randomly. For n -dimension issues, HM with size of N is depicted as given below:

$$\text{HM} = \begin{bmatrix} x_1^1, x_2^1, \dots, x_n^1 \\ x_1^2, x_2^2, \dots, x_n^2 \\ \vdots \\ x_1^{\text{HMS}}, x_2^{\text{HMS}}, \dots, x_n^{\text{HMS}} \end{bmatrix}, \quad (1)$$

where $[x_1^i, x_2^i, \dots, x_n^i]$ ($i = 1, 2, \dots, \text{HMS}$) implies a solution candidate.

- Step 2.** Maximize new solution $[x'_1, x'_2, \dots, x'_n]$ from HM. A component of a solution, x'_j is accomplished by concerning HM considering rate (HMCR). It is described as possibility for selecting an element from HM candidates and 1-HMCR is probability of producing random solution [17]. When x'_j is emerged from HM, it is selected from j th dimension of arbitrary HM member and mutated on the basis of pitching adjust rate (PAR). However, the maximization of $[x'_1, x'_2, \dots, x'_n]$ is identical to generate offspring in genetic algorithms (GAs) in conjunction with mutation and crossover process.
- Step 3.** Upgrade HM. When optimal fitness value is attained which is inferior, then it is replaced by better HM. Else, it is removed.
- Step 4.** Follow Step2 to Step3 till reaching the termination condition (maximum count of iterations).

As same as GA and swarm intelligence (SI) methods, the HS model is referred to as a random search model. In this approach, no prior domain knowledge is required like gradient details of objective functions. Therefore, it is diverse from population-centric methods, it applies single HM for evolution. The performance of HAS can be improved by using quantum computing concept and QHSA model.

Quantum computing is an area of computer science, mainly based on quantum computers with the concept of quantum procedure like state superposition, entanglement, and quantum gate. Based on the Dirac definition, the Q-bit can be defined as an integration of the states of $|0\rangle$ and $|1\rangle$ as given below:

$$|Q\rangle = \alpha|0\rangle + \beta|1\rangle \text{ such that } |\alpha|^2 + |\beta|^2 = 1 \quad (2)$$

where α and β indicate complex values. $|\alpha|^2$ (resp. $|\beta|^2$) is the possibility of identifying the Q-bit in state 0 (resp. in state 1). A quantum register of size n is afterward established from a collection of n Q-bits. It defines a superposition of n Q-bits comprising up to 2^n probable values concurrently. A quantum register can be defined using Eq. (3):

$$\Psi = \sum_{x=0}^{2^n-1} C_x |X\rangle \quad (3)$$

The amplitude C_x satisfies the subsequent property:

$$\sum_{x=0}^{2^n-1} |C_x|^2 = 1 \quad (4)$$

The condition of Q-bit can be predicted by a quantum gate (Q-gate), and defined as unitary operator U substituting on the Q-bit basis states filling $U^+U = UU^+$, where U^+ is the Hermitian adjoint of U .

To enhance the network duration of cluster-based WSN, optimal set of best CH has to be elected. This is accomplished by multi-objective fitness function (FF) with 4 attributes like residual node energy, node degree (ND), intra-cluster distance as well as coverage ratio.

Node Energy (Node_{energy}): The newly developed clustering protocol applies high-energy node as optimal candidate for CH election process. As CH is responsible for cluster management and data collection when compared with CM , where reasonable energy has to be deployed and facilitate considerable power utilization. Hence, the residual energy (RE) of sensor node is evaluated by using,

$$\text{Minimize Node}_{\text{energy}} = \sum_{i=1}^m \frac{1}{E_{\text{CH}_i}} \quad (5)$$

In this approach E_{CH_i} indicates RE of i th CH and m implies the count of CHs.

Node Degree (Node_{degree}): It is the count of sensor nodes which can be reached from CH . It is employed for balancing the overhead at CH .

$$\text{Minimize Node}_{\text{degree}} = \sum_{i=1}^m |\text{CM}_i| \quad (6)$$

Here, $|\text{CM}_i|$ defines the count of CM of the i th CH .

Intra-cluster distance ($D_{\text{intra-cluster}}$): It is illustrated as average intra-cluster distance of CH from the CM [18].

$$\text{Minimize } D_{\text{intra-cluster}} = \sum_{j=1}^m \left[\frac{\sum_{i=1}^{|\text{CM}_j|} d(\text{CH}_j, \text{CM}_i)}{|\text{CM}_j|} \right] \quad (7)$$

Followed by $d(\text{CH}_j, \text{CM}_i)$ implies the Euclidean distance between j th CH and i th CM.

Coverage of CH (CCH): The actual responsibility of this attribute is to eliminate the un-clustered sensor nodes and make sure the contribution of residual sensor nodes

in clustering. Hence, it enhances the coverage of elected CHs where the parameter evaluation is computed by:

$$\text{Minimize } \text{CH}_{\text{coverage}} = \frac{(N - m) - \sum_{j=1}^m |\text{CM}_j|}{\sum_{j=1}^m |\text{CM}_j|} \quad (8)$$

where N defines overall count of sensor nodes, m refers the count of CHs and $|\text{CM}_j|$ signifies number of CM in j th cluster. Finally, multi-objective FF (F) is evaluated as weighted sum of 4 parameters as depicted in the following:

$$F = w_1 \times \text{Node}_{\text{energy}} + w_2 \times \text{Node}_{\text{degree}} + w_3 \times D_{\text{intra-cluster}} + w_4 \times \text{CH}_{\text{coverage}} \quad (9)$$

Linear programming development for best position CH election problem is express as follows:

$$\begin{aligned} \text{Minimize } F &= w_1 \times \text{Node}_{\text{energy}} + w_2 \times \text{Node}_{\text{degree}} + w_3 \times D_{\text{intra-cluster}} \\ &\quad + w_4 \times \text{CH}_{\text{coverage}} \end{aligned}$$

subject to

$$\text{Node}_{\text{energy}} > E_{\text{th}} \quad (10)$$

$$\text{Node}_{\text{degree}} \leq ND_{\text{th}} \quad (11)$$

$$D_{\text{intra-cluster}} < T_{\text{max}} \quad (12)$$

$$w_1 + w_2 + w_3 + w_4 = 1, w_1, w_2, w_3 \text{ and } w_4 \in (0, 1) \quad (13)$$

Here, E_{th} defines threshold node energy, ND_{th} stands for threshold value of ND where the value is regarded as N/m . T_{max} depicts the high-communication radius of sensor node.

In order to select an optimal nest, the cost function is defined below:

$$x_1 = \max_{k=1,2,3,\dots,K} \left\{ \sum d(n_i, CH_{e,k}) / |C_{e,k}| \right\} \quad (14)$$

$$x_2 = \sum_{i=1}^N E(n_i) / \sum_{k=1}^K E(CH_{e,k}) \quad (15)$$

$$\text{cost} = \beta * x_1 + (1 + \beta) * x_2 \quad (16)$$

In this expression, x_1 implies high-average Euclidean distance among nodes and related $CH.C_{e,k}$ refers the count of nodes which are in transmission radius of cluster C_k of egg e. Function x_2 refers the ratio of energy in all nodes. The value of β is 0.5. Low value of function x_1 and x_2 guides in reducing the intra-cluster distance and for selecting optimal CH that limits power application.

3.2 ICS Based Routing Protocol

Once the clustering process is completed, a CH gathers the received sensor information from CM and forwards the collected information to BS through multi-hop transmission. In order to enhance the system duration, problem of selecting energy effective routing path from CH and BS is considered as NP-hard problem. In this work, ICS algorithm is applied for resolving the problem. The fundamental CS method depends upon the brood parasitism of cuckoo species by laying eggs in the nests of alternate host birds. Here, 3 ideal rules have been applied: (1) A cuckoo lays single egg at a time, and hides in a randomly selected set; (2) optimal nests with better qualified eggs would be carried for the future generations; (3) number of accessible host nests are allocated, and egg laid by a cuckoo is identified by host bird with a probability $p_a \in [0, 1]$. Moreover, the technique applied a balanced unification of a local random walk as well as global explorative random walk, managed by using a switching attribute p_a . Hence, local random walk is expressed as,

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t), \quad (17)$$

where x_j^t and x_k^t imply diverse solutions decided in random fashion, H refers a heaviside performance, ε implies a random value retrieved from uniform distribution, and s denotes the step size. Besides, global random walk is processed with the help of Lévy flights:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(s, \lambda). \quad (18)$$

In this approach, $\alpha > 0$ depicts the step size scaling factor [19]; Lévy (s, λ) signifies the step-lengths distributed on the basis of probability distribution depicted in (6) with infinite variance and infinite mean:

$$\text{Levy}(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}}. \quad (19)$$

For enhancing the searching capability of the model, the orthogonal as well as simulated annealing (SA) methods are combined with CS approach. The fundamental procedure of orthogonal design is to apply the features of fractional process and compute optimal level of integration. Here, orthogonal array of K factors and Q

levels with M combinations is implied as $L_M(Q^K)$, in which Q refers the prime value, $M = Q^J$, and J indicates a positive integer by $K = (Q^J - 1)/(Q - 1)$. The newly projected routing algorithm depends upon an improved CS optimization method. Here, a node estimates the hop count from BS. Next, a node i estimates the probability $P(i, j)$ of node selection j as next-hop node in routing from node i to BS by given expression:

$$P(i, j) = \begin{cases} \frac{h_j}{\sum_{k \in N_i} h_k} \times \frac{E_i}{\sum_{k \in N_i} E_k} & \text{if } j \in N_i \\ 0 & \text{else} \end{cases} \quad (20)$$

where N_i defines the count of neighboring nodes of node i . h_i and h_k depict the score of hop count from node i and j correspondingly. E_i Indicates RE of node i . Figure 2 demonstrates the flowchart of CS model.

4 Performance Validation

In HMA-CMHR method is estimated utilized simulated execution with different situations based on the position of BS. The group of 3 cases is demonstrated as S1, S2, and S3, respectively, and showcased in Figs. 3 and 4.

- S1-BS is located in the middle of the target region.
- S2-BS is positioned in the corner of the target region.
- S3-BS takes positioned distant from the target place.

The network with a collection of 300 nodes utilizing random deployment occurs in the target region of $200*200m^2$. The parameters retrieved to validate model are exhibited in Tables 1 and 2 and Fig. 4.

5 Conclusion

This paper has developed a new HMA-CMHR protocol for WSN. The presented model incorporates different phases such as node initialization, clustering, routing and data transmission. Primarily, the information exchange process among the nodes is carried out at the time of initialization, which finds useful during clustering and routing process. Next, the CHs are elected among the nodes by the QHSA algorithm by deriving a fitness function based on residual energy, node degree, intra-cluster distance, and coverage ratio; the rest of the nodes is declared as CMs. Followed by the optimal routes from the CHs to BS are determined by the use of ICS protocol. At last, the data transmission from the CHs to BS takes place via the chosen routes effectively. An extensive experimental analysis is carried out to ensure the betterment of the HMA-CMHR method in terms of energy consumption, network lifetime, and

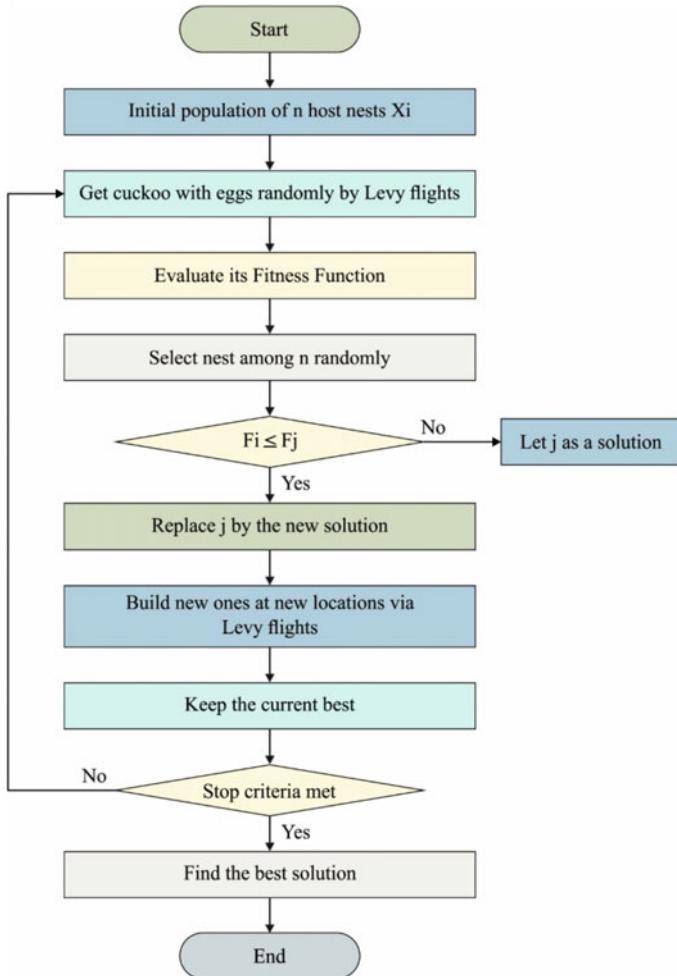


Fig. 2 Flowchart of CS algorithm

number of packets to the base station. The experimental outcome of the HMA-CMHR technique has been validated, and the results are determined under diverse aspects.

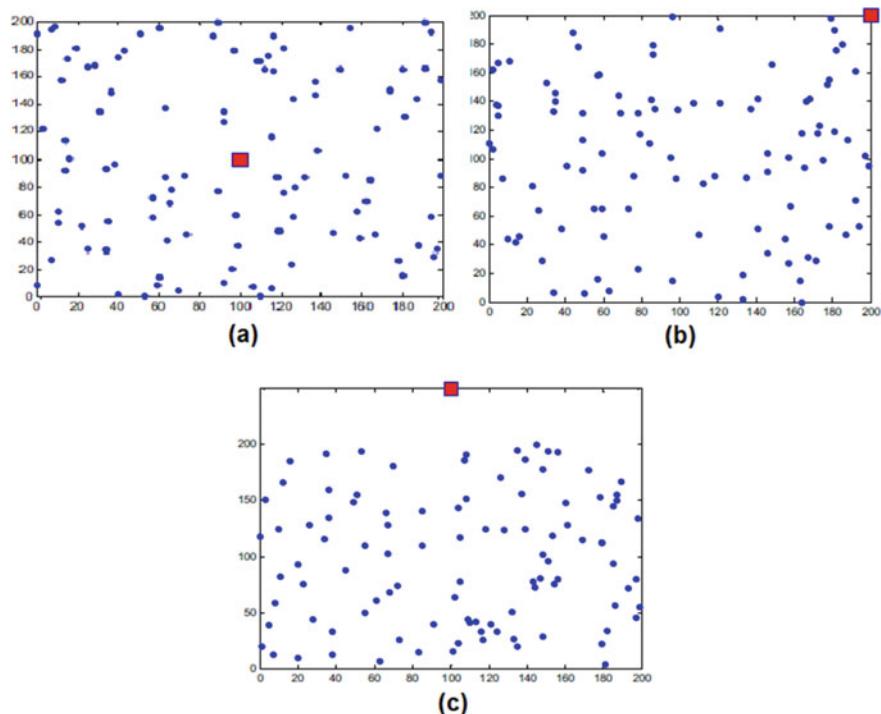


Fig. 3 **a** S1: BS at middle of the target area **b** S2: BS at the corner of the target area **c** S3: BS distant from the target

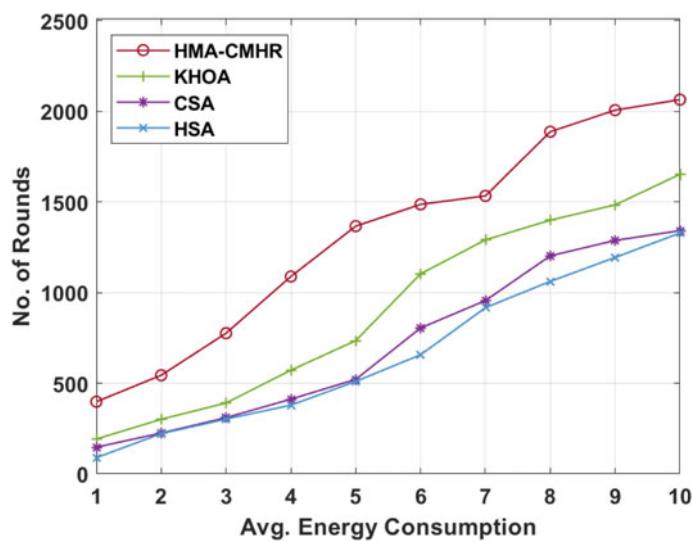


Fig. 4 Average energy consumption analysis of HMA-CMHR model on S1

Table 1 Simulation parameters

Parameters	Value
Area	200×200
E_0	0.5 J
E_{elec}	50nJ bit^{-1}
ε_{fs}	100pJbitm^{-2}
ε_{fs}	100pJbit m^{-2}
Packet size	4000 bits

Table 2 Network lifetime analysis of existing with HMA-CMHR method

Algorithms	Scenario 1			Scenario 2			Scenario 3		
	FND	HND	LND	FND	HND	LND	FND	HND	LND
HAS	881	1157	1329	630	954	1058	550	736	918
CSA	1041	1121	1342	841	935	1168	658	856	1008
KHOA	1190	1404	1653	1231	1396	1458	1071	1197	1213
HMA-CMHR	1995	2041	2065	1771	1983	2204	1559	1697	1799

References

1. Afsar MM, Tayarani-N M (2014) Clustering in sensor networks: a literature survey. *J Netw Comput Appl* 46:198–226
2. Pantazis NA, Nikolidakis SA, Vergados DD (2013) Energy-efficient routing protocols in wireless sensor networks: a survey. *IEEE Commun Surveys Tutorials* 15(2):551–591
3. Halawani S, Khan AW (2010) Sensors lifetime enhancement techniques in wireless sensor networks—a survey. *J Comput* 2(5):34–47
4. Idris MYI, Znaid AMA, Wahab AWA, Qabajeh LK, Mahdi OA (2017) Low communication cost (LCC) scheme for localizing mobile wireless sensor networks. *Wireless Netw* 23(3):737–747
5. Heinzelman WB, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd annual Hawaii international conference on system sciences (HICSS-33). IEEE
6. Smaragdakis G, Matta I, Bestavros A (2004) SEP: a stable election protocol for clustered heterogeneous wireless sensor networks. In: Proceedings of the international workshop on SANPA
7. Kumar D, Aseri TC, Patel RB (2009) EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks. *Comput Commun* 32:662–667
8. Mittal N, Singh U (2015) Distance-based residual energy-efficient stable election protocol for WSNs. *Arabian J Sci Eng* 40(6):1637–1646
9. Mittal N, Singh U, Sohi BS (2016) A stable energy efficient clustering protocol for wireless sensor networks. *Wireless Networks*
10. Hussain S, Matin AW (2006) Hierarchical cluster-based routing in wireless sensor networks. In: IEEE/ACM international conference on information processing in sensor networks, IPSN
11. Kuila P, Gupta SK, Jana PK (2013) A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. *Swarm Evol Comput* 12:48–56
12. Kuila P, Jana PK (2014) A novel differential evolution based clustering algorithm for wireless sensor networks. *Appl Soft Comput* 25:414–425

13. Shokouhifar M, Jalali A (2015) A new evolutionary based application specific routing protocol for clustered wireless sensor networks. *Int J Electron Commun* 69:432–441
14. Rao PC, Banka H (2015) Energy efficient clustering algorithms for wireless sensor networks: novel chemical reaction optimization approach. *Wireless Networks*
15. Rao PC, Banka H (2016) Novel chemical reaction optimization based unequal clustering and routing algorithms for wireless sensor networks. *Wireless Networks*
16. Rao PC, Jana PK, Banka H (2017) A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks. *Wireless Netw* 23(7):2005–2020
17. Gao XZ, Govindasamy V, Xu H, Wang X, Zenger K (2015) Harmony search method: theory and applications. *Comput Intell Neurosci* 2015
18. Gupta GP, Jha S (2018) Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques. *Eng Appl Artif Intell* 68:101–109
19. Wang J, Zhou B, Zhou S (2016) An improved cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation. *Comput Intell Neurosci* 2016
20. Gao D, Zhang S, Zhang F, Fan X, Zhang J (2019) Maximum data generation rate routing protocol based on data flow controlling technology for rechargeable wireless sensor networks. *CMC-Comput Mater Contin* 59:649–667
21. Vijayalakshmi K, Anandan P (2020) Global levy flight of cuckoo search with particle swarm optimization for effective cluster head selection in wireless sensor network. *Intell Autom Soft Comput* 26(2):303–311

Author Index

A

- Agarwal, Amit, 209
Agarwal, Drishti, 21, 615
Aherwadi, Nagnath, 481
Ahlawat, Anil, 713
Ahuja, Neeru, 347
Alam, Mansaf, 125
Aneja, Rattan Deep, 95
Ansari, Manzoor, 125
Arora, Sangeeta, 713
Arora, Shaveta, 505
Arun, P., 431
Arya, Vivek, 327

B

- Beniwal, Rohit, 49
Bhaskar, Shabina, 1
Bhatghare, Ashish, 209
Bhatia, Pradeep Kumar, 347
Bhatt, Devershi Pallavi, 113, 217
Bhatu, Vrinda, 267
Bindal, Amit Kumar, 95, 367
Bukhari, Syed Nisar Hussain, 275

C

- Careena, P., 431
Chandra Priya, J., 687
Chandrasekaran, K., 561
Chandrashekhar, Ankush, 561
Chaudhary, Vikas, 815
Chawla, Paras, 491
Chhabra, Charu, 305
Choudhry, Mahipal Singh, 241

Chugh, Medha, 605

D

- Devarakonda, Nagaraju, 721
Dhankhar, Amita, 637
Dharmale, Shivani G., 441
Dogra, Manmohan, 197, 199
Domala, Jayashree, 197, 199
Dsouza, Kevin, 197, 199

F

- Falor, Ayush, 293
Fernandes, Dwayne, 197, 199

G

- Garg, Anchal, 665
Gautam, Ritu, 357, 831
G.Elavarasi, 831
Ghai, Deepika, 673
Ghose, Shreyasi, 665
Goel, Anmol, 605
Goel, Kunal, 367
Gomase, Snehal A., 441
Goyal, Lalit, 153
Goyal, Vishal, 153
Gunwant, Harsh, 815
Gupta, Bhawna, 615
Gupta, Deepak, 73
Gupta, Paras, 615
Gupta, Priyanshi, 769
Gupta, Rahul, 797
Gupta, Shelley, 409

Gupta, Vishal, 739

H

Hannan, Ummae Hamida, 9
 Haq, Ehtishamul, 275
 Harikrishnan, V., 625
 Hegde, Aniket S., 37
 Hirani, Manav, 293
 Honnavalli, Prasad B., 37, 625
 Humayun Kabir, Md., 9
 Hussain, Imran, 537

J

Jagadeesh, S., 843
 Jain, Amit, 275
 Jain, Rachna, 21
 Jangra, Manisha, 61
 Jha, Kaustubh, 37
 Johari, Rahul, 605
 Joshi, Abhisht, 815

K

Kalyal, Dikshit, 491
 Kamath, Aayush, 267
 Kamparia, Aditya, 73
 Kapoor, Preeti, 505
 Karale, Nikhil E., 471
 Karthika, R. N., 687
 Kataria, Aman, 327
 Kaur, Narinder, 739
 Kaur, Prableen, 85, 357
 Kaur, Rajwinder, 85
 Kaur, Supreet, 789
 Khalique, Aqeel, 537
 Khamparia, Aditya, 471, 481, 547
 Khandelwal, Parth, 605
 Kohar, Rachna, 141
 Krishnan, Deepa, 229, 293
 Kumar, Akshay, 21
 Kumar, K. Suresh, 647
 Kumar, P. Y. V. N. Dileep, 547
 Kumar, Pranav Vigneshwar, 561
 Kumar, Shakti, 95
 Kumar, Vivek, 419

L

Lande, Milind V., 165
 Lemuel, K. B. J., 181
 Likhithkar, Praveen P., 459

M

Madan, Sajal, 21
 Majithia, Arjun, 209
 Malavath, Pallavi, 721
 Malhotra, Jigyasu, 209
 Mary Synthuja Jain Preetha, M., 431
 Maurya, Saumya, 241
 Mehta, Aina, 85
 Mehta, Priyank, 293
 Mishra, Sambit Kumar, 573, 585
 Mishra, Vinay K., 419
 Mittal, Shubham, 61
 Modani, U. S., 595
 Muthulakshmi, I., 843

N

Nagrath, Preeti, 21, 615
 Naim, Forhad An, 9
 Nanda, Sunpreet Kaur, 673

P

Pande, Sagar, 73, 441, 459, 471, 481, 547, 673
 Pant, Nilay, 797
 Paranjape, Tejas, 267
 Parouha, Raghav Prasad, 397
 Pimple, Kanchan M., 459
 Polkowski, Zdzislaw, 573, 585, 721
 Pundir, Meena, 85

R

Raman, Shatakshi, 769
 Rani, Annu, 153
 rani, Challapalli Jhansi, 721
 Rani, Poonam, 209
 Rani, Sita, 327
 Ranjan, Jayanthi, 409
 Rathee, Devanshu, 797
 Ridhorkar, Sonali, 165

S

Sahazeer, K. S., 625
 Sandhu, Jasminder Kaur, 85
 Sanket, H. S., 625
 Sanyal, Ipsita, 253
 Sapra, Luxmi, 85
 Saquib, Mohd, 537
 Sashanka, K., 21
 Sawant, Rupali, 267
 Sharma, Anshu, 337
 Sharma, Dilip K., 419

Sharma, Dilip Kumar, 701
Sharma, Manik, 305, 357
Sharma, Meghna, 305
Sharma, Moolchand, 815
Sharma, Pankaj Kumar, 595
Sharma, Preeti, 113
Shekhawat, Kirty, 217
Shokeen, Jyoti, 209
Singal, Anuj, 61
Singh, Archana, 409
Singh, Ashish, 49
Singh, Kuldeep, 61
Singh, Laiphakpam Dolendro, 387
Singh, Nishi, 665
Singh, Purnima, 481, 547
Singh, Rajpreet, 491
Singh, Rohit Pratap, 387
Solanki, Kamna, 637
Sonavane, Apurva, 141
Srinivasaraghavan, Anuradha, 197, 199
Subramanyaswamy, V., 181
Sudhamani, M. J., 253
Suganthi, N., 647

Suganthi, S., 37
Suresh Kumar, K., 687

T

Taneja, Harsh, 789
Taneja, Soham, 615
Tanwar, Poonam, 337
Tawalare, Swati C., 471
Thasleema, T. M., 1
Tilekar, Pruthvi, 481

V

Valarmathie, P., 687
Vanitha, M., 831
Varshney, Ravi Prakash, 701
Vatsya, Ritambhra, 665
Vedant, Henil, 293
Venkatesha, M. K., 253
Vinay, Siddarth, 625
Vividha, 615