

Vikrant Bhateja
Sheng-Lung Peng
Suresh Chandra Satapathy
Yu-Dong Zhang *Editors*

Evolution in Computational Intelligence

Frontiers in Intelligent Computing:
Theory and Applications (FICTA 2020),
Volume 1

Advances in Intelligent Systems and Computing

Volume 1176

Series Editor

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

Advisory Editors

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing,
Universidad Central de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagras, School of Computer Science and Electronic Engineering,
University of Essex, Colchester, UK

László T. Kóczy, Department of Automation, Széchenyi István University,
Gyor, Hungary

Vladik Kreinovich, Department of Computer Science, University of Texas
at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao
Tung University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information Technology,
University of Technology Sydney, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute
of Technology, Tijuana, Mexico

Nadia Nedjah, Department of Electronics Engineering, University of Rio de Janeiro,
Rio de Janeiro, Brazil

Ngoc Thanh Nguyen[✉], Faculty of Computer Science and Management,
Wrocław University of Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Shatin, Hong Kong

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at <http://www.springer.com/series/11156>

Vikrant Bhateja · Sheng-Lung Peng ·
Suresh Chandra Satapathy · Yu-Dong Zhang
Editors

Evolution in Computational Intelligence

Frontiers in Intelligent Computing: Theory
and Applications (FICTA 2020), Volume 1

Editors

Vikrant Bhateja
Department of Electronics
and Communication Engineering
Shri Ramswaroop Memorial Group
of Professional Colleges (SRMGPC)
Lucknow, Uttar Pradesh, India

Dr. A.P.J. Abdul Kalam Technical
University
Lucknow, Uttar Pradesh, India

Suresh Chandra Satapathy
School of Computer Engineering
Kalinga Institute of Industrial
Technology (KIIT)
Bhubaneswar, Odisha, India

Sheng-Lung Peng
Department of Computer Science
and Information Engineering
National Dong Hwa University
Hualien, Taiwan

Yu-Dong Zhang
Department of Informatics
University of Leicester
Leicester, UK

ISSN 2194-5357

ISSN 2194-5365 (electronic)

Advances in Intelligent Systems and Computing

ISBN 978-981-15-5787-3

ISBN 978-981-15-5788-0 (eBook)

<https://doi.org/10.1007/978-981-15-5788-0>

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

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

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

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

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

Organization

Chief Patrons

Prof. K. Balaveera Reddy, Chairman, BOG, NITK Surathkal
Prof. Karanam Umamaheshwar Rao, Director, NITK Surathkal

Patrons

Prof. Ananthanarayana V. S., Deputy Director, NITK Surathkal
Prof. Aloysius H. Sequeira, Dean Faculty Welfare, NITK Surathkal
Prof. G. Ram Mohana Reddy, Professor-HAG, IT Department, NITK Surathkal
Dr. S. Pavan Kumar, Head, School of Management, NITK Surathkal

Organizing Chairs

Dr. Ritanjali Majhi, Associate Professor, School of Management, NITK Surathkal
Dr. Sowmya Kamath S., Assistant Professor, Department of Information Technology, NITK Surathkal
Dr. Suprabha K. R., Assistant Professor, School of Management, NITK Surathkal
Dr. Geetha V., Assistant Professor, Department of Information Technology, NITK Surathkal
Dr. Rashmi Uchil, Assistant Professor, School of Management, NITK Surathkal
Dr. Biju R. Mohan, Assistant Professor and Head, Department of Information Technology, NITK Surathkal
Dr. Pradyot Ranjan Jena, Assistant Professor, School of Management, NITK Surathkal

Dr. Nagamma Patil, Assistant Professor, Department of Information Technology,
NITK Surathkal

Publicity Chairs

Dr. Suprabha K. R., Assistant Professor, SOM, NITK Surathkal, India
Dr. Geetha V., Assistant Professor, Department of IT, NITK Surathkal, India
Dr. Rashmi Uchil, Assistant Professor, SOM, NITK Surathkal, India
Dr. Biju R. Mohan, Assistant Professor and Head, Department of IT, NITK
Surathkal, India

Advisory Committee

Prof. Abrar A. Qureshi, Professor, University of Virginia's College at Wise, USA
Dr. Alastair W. Watson, Program Director, Faculty of Business, University of
Wollongong, Dubai
Prof. Anjan K. Swain, IIM, Kozhikode, India
Prof. Anurag Mittal, Department of Computer Science and Engineering, IIT
Madras, India
Dr. Armin Haller, Australian National University, Canberra
Prof. Arnab K. Laha, IIM Ahmedabad, India
Prof. Ashok K. Pradhan, IIT Kharagpur, India
Prof. Athanasios V. Vasilakos, Professor, University of Western Macedonia,
Greece/Athens
Prof. Atreyi Kankanhalli, NUS School of Computing, Singapore
Prof. A. H. Sequeira, Dean Faculty Welfare, NITK Surathkal, India
Prof. Carlos A. Coello Coello, Centro de Investigación y de Estudios Avanzados
del Instituto
Prof. Charles Vincent, Director of Research, Buckingham University, UK
Prof. Chilukuri K. Mohan, Professor, Syracuse University, Syracuse, NY, USA
Prof. Dipankar Dasgupta, Professor, The University of Memphis, TN
Prof. Durga Toshniwal, IIT Roorkee, India
Dr. Elena Cabrio, University of Nice Sophia Antipolis, Inria, CNRS, I3S, France
Prof. Ganapati Panda, Ex. Deputy Director, IIT Bhubaneswar
Prof. Gerardo Beni, Professor, University of California, CA, USA
Dr. Giancarlo Giudici, Politecnico di Milano, DIG School of Management, Milano,
Italy
Prof. G. K. Venayagamoorthy, Professor, Clemson University, Clemson, SC, USA
Mr. Harish Kamath, Master Technologist, HP Enterprise, Bengaluru
Prof. Heitor Silvério Lopes, Professor, Federal University of Technology Paraná,
Brazil

Prof. Hoang Pham, Distinguished Professor, Rutgers University, Piscataway, NJ, USA

Prof. Jeng-Shyang Pan, Shandong University of Science and Technology, Qingdao, China

Prof. Juan Luis Fernández Martínez, Professor, University of Oviedo, Spain

Prof. Kailash C. Patidar, Senior Professor, University of the Western Cape, South Africa

Prof. Kerry Taylor, Australian National University, Canberra

Prof. Kumkum Garg, Ex. Prof. IIT Roorkee, Pro-Vice Chancellor Manipal University, Jaipur

Prof. K. Parsopoulos, Associate Professor, University of Ioannina, Greece

Prof. Leandro Dos Santos Coelho, Associate Professor, Federal University of Parana, Brazil

Prof. Lexing Xie, Professor of Computer Science, Australian National University, Canberra

Prof. Lingfeng Wang, University of Wisconsin-Milwaukee Milwaukee, WI, USA

Mr. Mahesha Nanjundaiah, Director of Engineering, HP Enterprise, Bengaluru

Prof. Maurice Clerc, Independent Consultant, France Télécom, Annecy, France

Prof. M. A. Abido, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

Prof. Naeem Hanoon, Multimedia University, Cyberjaya, Malaysia

Prof. Narasimha Murthy, Department of Computer Science and Automation, IISc, Bangalore

Prof. Oscar Castillo, Professor, Tijuana Institute of Technology, Mexico

Prof. Pei-Chann Chang, Professor, Yuan Ze University, Taoyuan, Taiwan

Prof. Peng Shi, Professor, University of Adelaide, Adelaide, SA, Australia

Dr. Prakash Raghavendra, Principal Member of Technical Staff, AMD, India

Prof. Rafael Stubs Parpinelli, Professor, State University of Santa Catarina, Brazil

Prof. Raj Acharya, Dean and Rudy Professor of Engineering, Computer Science and Informatics, Indiana University, USA

Prof. Raghav Gowda, Professor, University of Dayton, Ohio, USA

Prof. Roderich Gross, Senior Lecturer, University of Sheffield, UK

Mr. Rudramuni, Vice President, Dell EMC, Bengaluru

Prof. Saman Halgamuge, Professor, University of Melbourne, Australia

Prof. Subhadip Basu, Professor, Jadavpur University, India

Prof. Sumanth Yenduri, Professor, Kennesaw State University, USA

Prof. Sumit Kumar Jha, Department of Computer Science, University of Central Florida, USA

Prof. S. G. Ponnambalam, Professor, Subang Jaya, Malaysia

Dr. Suyash P. Awate, Department of Computer Science and Engineering, IIT Bombay

Dr. Valerio Basile, Research Fellow, University of Turin, Italy

Dr. Vineeth Balasubramanian, Department of Computer Science and Engineering, IIT Hyderabad

Dr. Vikash Ramiah, Associate Professor, Applied Finance, University of South Australia

Prof. X. Z. Gao, Docent, Aalto University School of Electrical Engineering, Finland

Prof. Ying Tan, Associate Professor, The University of Melbourne, Australia

Prof. Zong Woo Geem, Gachon University in South Korea

Technical Program Committee

Dr. Anand Kumar M., Assistant Professor, Department of IT, NITK Surathkal

Dr. Babita Majhi, Assistant Professor, Department of IT, G. G. University, Bilaspur

Dr. Bhawana Rudra, Assistant Professor, Department of IT, NITK Surathkal

Dr. Bibhu Prasad Nayak, Associate Professor, Department of HSS, TISS Hyderabad

Dr. Bijuna C. Mohan, Assistant Professor, School of Management, NITK Surathkal

Dr. Dhishna P., Assistant Professor, School of Management, NITK Surathkal

Mr. Dinesh Naik, Assistant Professor, Department of Information Technology, NITK Surathkal

Prof. Geetha Maiya, Department of Computer Science and Engineering, MIT Manipal

Dr. Gopalakrishna B. V., Assistant Professor, School of Management, NITK Surathkal

Dr. Keshavamurthy B. N., Associate Professor, Department of CSE, NIT, Goa

Dr. Kiran M., Assistant Professor, Department of Information Technology, NITK Surathkal

Prof. K. B. Kiran, Professor, School of Management, NITK Surathkal

Dr. Madhu Kumari, Assistant Professor, Department of CSE, NIT Hamirpur

Dr. Mussarrat Shaheen, Assistant Professor, IBS, Hyderabad

Dr. Pilli Shubhakar, Associate Professor, Department of CSE, MNIT Jaipur

Dr. P. R. K. Gupta, Institute of Finance and International Management, Bengaluru

Dr. Rajesh Acharya H., Assistant Professor, School of Management, NITK Surathkal

Dr. Ranjay Hazra, Assistant Professor, Department of EIE, NIT, Silchar

Dr. Ravikumar Jatot, Associate Professor, Department of ECE, NIT, Warangal

Dr. Rohit Budhiraja, Assistant Professor, Department of EE, IIT Kanpur

Dr. Sandeep Kumar, Associate Professor, Department of CSE, IIT Roorkee

Dr. Savita Bhat, Assistant Professor, School of Management, NITK Surathkal

Dr. Shashikantha Koudur, Associate Professor, School of Management, NITK Surathkal

Dr. Shridhar Domanal, IBM India, Bengaluru

Dr. Sheena, Associate Professor, School of Management, NITK Surathkal

Dr. Sreejith A., Assistant Professor, School of Management, NITK Surathkal

Dr. Sudhindra Bhat, Professor and Deputy Vice Chancellor, Isbat University, Uganda

Dr. Surekha Nayak, Assistant Professor, Christ University, Bangalore

Dr. Suresh S., Associate Professor, Department of IT, SRM University, Chennai

Dr. Tejavathu Ramesh, Assistant Professor, Department of EE, NIT, Andhra Pradesh

Dr. Yogita, Assistant Professor, Department of CSE, NIT, Meghalaya

Preface

This book is a collection of high-quality peer-reviewed research papers presented at the 8th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA-2020) held at the National Institute of Technology, Karnataka, Surathkal, India, from January 4–5, 2020.

The idea of this conference series was conceived by few eminent professors and researchers from premier institutions of India. The first three editions of this conference: FICTA-2012, 2013, and 2014 were organized by Bhubaneswar Engineering College (BEC), Bhubaneswar, Odisha, India. The fourth edition FICTA-2015 was held at NIT, Durgapur, West Bengal, India. The fifth and sixth editions FICTA-2016 and FICTA-2017 were consecutively organized by KIIT University, Bhubaneswar, Odisha, India. FICTA-2018 was hosted by Duy Tan University, Da Nang City, Viet Nam. All past seven editions of the FICTA conference proceedings are published in Springer AISC Series. Presently, FICTA-2020 is the eighth edition of this conference series which aims to bring together researchers, scientists, engineers, and practitioners to exchange and share their theories, methodologies, new ideas, experiences, applications in all areas of intelligent computing theories, and applications in various engineering disciplines like computer science, electronics, electrical, mechanical, bio-medical engineering, etc.

FICTA-2020 had received a good number of submissions from the different areas relating to computational intelligence, intelligent data engineering, data analytics, decision sciences, and associated applications in the arena of intelligent computing. These papers have undergone a rigorous peer-review process with the help of our technical program committee members (from the country as well as abroad). The review process has been very crucial with minimum 02 reviews each and in many cases 3–5 reviews along with due checks on similarity and content overlap as well. This conference witnessed more than 300 papers including the main track as well as special sessions. The conference featured five special sessions in various cutting-edge technologies of specialized focus which were organized and chaired by eminent professors. The total toll of papers included submissions received cross-country along with 06 overseas countries. Out of this pool, only 147 papers were given acceptance and segregated as two different volumes for

publication under the proceedings. This volume consists of 74 papers from diverse areas of evolution in computational intelligence.

The conference featured many distinguished keynote addresses in different spheres of intelligent computing by eminent speakers like Dr. Venkat N. Gudivada, (Professor and Chair, Department of Computer Science, East Carolina University, Greenville, USA); Prof. Ganapati Panda (Professor and Former Deputy Director, Indian Institute of Technology, Bhubaneswar, Orissa, India); Dr. Lipo Wang (School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore). Last but not the least, the invited talk on “Importance of Ethics in Research Publishing” delivered by Mr. Aninda Bose (Senior Editor—Interdisciplinary Applied Sciences, Publishing Department, Springer Nature) received ample applause from the vast audience of delegates, budding researchers, faculty, and students.

We thank the advisory chairs and steering committees for rendering mentor support to the conference. An extreme note of gratitude to Prof. Suresh Chandra Satapathy (KIIT University, Bhubaneshwar, Odisha, India) for providing valuable guidelines and being an inspiration in the entire process of organizing this conference. We would also like to thank the School of Management and the Department of Information Technology, NIT Karnataka, Surathkal, who jointly came forward and provided their support to organize the eighth edition of this conference series.

We take this opportunity to thank authors of all submitted papers for their hard work, adherence to the deadlines, and patience with the review process. The quality of a refereed volume depends mainly on the expertise and dedication of the reviewers. We are indebted to the technical program committee members who not only produced excellent reviews but also did these in short time frames. We would also like to thank the participants of this conference, who have participated in the conference above all hardships.

Lucknow, India
Hualien, Taiwan
Bhubaneswar, India
Leicester, UK

Volume Editors
Dr. Vikrant Bhateja
Dr. Sheng-Lung Peng
Dr. Suresh Chandra Satapathy
Dr. Yu-Dong Zhang

Contents

Faster Result Retrieval from Health Care Product Sales Data Warehouse Using Materialized Queries	1
Sonali Chakraborty and Jyotika Doshi	
Sense Scheduling for Robotics Cognitive Intelligence	11
Mahendra Bhatu Gawali and Swapnali Sunil Gawali	
Feature Selection Using Ant Colony Optimization and Weighted Visibility Graph	17
Leena C. Sekhar and R. Vijayakumar	
From Generic to Custom: A Survey on Role of Machine Learning in Pharmacogenomics, Its Applications and Challenges	33
Sana Aimani and Kiran Kumari Patil	
Personalised Structure Balance Theory-Based Movie Recommendation System	43
Aishwarya Sivakumar, Nidheesa Amedapu, Vasudha Avuthu, and M. Brindha	
Machine Learning Techniques for the Investigation of Phishing Websites	55
Ajaykumar K. B. and Bhawana Rudra	
Feature Extraction and Classification of Gestures from Myo-Electric Data Using a Neural Network Classifier	65
Praahas Amin, Airani Mohammad Khan, Akshay Ram Bhat, and Gautham Rao	
Text-Convolutional Neural Networks for Fake News Detection in Tweets	81
Harsh Sinha, Sakshi, and Yashvardhan Sharma	

Effect of Soil and Climatic Attribute on Greenhouse Gas Emission from Agriculture Sector	91
Pranali K. Kosamkar and Vrushali Y. Kulkarni	
Optimal Image Feature Ranking and Fusion for Visual Question Answering	103
Sruthy Manmadhan and Binsu C. Kovoor	
An Investigation on Indoor Navigation Systems	115
J. Akilandeswari, A. Naveenkumar, R. S. Sabeenian, P. Iyyanar, M. E. Paramasivam, and G. Jothi	
Conceptualization and Design of Remotely-Accessible Hardware Interface (RAHI) Laboratory	125
Shivam Mahesh Potdar, Vanshika Gupta, Pruthviraj Umesh, and K. V. Gangadharan	
A Non-invasive approach for Driver Drowsiness Detection using Convolutional Neural Networks	135
Sreelakshmi K. K. and J. Jennifer Ranjani	
Disaster Severity Analysis from Micro-Blog Texts Using Deep-NN	145
Ramesh Wadawadagi and Veerappa Pagi	
WEKA Result Reader—A Smart Tool for Reading and Summarizing WEKA Simulator Files	159
Ranjit Panigrahi, Samarjeet Borah, and Udit Kumar Chakraborty	
Predicting Reliability of Web Services Using Hidden Markov Model	169
Shridhar Allagi and Pradeep Surasura	
Optimal Contrast and Size-Invariant Recursive VCS Using Perfect Reconstruction of White Pixels	181
T. E. Jisha and Thomas Monoth	
Performance Analysis of Brain Imaging Using Enriched CGLS and MRNSD in Microwave Tomography	191
N. Nithya, R. Sivani Priya, and M. S. K. Manikandan	
Analysis and Identification of EEG Features for Mental Stress	201
Mitul Kumar Ahirwal	
Blockchain-Based Grievance Management System	211
Rakshitha Shettigar, Nishant Dalvi, Ketan Ingale, Farhan Ansari, and Ramkrushna C. Maheshwar	
Utility of Neural Embeddings in Semantic Similarity of Text Data	223
Manik Hendre, Prasenjit Mukherjee, and Manish Godse	

Big Data and Machine Learning Analytics to Detect Epileptic Seizures with Minimum Delay Using Random Window Optimization	233
S. Sanila and S. Sathyalakshmi	
An Efficient Evaluation of Spatial Search on Road Networks Using G-Tree	243
C. P. Shahina	
Investigation into the Efficacy of Various Machine Learning Techniques for Mitigation in Credit Card Fraud Detection	255
S. R. Lenka, M. Pant, R. K. Barik, S. S. Patra, and H. Dubey	
Temporal Modeling of On-Street Parking Data for Detection of Parking Violation in Smart Cities	265
Shiv Kumar Sahoo, Niranjan Panigrahi, Debasis Mohapatra, Asutosh Panda, and Arvind Sinha	
Performance Optimization of Big Data Applications Using Parameter Tuning of Data Platform Features Through Feature Selection Techniques	273
Tanuja Pattanshetti and Vahida Attar	
Development of Emotional Decision-Making Model Using EEG Signals	281
Mitul Kumar Ahirwal and Mangesh Ramaji Kose	
HMM Classifier Object Recognizing System in Brain–Computer Interface	287
H. S. Anupama, Raj V. Jain, Revannur Venkatesh, N. K. Cauvery, and G. M. Lingaraju	
Deep Learning for Stock Index Tracking: Bank Sector Case	295
R. Arjun, K. R. Suprabha, and Ritanjali Majhi	
Rank Consensus Between Importance Measures in Hypergraph Model of Social Network	305
Debasis Mohapatra and Manas Ranjan Patra	
Amplifying the Polarity Categorization on Twitter Data Using Tweet Polarizer Algorithm and Emoticons Score	315
D. N. V. S. L. S. Indira and J. N. V. R. Swarup Kumar	
Classification of Fashion Images Using Transfer Learning	325
Raji S. Pillai and K. Sreekumar	
A Novel Adaptive Out of Step Protection in Synchronous Generators Using Support Vector Machine Algorithm	333
R. Hemavathi, I. Limsha Deborah, and M. Geethanjali	

Sentiment Analysis of Movie Reviews Using Support Vector Machine Classifier with Linear Kernel Function	345
A. Sheik Abdullah, K. Akash, J. ShaminThres, and S. Selvakumar	
Blockchain-Based Sybil-Secure Data Transmission (SSDT) IoT Framework for Smart City Applications	355
Sonal Kumar, Ayan Kumar Das, and Ditipriya Sinha	
An Empirical Study of Neural Network Hyperparameters.	371
Aditya Makwe and Abhishek Singh Rathore	
Waste Management System: Approach with IoT, Prediction, and Dashboard.	385
Viswanadhapalli Bhanuja, Ramai Varangaonkar, Yashveer Girdhar, and Kumar Kannan	
Brain Tumour Detection in MRI Using Deep Learning	395
S. Shanmuga Priya, S. Saran Raj, B. Surendiran, and N. Arulmurugaselvi	
Parallel Implementation of Luhn's Algorithm for Credit Card Validation Using MPI and CUDA	405
P. Karthik G. Kudva, M. L. Shreyas, B. Ashwath Rao, Shwetha Rai, and N. Gopalakrishna Kini	
Malayalam POS Tagger—A Comparison Using SVM and HMM	413
K. Usha and S. Lakshmana Pandian	
Comparison of CutShort: A Hybrid Sorting Technique Using MPI and CUDA	421
Harshit Yadav, Shraddha Naik, B. Ashwath Rao, Shwetha Rai, and Gopalakrishna Kini	
Self-Learning and Self-Organizing Log Files by Generating Recursive Associations	429
K. Indra Gandhi and A. Balaji	
Analysis and Prediction of Fantasy Cricket Contest Winners Using Machine Learning Techniques	443
K. Karthik, Gokul S. Krishnan, Shashank Shetty, Sanjay S. Bankapur, Ranjit P. Kolkar, T. S. Ashwin, and Manjunath K. Vanahalli	
IoT Stream Data Compression Using LDPC Coding	455
Rajni Jindal, Neetesh Kumar, and Sanjay Patidar	
Improved PSO for Task Scheduling in Cloud Computing	467
Richa and Bettahally N. Keshavamurthy	
Parallel Implementation of kNN Algorithm for Breast Cancer Detection	475
Suhas Athani, Shreesha Joshi, B. Ashwath Rao, Shwetha Rai, and N. Gopalakrishna Kini	

Correlated High Average-Utility Itemset Mining	485
Krishan Kumar Sethi and Dharavath Ramesh	
Interactive Labelled Object Treemap: Visualization Tool for Multiple Hierarchies	499
Mahipal Jadeja, Hitarth Kanakia, and Rahul Muthu	
Medical Transcriptions and UMLS-Based Disease Inference and Risk Assessment Using Machine Learning	509
Thamizharuvi Arikrishnan and S. Swamynathan	
Parallel Message Encryption Through Playfair Cipher Using CUDA	519
Saloni Goyal, Balie Shalomi Pacholi, B. Ashwath Rao, Shwetha Rai, and N. Gopalakrishna Kini	
Text Classification Using Multilingual Sentence Embeddings	527
Anant Saraswat, Kumar Abhishek, and Sheshank Kumar	
Real-Time Yawn Extraction for Driver's Drowsiness Detection	537
Sumeet Saurav, Mehul Kasliwal, Raghav Agrawal, Sanjay Singh, and Ravi Saini	
FIoT: A QoS-Aware Fog-IoT Framework to Minimize Latency in IoT Applications via Fog Offloading	551
K. S. Arikumar and V. Natarajan	
A Study on Implementation of Text Analytics over Legal Domain	561
Dipanjan Saha, Riya Sil, and Abhishek Roy	
Advanced Key Management System (AKMS) for Security in Public Clouds	573
J. L. Amarnath, Pritam G. Shah, and H. Chandramouli	
Machine Learning and Feature Selection Based Ransomware Detection Using Hexacodes	583
Bheemidi Vikram Reddy, Gutha Jaya Krishna, Vadlamani Ravi, and Dipankar Dasgupta	
Hypertension Risk Prediction Using Deep Neural Network	599
M. J. Sivambigai and E. Murugavalli	
Machine Learning Approach for Student Academic Performance Prediction	611
Sachin Rai, K. Aditya Shastry, Surendra Pratap, Saurav Kishore, Priyanka Mishra, and H. A. Sanjay	
Detection of A Stable Age in Children for Face Recognition Application	619
R. Sumithra, N. Vinay Kumar, and D. S. Guru	

An Automatic Predictive Model for Sorting of Artificially and Naturally Ripened Mangoes	633
Anitha Raghavendra, D. S. Guru, and Mahesh K. Rao	
A Comparative Analysis of Different Classifiers to Propose a Genetically Optimized Neural Network	647
Ankita Tiwari and Bhawana Sahu	
Vehicle Direction-Based B-MFR Routing Protocol for VANET	659
Parimala Gar nepudi and D. Venkatesulu	
Glacier Surface Flow Velocity of Hunza Basin, Karakoram Using Satellite Optical Data	669
S. Sivarajanji, M. Geetha Priya, and D. Krishnaveni	
Diabetic Retinopathy Detection Using Transfer Learning and Deep Learning	679
Akhilesh Kumar Gangwar and Vadlamani Ravi	
Prediction of Admissions and Jobs in Technical Courses with Respect to Demographic Location Using Multi-linear Regression Model	691
Anjali Mishra, Aishwary Kumar, Shakti Mishra, and H. A. Sanjay	
Content-Based Image Retrieval Using Statistical Color Occurrence Feature on Multiresolution Dataset	701
Debanjan Pathak, U. S. N. Raju, Sukhdev Singh, G. Naveen, and K. Anil	
EEDCHS-PSO: Energy-Efficient Dynamic Cluster Head Selection with Differential Evolution and Particle Swarm Optimization for Wireless Sensor Networks (WSNS)	715
T. Guhan, N. Revathy, K. Anuradha, and B. Sathyabama	
An Unsupervised Searching Scheme over Encrypted Cloud Database	727
T. Janani and M. Brindha	
Impact of Dimension Reduced Spectral Features on Open Set Domain Adaptation for Hyperspectral Image Classification	737
Krishnendu C. S., V. Sowmya, and K. P. Soman	
Fog-Based Video Surveillance System for Smart City Applications	747
B. V. Natesha and Ram Mohana Reddy Gudde ti	
Performance Improvement of Deep Residual Skip Convolution Neural Network for Atrial Fibrillation Classification	755
Sanjana K., V. Sowmya, E. A. Gopalakrishnan, and K. P. Soman	
Detection and Classification of Faults in Photovoltaic System Using Random Forest Algorithm	765
C. Sowthily, S. Senthil Kumar, and M. Brindha	

Clustering Enhanced Encoder–Decoder Approach to Dimensionality Reduction and Encryption	775
B. R. Mukesh, Nara Madhumitha, N. Pai Aditya, Srinivas Vivek, and Anand Kumar M.	
Cryptographic Algorithm Identification Using Deep Learning Techniques	785
Sandeep Pamidiparthi and Sirisha Velampalli	
Author Index	795

About the Editors

Vikrant Bhateja is Associate Professor, Department of ECE in SRMGP, Lucknow. His areas of research include digital image and video processing, computer vision, medical imaging, machine learning, pattern analysis and recognition. He has around 150 quality publications in various international journals and conference proceedings. He has edited more than 25 volumes of conference proceedings with Springer Nature (AISC, SIST, LNEE, LNNS Series). He is associate editor of IJSE and IJRSDA. And presently EiC of IJNCR journal under IGI Global.

Sheng-Lung Peng is a Professor at the Department of Computer Science and Information Engineering at National Dong Hwa University, Hualien, Taiwan. He is also a supervisor at the Chinese Information Literacy Association, and the Association of Algorithms and Computation Theory. His research interests include designing and analyzing algorithms for bioinformatics, combinatorics, data mining and networks. He has edited several special issues in respected journals and published more than 100 international conference and journal papers.

Suresh Chandra Satapathy is a Professor at the School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, India. His research interests include machine learning, data mining, and swarm intelligence studies and their applications to engineering. He has published over 140 publications in various respected international journals and conference proceedings. He has edited several volumes of AISC, LNEE and SIST, and is a senior member of IEEE and life member of the Computer Society of India.

Yu-Dong Zhang received his Ph.D. degree from Southeast University, China, in 2010. From 2010 to 2013, he worked as a post-doc and then as a research scientist at Columbia University, USA. He served as a Professor at Nanjing Normal University from 2013 to 2017, and is currently a Full Professor at the University of Leicester, UK. His research interests include deep learning in communication and signal processing, and medical image processing.

Faster Result Retrieval from Health Care Product Sales Data Warehouse Using Materialized Queries



Sonali Chakraborty  and Jyotika Doshi

Abstract Existing approaches for result retrieval from a Data Warehouse, i.e., Data Cubes and Materialized Views, incur more processing, maintenance and storage cost. For faster retrieval of query results from Data Warehouse, authors suggest storing executed OLAP queries and their results along with metadata in a relational database referred here as *Materialized Query Database (MQDB)*. For stored queries, processing incremental results using Data Marts is faster as compared to using Data Warehouse. Therefore, a significant reduction in query processing time is achieved using *MQDB*. Authors depict the working of proposed *MQDB* approach on the sales data of a health care product manufacturing organization by placing Data Warehouse on Centralized and on Cloud Server.

Keywords Data Warehouse · Data Marts · OLAP · Materialized queries · Faster query result retrieval

1 Introduction

Results of *OLAP (Online Analytical Processing)* queries are generated by traversing through warehouse data. For frequent queries, Data Warehouse is invoked repeatedly for generating same results. This is quite time consuming. Major approaches for result retrieval from Data Warehouse are Multidimensional Data Cubes [1–6] and Materialized Views [7–12]. Data Cubes and Materialized Views incur more storage, processing and maintenance cost. There exists a trade-off between materializing of the cube and the cost to materialize them [1]. Various techniques are proposed for reducing the materialization cost of Data Cubes [2–6]. Materialized Views are described as derived relations with respect to the base relations that are materialized by storing in database [7]. Authors [8–12] discuss about the view maintenance overhead issues and discuss various techniques to overcome them.

S. Chakraborty (✉) · J. Doshi
GLS University, Ahmedabad, Gujarat, India
e-mail: chakrabartysonali@gmail.com

Authors suggest storing executed *OLAP* queries with their results and metadata information in a relational database referred here as *Materialized Query Database (MQDB)*. Metadata includes *timestamp*, *frequency*, *threshold*, *number of records in output*, *path of result table* and *path of Data Mart (for processing incremental data)*. When an *OLAP* query is executed, it is determined if its *synonymous* query exists in *MQDB*. Two queries generating same results are referred as *synonymous* queries [13]. Thereafter, it is determined if the query requires an incremental update. If no incremental updates are required, then existing results are fetched from *MQDB* [13]. For queries requiring incremental updates, incremental results are generated using Data Marts as they are faster compared to using Data Warehouse [14]. Final results are derived by combining stored results with incremental results [15].

Here, authors illustrate the working of proposed *MQDB* approach for *Biotechnika Healthcare Pvt. Ltd.* a health care product manufacturing organization.

In further part of the literature, the following abbreviations are referred: Data Warehouse—DW; Data Mart—DM; Data Cubes—DC, Materialized Views—MV; Materialized Query Database—*MQDB*; Centralized Server—CenS; Cloud Server—CldS

2 Implementation of *MQDB* for *Biotechnika Healthcare Pvt. Ltd*

The organization manufactures blood collection tubes under the brand name *iCollekt*. There are two categories of blood collection tubes; *Vacuum* and *Non-Vacuum* with many sub-variants. They are packed in trays of 100 tubes and are sold through distributors all over India divided into four zones (north, south, east and west). Implementation of *MQDB* includes the steps depicted in Algorithm 1.

Algorithm 1: Processing steps of MQDB approach

- I. Initialization: Identifier Generation
- II. For the input query, generate its *query identifier element*
 - a. If *query identifier element* followed by query criteria matches with a stored query in *MQDB*; (i.e., *synonymous* query exists)
 - i. If *synonymous* query results does not require incremental update
 - Fetch results from *MQDB* and update query *timestamp* and *frequency*
 - ii. Else, (*synonymous* query result requires an incremental update)
 - Process incremental data using DM
 - Compile stored results and incremental results to generate final results
 - Update query result table with query *timestamp* and *frequency*
 - b. Else, (for the input query, no *synonymous* query exists in *MQDB*)
 - i. Generate results of input query and store in *MQDB* as a new query

a. **Initialization [14, 16]**

During this phase tables, fields, functions, criteria clause and relational operators probable to be used in the queries are assigned identifiers [16].

- Here, **table identifier** for one de-normalized DW ‘dw_bt_sales’ is ‘01’.
- **Field identifiers** for table ‘dw_bt_sales’ are: (rec_id,01), (year, 02), (month, 03), (tube_name, 04), (var_name, 05), (production_box, 06), (dis_name, 07), (zone_name, 08), (state_name, 09), (trays_sold, 10), (entry_date,11)
- **Function identifiers** are: (no function, 00), (group by, 10), (limit, 20), (order by asc, 30), (order by desc, 40), (sum, 01), (average, 02), (min, 03), (max, 04), (count, 05), (std dev, 06), (variance, 07)
- **Criteria clause identifiers** are: (No Criteria, 00), (WHERE, 01), (HAVING, 02)
- Since the numeric criterion of the query is converted to a range with a minimum and maximum value; three **relational operators identifiers** are: (no operator, 00), (=, 01), (!=, 02), (BETWEEN, 03) **For example:** If query criteria is ‘salary < 10,000’; then the range after converting to ‘BETWEEN’ operator is (0, 9999).

b. **Storing Queries in MQDB [16].**

In *MQDB* approach, queries are stored in ‘*Stored_query*’ table while metadata information is stored in ‘*Materialized_query*’ table. The following instances of *OLAP* queries considering sales data of *Biotechnika Healthcare Pvt. Ltd* are used for testing.

B1: Find the average sales for Non-Vacuum Gel BCT in north zone for the year 2018.

```
SELECT tube_name, var_name, avg(trays_sold) FROM dw_bt_sales WHERE year = 2018 AND tube_name = 'Non-vacuum' AND var_name = 'Gel' AND zone_name = 'North'
```

B2: Find the total sales of each distributor for Vacuum Clot Act type of BCT for each year.

```
SELECT tube_name, var_name, dis_name, zone_name, sum (trays_sold), year
FROM dw_bt_sales WHERE tube_name = 'Vacuum' AND var_name = 'Clot Act'
GROUP BY dis_name, year
```

B3: Find the variance in sales of Vacuum and Non-vacuum BCT for each zone in each year.

```
SELECT tube_name, var (trays_sold), zone_name, year FROM dw_bt_sales
GROUP BY tube_name, zone_name, year
```

B4: Count the number of distributors of K2 EDTA Vacuum BCT in north zone in each year.

Table 1 ‘Stored_query’ table in MQDB

sq_id	qry_id	table_id	fld_id	func_id	criteria_id	op_id	List	min_range	max_range
sq01	q1	01	04	00	01	01	Non-Vacuum	NULL	NULL
sq02	q1	01	05	00	01	01	Gel	NULL	NULL
sq03	q1	01	10	02	00	00	NULL	NULL	NULL
sq04	q1	01	02	00	01	01	2018	NULL	NULL
sq05	q1	01	08	00	01	01	North	NULL	NULL

Table 2 ‘Materialized_query’ table in MQDB

qry_id	Timestamp	Frequency	Threshold	num_records	result_table_path	Data_Mart_path
q1	2019-04-23	5	10	1	q1_result	dm_bt_sales

SELECT tube_name, var_name, zone_name, year, count(dis_name)
FROM dw_bt_sales WHERE tube_name = ‘Vacuum’ AND var_name = ‘K2EDTA’
AND zone_name = ‘North’ GROUP BY year

B5: Find the number of trays sold by distributor ‘Medico Enterprise’ for Vacuum Sodium BCT in Gujarat state April, 2018.

SELECT tube_name, category_name, trays_sold FROM dw_bt_sales WHERE year = ‘2018’ AND month = ‘April’ AND tube_name = ‘Vacuum’ AND var_name = ‘Sodium’ AND dis_name = ‘Medico Enterprise’ AND state_name = ‘Gujarat’

Table 1 depicts storing of query B1 in ‘Stored_query’ table while Table 2 depicts storing of metadata information in ‘Materialized_query’ table.

c. Checking for existence of synonymous query in MQDB [14]

Consider **B_query1: Find the average sales for Non-Vacuum Gel BCT in the year 2018 for north zone.**

SELECT avg(trays_sold), tube_name, var_name, zone_name FROM dw_bt_sales
WHERE tube_name = ‘Non-vacuum’ AND var_name = ‘Gel’ AND zone_name = ‘North’ AND year = 2018

Identifier codes generated by combining each table-field -function for B_query1 are: (011002), (010400), (010500), (010800), (010200). Putting all identifier codes in one set forms the **query identifier element**. Thereafter, every member of *query identifier element* is matched with those of each stored query in MQDB irrespective of the sequence. Here, the *query identifier element* of B_query1 matches with that of B1 stored in MQDB. Criteria of B_query1 and B1 are compared and it is found that they are *synonymous* to each other. Average search time for the existence of a *synonymous* query is computed by taking five runs of program and populating MQDB with 50, 100 up to 400 queries. The results are tabulated in Table 3.

It is observed that with increase in number of queries stored in MQDB, search time increases. With less than 200 queries, rate of increase in search time is very less.

Table 3 Average time for searching for a *synonymous* query from *MQDB*

Number of queries stored in <i>MQDB</i>	Time(seconds)	Standard deviation	Standard error of mean
50	0.0237	0.0050	0.0022
100	0.0247	0.0022	0.0010
200	0.0258	0.0059	0.0026
300	0.0318	0.0040	0.0018
400	0.0440	0.0066	0.0030

d. **Checking for incremental updates [14]**

For incremental updates, *timestamp* is compared with last DW refresh date. Here, *timestamp* of *B1* ('2019-04-23') (yyyy-mm-dd) is less than last DW refresh ('2019-05-01'). Therefore, results of query *B1* require processing of incremental data.

e. **Synonymous query processing with incremental data from DM and compiling final results [14, 15]**

Since DM contains fewer incremental records, significant amount of query processing time is reduced when incremental data is considered from DM as compared to using DW [14]. Final results are derived using stored results from *MQDB* and incremental results using the methods discussed in [15]. Updated results for query *B1* are computed using the formula to compute combined average [15].

3 Experimental Results

DW '*dw_bt_sales*' is created having 100,000 sales records while a DM '*dm_bt_sales*' is populated with 10,000 incremental records. *MQDB* is populated with 200 queries. Experimental results are recorded by placing *MQDB* and query result tables on CenS while DW, DM and DC are placed on CenS and on CldS. Storage and maintenance cost of CenS can be reduced by placing DW on CldS offered by third party services. DC containing 70,000 aggregates is created for testing. Five MV are created for each query and are placed on CenS.

Application programs are created using *Python* programming language and *MySQL 5.7.19* is used as DBMS. System configuration for DW and DM on CenS are Intel(R) Core (TM) 2 Duo CPU E8400 @ 3 GHz, 3000 MHz processor with 2 GB RAM and MS Windows 7 Professional as operating system. For DW on CldS, experimental results are recorded using *Amazon RDS service for MySQL* with *MySQL Workbench 8.0 CE* as client application. The instance is placed in us-east-1d (N Virginia). Connection time to a cloud instance depends on cloud service provider, location of instance on CldS, connection bandwidth and latency time. Therefore, an additional time of approx. 3.6272 s is recorded for connecting to cloud instance irrespective of query complexity.

Using *MQDB*, total query processing time includes the following processing timings:

- i. Searching for existence of *synonymous* query in *MQDB*
- ii. Determining the need of incremental update for the stored query results
- iii. Fetching existing stored results from *MQDB* and updating metadata information
- iv. Generating incremental results from DM
- v. Compiling existing and incremental results to generate final results.

a. *Synonymous query processing without incremental data*

Experimental results are recorded for processing *synonymous* queries without incremental data using the following methods:

Method DW: Total query processing time is the time required to traverse all records of DW and generate results.

Method DC: Since DC stores results of aggregate queries only, query processing time for non-aggregate queries (query *B5*) is same as that of method DW. For queries with aggregate function, query processing time is the time required to traverse all records of DC to fetch the result.

Method MV: Query processing time is the time taken to explicitly invoke views placed on CenS and fetch the stored results.

Method MQDB: Since *MQDB* is placed on CenS and processing *synonymous* queries with no incremental data does not require invocation to DW, total processing time is: $T1 = (i) + (ii) + (iii)$.

Table 4 and Fig. 1 depicts the query processing time for *synonymous* queries without incremental data using DW, DC, MV and *MQDB*. For query processing using *MQDB*, variation in time to process the operations in *T1* is very insignificant and is noticeable with six significant decimal digits. Rounding off to four significant decimal digits gives same query processing time for all queries irrespective of query complexity.

- Though MV takes less time compared to using *MQDB*, the major drawback is that they are not supported by all DBMS. They require explicit invocation for execution and end user has to remember all views and the queries within it. Practically, it is difficult, and therefore, MV will not be considered further in the current literature.

b. *Processing synonymous queries with incremental data from DM*

Processing *synonymous* queries with incremental data is done using DM. Experimental results are recorded for processing *synonymous* queries with incremental data from DM placed on CenS and CldS using the following methods:

Method DW: Total query processing time is same as depicted in Table 4.

Method DC_DM_{incr}: Query processing time is same as method DC, respectively, with additional time required to generate incremental results from DM

Method MQDB_DM_{incr}: Total processing time is: $T1 + (iv) + (v)$

Table 5 and Fig. 2 depicts the process timings using these methods.

Table 4 Average query processing time for *synonymous* queries without incremental data using DW, DC, MV and *MQDB*

Location	Query	Average query processing time (s)				% reduction in query processing time		
		DW	DC	MV	<i>MQDB</i>	<i>MQDB</i> as compared to DW (%)	<i>MQDB</i> as compared to DC (%)	MV as compared to <i>MQDB</i> (%)
CenS	B1	0.8999	0.1865	0.0348	0.0415118	95.39	77.75	16.14
	B2	0.8985	0.1764	0.0350	0.0415110	95.38	76.47	15.66
	B3	0.9389	0.1895	0.0349	0.0415128	95.58	78.10	15.90
	B4	0.8714	0.1796	0.0351	0.0415106	95.24	76.89	15.42
	B5	0.8610	0.8610	0.0368	0.0415104	95.18	95.18	11.33
CldS	B1	4.5271	3.8137	–	0.0415118	99.08	98.91	–
	B2	4.5257	3.8036	–	0.0415110	99.08	98.91	–
	B3	4.5661	3.8167	–	0.0415128	99.09	98.91	–
	B4	4.4986	3.8068	–	0.0415106	99.08	98.91	–
	B5	4.4882	4.4882	–	0.0415104	99.08	99.08	–

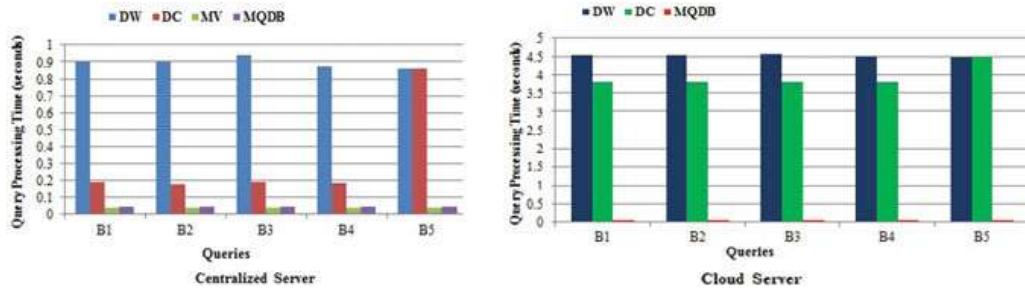


Fig. 1 *Synonymous* query processing without incremental data using DW, DC, MV and *MQDB*

4 Conclusion

Experimental results conclude that significant reduction in query processing time is achieved using *MQDB* as compared to using DW and DC whether they reside on CenS or on CldS. On an average, 95% time is reduced using *MQDB* with no incremental data as compared to using DW and 80% as compared to using DC when DW and DC are placed on CenS. While with CldS, average reduction in time using *MQDB* is 99% and 98% as compared to using DW and DC, respectively. For *synonymous* queries with incremental data, average reduction in time using *MQDB* is 87% and 67% as compared to DW and DC, respectively, on CenS. On CldS, average reduction in processing time is 17% and 50% as compared to using DW and DC, respectively.

Table 5 Average query processing time for *synonymous* queries with incremental data from DM using *MQDB*, DW and DC

Location	Query	Average query processing time (% reduction in query processing time using <i>MQDB</i> as compared to	
		DC_DM _{incr}	MQDB_DM _{incr}	DW (%)	DC_DM _{incr} (%)
CenS	B1	0.2885	0.1020	88.67	64.64
	B2	0.2862	0.1098	87.78	61.64
	B3	0.3090	0.1195	87.27	61.33
	B4	0.2846	0.1050	87.95	63.11
	B5	0.8610	0.1095	87.28	87.28
CldS	B1	7.5429	3.7292	17.62	50.56
	B2	7.5406	3.7370	17.43	50.44
	B3	7.5634	3.7467	17.95	50.46
	B4	7.5390	3.7322	17.04	50.49
	B5	8.1154	3.7367	16.74	53.96

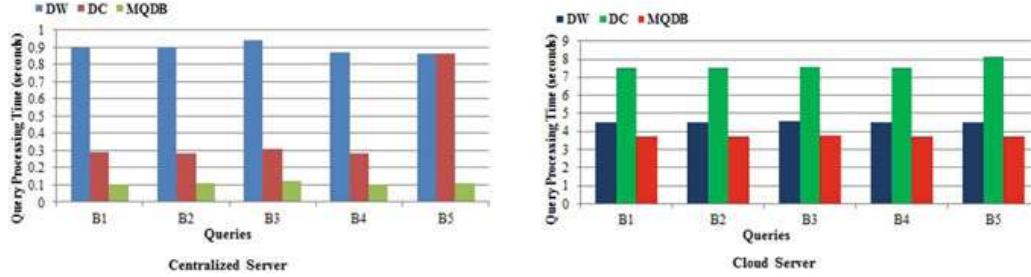


Fig. 2 *Synonymous* query processing using DW, DC and *MQDB* with incremental data

References

1. Harinarayan, V., Rajaraman, A., Ullman, J.: Implementing data cubes efficiently. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of data, Montreal, pp 205–216 (1996)
2. Agrawal, R., Gupta, A., Sarawagi, S.: Modeling multidimensional databases. In: Proceedings 13th International Conference on Data Engineering, pp. 232–243 (1997)
3. Datta, A., Thomas, H.: The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Decis. Support Syst. **27**(3), 289–301 (1999)
4. Deshpande, P., Agarwal, S., Naughton, J., Ramakrishnan, R.: Computation of multidimensional aggregates. In: Proceedings of the 22nd VLDB Conference, Mumbai, pp. 506–521 (1996)
5. Chun, S., Chung, C., Lee, J., Lee, S.: Dynamic update cube for range-sum queries. In: Proceedings of the 27th VLDB Conference, Rome (2001)
6. Shanmugasundaram, J., Fayyad, U., Bradley, P.: Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Deigo, pp. 223–232 (1999)

7. Gupta, A., Mumick, I., Subrahmanian, V.: Maintaining views incrementally. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, pp 157–166 (1993)
8. Gupta, A., Mumick, I.: Maintenance of materialized views: problems, techniques and applications. *Bull. Tech. Committee Data Eng. IEEE Comput. Soc.* **18**(2), 3–18 (1995)
9. Quass, D.: Maintenance Expressions for Views with Aggregation. *Views* (1996)
10. Zhuge, Y., Molina, H., Hammer, J., Widom, J.: View maintenance in a warehousing environment. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, pp. 316–327 (1995)
11. Gupta, A., Jagadish H., Mumick, I.: Data integration using self-maintainable views. In: *Advances in Database Technology—EDBT '96*, LNCENS, vol. 1057, pp. 140–144 (1996)
12. Mumick, I., Quass, D., Mumick, B.: Maintenance of data cubes and summary tables in a warehouse. In: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, Tuscon, pp. 100–111 (1997)
13. Chakraborty, S., Doshi, J.: Performance evaluation of materialized query. *Int. J. Emerg. Technol. Adv. Eng.* **8**(1), 243–249 (2018)
14. Chakraborty, S., Doshi, J.: Incremental updates using data warehouse versus data marts. In: 4th International Conference for Convergence in Technology (I2CT), IEEE Xplore® Digital Library (2018). (In Press)
15. Chakraborty, S., Doshi, J.: Deriving aggregate results with incremental data using materialized queries. *Int. J. Comput. Sci. Eng.* **5**(8), 835–839 (2018)
16. Chakraborty, S., Doshi, J.: Reducing query processing time for non-synonymous materialized queries with differed criteria. *Int. J. Nat. Comput. Res.* **8**(2), 75–93 (2019). (IGI Global Publishers)

Sense Scheduling for Robotics Cognitive Intelligence



Mahendra Bhatu Gawali  and Swapnali Sunil Gawali

Abstract Probably human is the most intelligent animal live on the earth as compared to other animals. Human is intelligent because nature gifted him an extraordinary speciality that is thinking ability. On the basis of this thinking ability, human can decide what is wrong or correct for him or others. Twenty-first century is known for the artificial intelligence. Where humans are forming artificial intelligence (robots) to complete the daily works. But, it is a long lasting challenge for researchers/scientist to introduce sense in artificial intelligence. We proposed a human thought process method (HTPM) to decide a certain thing out of many options. We recorded every person's thought while taking a decision. Scientifically, we concluded that our proposed HTPM method has given outstanding results when implemented in humanoid.

Keywords Cognitive intelligence · Machine learning · Artificial intelligence · Robotics · Humanoid

1 Introduction

Cognitive intelligence is the skill to handle reasoning, complex problem solving, applying various solutions, think abstractly, learn fast as compare to others and learn from experience [1]. A thin line difference between an artificial intelligence (AI) and cognitive intelligence (CI) is that AI is literate computers to solve complex problems. But, cognitive intelligence is about making computers solve complex problems as same as how humans solve problems [2]. The journey of AI and cognitive intelligence is summarized by Forbus [3]. Both the technology depends upon the common term known as data. The data is most valuable wealth in twenty-first century. Earlier data centers were used for to store the data which are generated by their users through

M. B. Gawali (✉) · S. S. Gawali
Sanjivani College of Engineering, Ahmednagar, India
e-mail: gawalimahendrait@sanjivanicoe.org.in

S. S. Gawali
e-mail: gawaliswapnaliit@sanjivani.org.in

any medium like social network, official mails, etc. In the first decade of twenty-first century, this concept was very popular to store the data at remote servers. This concept shown remarkable achievement in the field of technology. The data size store at server level is get increased in very rapid way. The challenge for scientist, researchers is to produce some meaningful information from this stored data. The birth of “big data analytic” is happened through this problem. On the other hand, the same data has used as fuel to produce new technology as artificial intelligence. An artificial intelligence is a field which inspired by the human life. The key role of artificial intelligence is to reduce human efforts and time by proposing machines which work as same as human works. Here, the point of observation is machine can work as same as human works. It sounds that to complete a certain work human needs to gain some skills by which the respective work will be complete within stipulated time with less efforts. This phenomenon is called as an intelligence. In human life, such intelligence observed by an experience or study. In other hand, only intelligence is not enough to complete the certain work, human needs good decision power to accept a specific task by observing its risk and rewards. In both intelligence and decision power situation, human always depends upon his/her previous experience or data which possessed by human. Cognitive intelligence is the field which is inspired by human brains. The most of the leading companies Microsoft, Yahoo, Amazon, and IBM are inspired to adapt the strengths of these technologies into technology by which machines can work parallel with human being but without pause [2, 4]. As human brain has taken lot of tasks as an input for completion, but decides which should be done first. The performance of such human brain has been measure in terms of task’s completion time or overall assigned tasks completion time with high accuracy. This is a challenge for scientist, researchers to propose a decision system through cognitive intelligence so that machine can take the best decision and work accordingly. The major objective of this research paper is to propose a human thought process method for cognitive intelligence in which machine can take a specific decision.

The major contributions in this paper are summarized as follows

- (1) Proposed a test to check human thought process.
- (2) Perform a live test for 75 students.
- (3) Recorded the opinion of each student during the test.
- (4) Summarized all opinion received from students
- (5) Proposed a novel human thought process method for humanoid.

The rest of the paper is organized as follows. Section 1 describes the introduction of cognitive intelligence and its issues especially scheduling. Section 2 focuses up on the related work of scheduling in cognitive intelligence. Section 3 deals with human thought process test. Section 4 deals with architecture of proposed system. Section 5 explains the propose methodology. Section 6 focuses on an evaluation of proposed scheduling approach. Finally, concluding remark with future directions are presented in Sect. 7.

2 Related Work

This section gives a brief review about the scheduling techniques proposed by scientist/researchers specifically for cognitive intelligence. Actually, very few literature have focused this scheduling issue for cognitive intelligence that related on specific areas such as wireless network and IoT. But none of the research has focused on fundamental scheduling issue in cognitive intelligence. Author Yingxu Wang has proposed a theoretical framework for cognitive informative based on the layered reference model of the brain (LRMB), the information representation by OAR model, natural intelligence (NI) and artificial intelligence (AI), autonomic computing (AC) and imperative computing, cognitive intelligence laws of software, and human perception process mechanism [5]. Cote and Miners [6] have proposed a model based on emotional intelligence and cognitive intelligence. Garrido et al. [7] have proposed a framework for scheduling problem in cognitive intelligence. This framework has combined emotional and intellectual factors to handle a particular situation. LoPresti et al. [8] have proposed a prototype which focused on interactive task guidance capabilities. Basu et al. [9] have proposed a cognitive scheduling model for an IoT applications in cloud computing environment. This model is based on bio-inspired approach to find an optimal solution for scheduling. Authors have combined genetic algorithm(GA) and ant colony optimization(ACO) for to produce a scheduling model. Zhang et al. [10] have proposed an optimal data transmission scheduling approach based on a deep Q-learning in cognitive vehicular networks. This approach is used to minimize transmission costs and at other side to utilize communication nodes and resources properly. Higgins [11] has elaborated his work as follow.

- The factors influencing scheduling decisions in small-batch manufacture and the role of humans in the scheduling process.
- The position of the human scheduler in hybrid intelligent decision-making processes.
- The Gantt charts have used as an interface for human–computer interaction in decision making.

Above listed literature has elaborated the existing work in cognitive intelligence. But, still there is a scope for

3 Human Thought Process Test

4 Proposed System Architecture

5 Proposed Methodology

6 Result and Discussion

7 Conclusion

8 Conclusion

Acknowledgements We would like to thank Management of Sanjivani College of Engineering, Kopargaon, India for providing the infrastructure to carry out the proposed research work.

Competing Interests The authors declare that they have no competing interests.

References

1. T. E. of Encyclopaedia Britannica: Cognition thought process. Accessed 19 Aug 2019. [Online]. Available: <https://www.britannica.com/topic/cognition-thought-process>
2. Benjamins, R.: Artificial Intelligence vs Cognitive Computing: What's the difference? Accessed 19 Aug 2019. [Online]. Available: <https://business.blogthinkbig.com/artificial-intelligence-vs-cognitive/>
3. Forbus, K.D.: AI and Cognitive Science: The Past and Next 30 Years. Accessed 19 Aug 2019. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1756-8765.2010.01083.x>
4. Jones, N.: The Learning Machines. Accessed 19 Aug 2019. [Online]. Available: https://www.nature.com/news/polopoly_fs/1.14481!/menu/main/topColumns/topLeftColumn/pdf/505146a.pdf
5. Wang, Y.: The theoretical framework of cognitive informatics. Int. J. Cogn. Inf. Nat. Intell. (IJCINI) **1**(1), 1–27 (2007)
6. Cote, S., Miners, C.T.: Emotional intelligence, cognitive intelligence, and job performance. Adm. Sci. Q. **51**(1), 1–28 (2006)
7. Garrido, L., Brena, R., Sycara, K.: Cognitive Modeling and Group Adaptation in Intelligent Multi-agent Meeting Scheduling (1996)
8. LoPresti, E.F., Simpson, R.C., Kirsch, N., Schreckenghost, D., Hayashi, S.: Distributed cognitive aid with scheduling and interactive task guidance. J. Rehabil. Res. Dev. **45**(4), 505–522 (2008)
9. Basu, S., Karuppiah, M., Selvakumar, K., Li, K.-C., Islam, S.H., Hassan, M.M., Bhuiyan, M.Z.A.: An intelligent/cognitive model of task scheduling for iot applications in cloud computing environment. Future Gener. Comput. Syst. **88**, 254–261 (2018)

10. Zhang, K., Leng, S., Peng, X., Pan, L., Maharjan, S., Zhang, Y.: Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE Internet Things J.* **6**(2), 1987–1997 (2018)
11. Higgins, P.G.: Interaction in hybrid intelligent scheduling. *Int. J. Hum. Factors Manuf.* **6**(3), 185–203 (1996)

Author Biographies



Mahendra Bhau Gawali Mahendra Bhau Gawali received his B.E. degree in 2008, M.E. degree in 2013 and Ph.D. degree in 2019 from University of Mumbai, MS, India. Currently he working as an associate professor in IT department of Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, Pune, MS, India. His area of interests are Cognitive Intelligence, Artificial Intelligence, Cloud Computing, Optimization etc.



Swapnali Sunil Gawali Swapnali S. Gawali is working as an assistant professor in the Department of Information Technology of Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, Pune, MS, India. She has completed her B.E. in 2014 and ME from 2016 respectively from Savitribai Phule Pune University, Pune. Her area of interest are Data Mining, Artificial Intelligence etc.

Feature Selection Using Ant Colony Optimization and Weighted Visibility Graph



Leena C. Sekhar and R. Vijayakumar

Abstract Feature selection technique has an important role in the elimination of unrelated features and noises from the high-dimensional data. It simplifies and enhances the quality of dataset by selecting salient features. Good feature selection algorithm leads to accurate classification. Feature selection of high-dimensional dataset addresses the problem with redundancy, accuracy, and computational complexity. Ant colony optimization (ACO) is a modern algorithm for feature selection. It is an evolutionary algorithm inspired by the foraging behavior of ants. This paper proposes the technique of weighted visibility graph and ACO method for feature extraction and feature selection. In this method, high-dimensional dataset is converted into the complex network and after extracting eight well-suited features from the dataset, feature selection is performed. Naive Bayes method is utilized to classify the selected features. Experimental results indicate that the classification accuracy is more accurate using the proposed method.

Keywords Visibility graph · Ant colony optimization · Feature selection · Feature extraction

1 Introduction

The high-dimensional dataset causes severe problems in the performance of machine learning problems. Feature selection (FS) simplifies and enhances the quality of a dataset by selecting salient features and reduces the complexity of the overall learning process. FS is a promising issue in the area of machine learning, cluster analysis,

L. C. Sekhar (✉)
MES College, Marampally, Aluva, Kerala, India
e-mail: leena@mesmarappally.org

R. Vijayakumar
School of Computer Science, Mahatma Gandhi University, Kottayam, Kerala, India
e-mail: vijayakumar@mgu.ac.in

pattern recognition, image classification, and image retrieval. It is a preprocessing technique because irrelevant features and noises are eliminated from the original data. Significant information about the problem is represented by the relevant features and hence it takes the main role in classification problem and the accuracy of the classifier is reduced by the irrelevant features [1]. Feature selection method has the following benefits:

1. Improved data visualization
2. Data understanding
3. Reduced storage requirements
4. Reduced training time
5. Improved prediction performance.

There are two different approaches in FS, (1) Filter approach (2) Wrapper approach. Feature subsets are evaluated by utilizing the distance, information, dependency, and consistency measures in the filter method. In the Wrapper approach, the classifier is utilized to evaluate the feature subset. Wrapper method generally produces better results than the filter method, but it is a more expensive technique. So filter methods are used in many cases to obtain the reduced set of features [2]. The subsequent optimization techniques of genetic algorithm, evolutionary algorithm, and ant colony algorithm are incorporated with the Wrapper approach for feature subset selection [3].

The filter-based feature selection approaches have been rewarded much consideration to the computational interval and normally are quicker, though the unconfirmed Wrapper approaches measure the usefulness of features based on the classifier performance. The medical image classification area is very challenging to identify the relevant features from the high-dimensional dataset. One of the main causes of death among women is breast cancer and more than 1.5 million women are affected by this every year [4]. Selection of data taken from patient records will be useful to experts for efficient diagnosis but the selection of possible features by the existing methods deals with the following issues:

- Redundancy
- Computational complexity
- Increased computational cost
- Minimum classification accuracy.

Utilization of effective algorithm reduces the drawbacks and improves the performance of the existing method. In this proposed work, visibility graph (VG) construction is used for feature extraction and feature selection is done by using ACO. The classification process is done by Naive Bayes technique. Optimal features [5] are selected based on the measures of local features and also the entire performance of subsets by the method of ACO [6].

The remaining sections of the paper are organized as follows: Sect. 2 explains the review of literature for feature selection and classification techniques. The proposed technique of feature selection and implementation using VG is detailed in Sect. 3. Section 4 gives the performance analysis of the proposed method and the conclusion is given in Sect. 5.

2 Review of Literature

In 1992, M. Dorigo et al. introduced ACO [7]. The technique of ACO depends on the well-coordinated and self-organized behavior of ants. Ants communicate with each other using a volatile chemical substance known as pheromone. Shortest route [8] among the nest and source of food can easily be identified by the process of Pheromone Update and path construction. ACO is based on the concept of pheromone laying and following by the real ants. More pheromone on the path implies the increase in the probability of the same path being followed by other ants. This self-organizing behavior helps real ants to find the smallest path for their foraging activity. When the food source exhausts, the returning ants do not deposit any pheromone and the volatile pheromone will slowly evaporate. This negative feedback behavior helps ants to deal with changes in their environment. This strategy of real ants to find the shortest path from their nest to the food source can thus be effectively utilized in feature selection. ACO has the following benefits in feature selection [9]: (1) positive feedback, (2) utilization of long-term memory, and (3) local and global searching competency.

The process of feature selection [10] plays an important role in the area of machine learning. The selection of a feature subset with minimum length and maximum accuracy is the motive of the feature selection method. A reformed ant colony optimization algorithm was proposed to detect the feature set which is more important to the classification process. New heuristic information component is incorporated by the algorithm to improve the classification accuracy. The suggested technology was applied to the capsule endoscope images under the multiclass classification problem and the provided algorithm effectively detected the required feature subset. The comparative study proved that the selected algorithm improved the accuracy, sensitivity, and computational time.

Clustering-based ACO [11] was mostly utilized in the field of data mining and it focused mainly on clustering but the area of clustering still requires more concentration. So, two medoid-based ACO clustering algorithms were suggested, where the first algorithm predicted the optimal medoid set by ACO procedure and the second algorithm chose the number of clusters by the automatic selection method. The algorithm was compared with the conventional clustering algorithms by synthetic

datasets and real-world datasets. Heuristic information can be included in the algorithm to improve the research in a further direction. An Intrusion Detection System (IDS) regularly handled the huge quantity of data traffic that comprises redundant and unrelated features which influenced the performance of the IDS negatively. The component of IDS is utilized in network security to protect the data from the unauthorized users [12] and the technique of feature selection eliminates the unrelated features of IDS. This work reviewed the feature selection algorithms for IDS. It points to the fact that classification accuracy is dependent on the effective feature selection method.

The relevant subset of features [13] is predicted from the datasets by the application of the optimized feature selection method. This work did an in-depth analysis of the method of feature selection utilizing Artificial Bee Colony (ABC) Algorithm for classification approach. The results indicate that high classification accuracy was achieved with a minimum number of features. Microarray datasets are generally preferred in dimension reduction method because of the increased number of features [14]. For managing the high-dimensional datasets, a bionic optimization algorithm dependent on dimension reduction method was proposed, which was named as Ant Colony Optimization-Selection (ACO-S). Non-significant features (genes) are filtered from high-dimensional space by modified ant system and the required genes are selected at the next stage. A finer classification accuracy was achieved with the selected algorithm.

In [3], a selection technique dependent on a modified binary coded ant colony optimization algorithm (MBACO) combined with a genetic algorithm (GA) is proposed. The technique provided better results and the results are adaptive and robust compared to the existing methods. But it has less computational efficiency which can be improved by further modification in the algorithm.

In the field of bio-informatics [15] enhancement of biologically relevant design from gene expression becomes an interesting topic and this is also useful in research about the genetic diseases. The process of medical diagnoses takes more time and it involves a higher expense. So a minimum number of genes are selected by modified Artificial Bee Colony Algorithm (ABC) with improved predictive accuracy for the classification process of cancer. The results are compared with other work by the same dataset. The accurate classification was achieved by the selected subset of genes which was indicated by the evaluation results.

The process of feature selection in [16] leads to fast classification by identifying the relevant information by combining the features of ACO algorithm and ABC algorithm leading to AC-ABC Hybrid.

The technique of pattern recognition [16] and machine learning considered feature selection as an important preprocessing step. Hence, this work projected an innovative feature selection method which is dependent on graph clustering technique and ant colony optimization. Total feature sets are projected as a graph and the features

are categorized into the variety of clusters by utilizing the community detection algorithm. Finally, feature subsets are selected by the innovative search strategy which is based on the ACO technique.

Epilepsy detection [17] of the brain from the EEG signal is considered as a difficult task because the signals comprise lot of varying information about the practical behavior of the brain and make it tough to differentiate the complex network of EEG signals. In [17] a new epilepsy detection technique is proposed by converting the time series data into a weighted visibility graph, an effective tool for the analysis of the time series [18].

In high-dimensional datasets, the classification problem is a challenging and computationally expensive process for feature selection [19]. This is because the classification accuracy obtained using the reduced feature subset is greater than the original feature set. A hybrid feature selection algorithm proposed in [20] shows a significant reduction in the dimension of gene data.

A hybrid method in [21] combines ABC optimization techniques with evolution algorithm for the classification task and the results show selection of good features for classification. The performance of Anisotropic Diffusion filtering algorithm is further improved using ACO algorithm by selecting optimal parameters and thus reduces the residual speckle of noisy images in [22].

3 Proposed Work and Experiment

This section details the proposed work. A high-dimensional dataset of breast cancer Wisconsin (prognostic) (wpbc) and breast cancer Wisconsin diagnostic (wdbc) are considered from the UCI data repository for the proposed work. Wpbc contains 32 attributes, 398 instances, and two classes and wdbc with 34 attributes, 196 instances, and 2 classes. The work is categorized into three major sections:

1. Feature extraction
2. Feature selection
3. Feature classification.

The flow diagram of the proposed method is shown in Fig. 1.

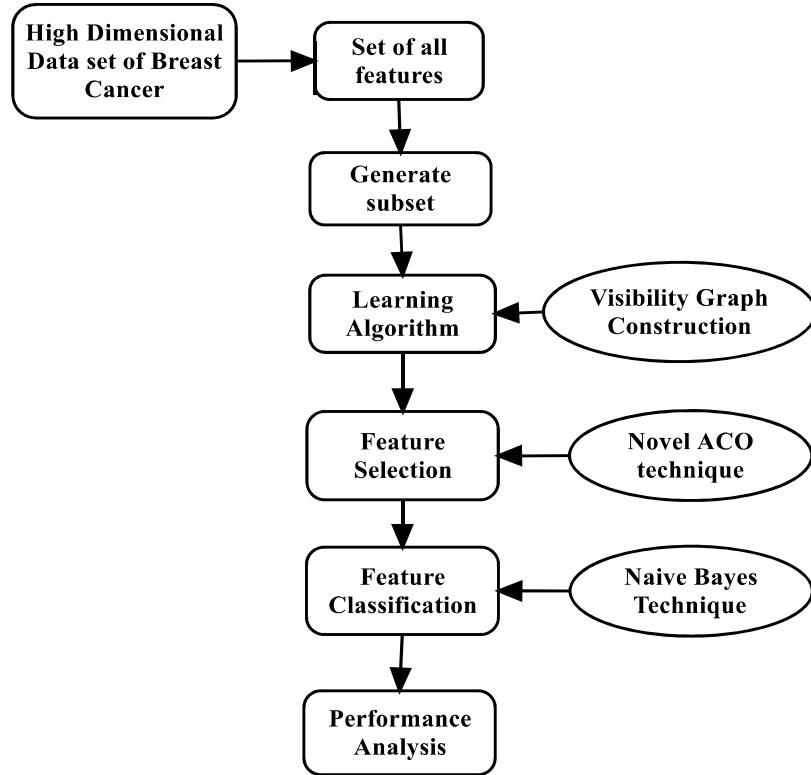


Fig. 1 Flow of the proposed system

3.1 Feature Extraction

A high-dimensional dataset of breast cancer is selected for the feature extraction and the data is converted into weighed visibility graph as follows. Initially, the attributes required for the graph construction are derived from the overall dataset. Three nodes a, b, and c are selected for connectivity check. Weight between the nodes is checked by utilizing the formula mentioned in Algorithm 1. If the connectivity is detected between the nodes, weight calculation is carried out on the subsequent steps. Then, the graph is constructed based on the calculation for feature extraction. Weight calculation is not carried out if the connection between the nodes fails. Algorithm for optimal visibility graph-based feature extraction is given below.

Algorithm: 1 (Optimal Visibility Graph-based Feature Extraction)**Input:** Input Dataset (CD_{vals})**Output:** Visibility Graph (opt_{vg})**Procedure:**Step1: (CD_{vals}) \leftarrow overall dataset

Step2: Let N represents attributes in Dataset

Step3: Let T (t_t) be time series of attributes

Step4: Let a, b, c be the nodes to be checked

Step5: Let W(CD_{Edge}) be Weight between nodesStep6: (CD_{feat}) \leftarrow Attributes for Disease predictionStep7: While $((CD_{feat})! = null)$ doStep8: (CD_E) $\leftarrow \sum_1^{N-2} T(t_a) + (T(t_b) - T(t_c)) ; a < c < b$ Step9: (CD_{Ed}) $\leftarrow \sum_1^{N-2} \frac{t_c - t_a}{t_b - t_a}$ Step10: (CD_{Edge}) $\leftarrow (CD_E) * (CD_{Ed})$ Step11: if T (t_c) $<$ (CD_{Edge})Step12: $W(CD_E) = \frac{T(t_b) - T(t_a)}{t_b - t_a} ; a < b$ Step13: $W(CD_{Edge}) = \tan^{-1}(W(CD_E))$ Step14: $(CD_{feat})_{a \rightarrow b} = W(CD_{Edge})$ Step15: (opt_{vg}) $\leftarrow (CD_{feat})_{a \rightarrow b}$

Step16: End While

In the process of classification, feature extraction plays an important role. Identifiable measurements obtained from the high-dimensional data using the visibility graph are represented by the feature. The process of feature extraction compresses the large quantity of breast cancer dataset into the relevant feature vector set with minor loss in the information. So, it helps the simple and fast classification process. In this work, eight features extracted include:

1. Average weight of each node
2. Closeness centrality
3. Degree of node
4. Degree centrality score
5. Average path length
6. Link density
7. Average vertex eccentricity
8. Graph radius.

3.2 Feature Selection

Subset of original features from the breast cancer dataset is selected by the ant colony optimization algorithm. Features extracted from the previous step are given as the input to generate the subset of features. Dynamic parameters, like number of ants, initial pheromone values, are initialized at the first step. Solution creation is the next step of the algorithm and each ant produces partial solution to update the pheromone. Each ant selects an edge during the selection creation phase to update the pheromone of the same traversed edge. If the selected number of features does not make further movement on their path, solution construction of the ant is considered as completed. The solution is evaluated and updated at the final step. Algorithm 2 gives this procedure. Features thus selected are classified using Naive Bayes classifier which is shown in Algorithm 3. Generated feature subsets are given to the classifier to predict the best solution.

Algorithm: 2 (Feature Selection from Visibility Graph by Ant Colony Optimization-VGACO)

Step1: NA($number_{ants}$) be number of ants
 Step2: Mn(CD_{feat}) be minimum value from feature
 Step3: Mx(CD_{feat}) be maximum value from feature
 Step4: Av(CD_{feat}) be average value from feature
 Step5: Thr(CD_{feat}) be threshold value from feature
 Step6: Feat($Soln_{feat}$) be selected Feature
 Step6: For NA($number_{ants}$) = 1 to (CD_{feat}).size()
 Step7: For c=1 to (CD_{feat})
 Step8: For d=1 to (opt_{vg})
 Step9: (CD_{feat}) \leftarrow c
 Step10: Mn(CD_{feat}) = Min(CD_{feat})
 Step11: Mx(CD_{feat}) = Max(CD_{feat})
 Step12: Av(CD_{feat}) = Avg(CD_{feat})
 Step13: Thr(CD_{feat}) = (Mn(CD_{feat}) +
 Mx(CD_{feat})) / Av(CD_{feat})
 Step14: End For
 Step15: if (Thr(CD_{feat}) > c)
 Step16: Feat($Soln_{feat}$) \leftarrow Update Solution
 Step17: End if
 Step18: End For
 Step19: Feat($Soln_{feat}$) \leftarrow Feat($Soln_{feat}$) + Feat($Soln_{feat}$)
 Step20: End For

Algorithm: 3 (Classification based on VG Selected Feature)

Step1: Let N, R be the Labels in (CD_{vals})

Step2: Let $CN(N_{label})$ be count of 'N' Label

Step3: Let $CN(R_{label})$ be count of 'R' Label

Step4: Let $Class(N_{Label})$ be the result of classification

Step5: Let $Class(R_{Label})$ be the result of classification

Step6: Let $Class(Class_{Label})$ final result

Step6: $P(C_N) = CN(N_{label}) * \text{Feat}(Soln_{feat})$

Step7: $P(C_R) = CN(R_{label}) * \text{Feat}(Soln_{feat})$

Step8: $Class(N_{Label}) \leftarrow \text{Feat}(Soln_{feat}), CN(N_{label})$

$Class(N_{Label}) :$

$$\text{Count}(Soln_{feat}) \leftarrow \sum_1^{\text{Feat}(Soln_{feat})} \text{count}(\text{Feat}(Soln_{feat}))$$

$$\text{Count}(Soln_{feat}) = \text{Count}(Soln_{feat})^2 + \text{Count}(Soln_{feat})^4$$

$$Class(N_{Label}) = \frac{\sum \text{Count}(Soln_{feat})}{CN(N_{label})}$$

Step9: $Class(R_{Label}) \leftarrow \text{Feat}(Soln_{feat}), CN(N_{label})$

$Class(R_{Label}) :$

$$\text{Count}(Soln_{feat}) \leftarrow \sum_1^{\text{Feat}(Soln_{feat})} \text{count}(\text{Feat}(Soln_{feat}))$$

$$\text{Count}(Soln_{feat}) = \text{Count}(Soln_{feat})^2 + \text{Count}(Soln_{feat})^4$$

$$Class(R_{Label}) = \frac{\sum \text{Count}(Soln_{feat})}{CN(R_{label})}$$

Step10: if $Class(N_{Label}) \geq Class(R_{Label})$

Step11: $Class(Class_{Label}) \leftarrow 'N'$

Step12: else

Step13: $Class(Class_{Label}) \leftarrow 'R'$

Step14: End if

4 Performance Analysis

Here the result of the proposed VGACO technique is discussed. For evaluating the performance of these methods, the datasets wpbc and wdbc are considered. This includes different attribute information such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. Also, various performance measures like precision, recall, specificity, and F1-score are used and results are compared with the existing methods of classification.

4.1 Precision, Recall, Specificity, and F1-Score

The measures of precision, recall, and specificity are commonly used in evaluating the performance of the classification techniques used in many applications. In this paper, these measures are used for evaluating the exactness of classifier. Generally, precision is defined as the positive predictive value that offers highly relevant results to the classification. Precision is evaluated as

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (1)$$

where TP is true positive and FP is false positive. Similarly, recall is defined as the measure of sensitivity, which is used to estimate the relevant results during classification. It is calculated as follows:

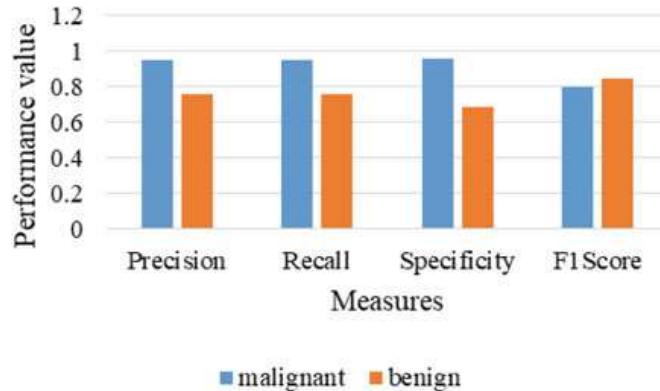
$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (2)$$

where FN is false negative. A true negative is termed as specificity which evaluates the fraction of negatives that are appropriately recognized.

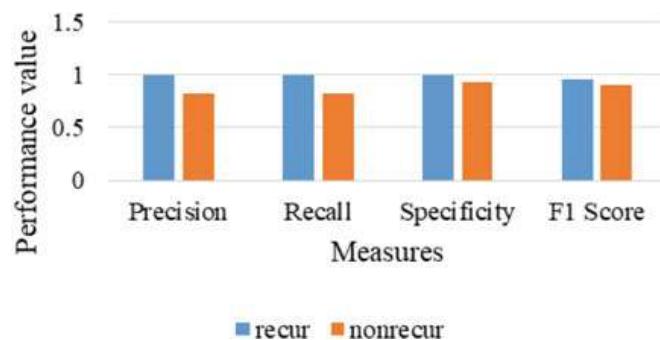
$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (3)$$

where TN is the amount of true negatives that are properly classified and FP gives the amount of false positives that incorrectly classify the disease. The F1 score is considered as the harmonic average of precision and recall. The F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. Figure 2 shows the various measures with respect to different disease diagnosis.

Fig. 2 **a** Precision, recall, specificity, and F1-score of wdbc dataset. **b** Precision, recall, specificity, and F1-score of wpbc dataset

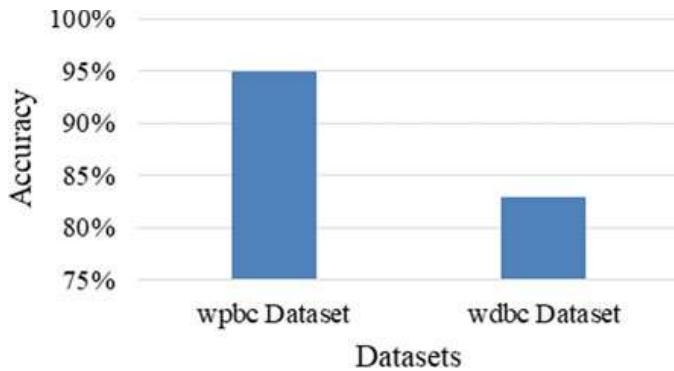


(a). Precision, recall, specificity and F1-score of wdbc dataset



(b). Precision, recall, specificity and F1-score of wpbc dataset

Fig. 3 Classification accuracy of the two dataset



4.2 Accuracy

The efficiency of classifier is estimated by using accuracy, which is determined based on the values of sensitivity and specificity. The overall performance of the classification system is determined by the measure of accuracy, which is calculated as follows:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{(\text{TN} + \text{TP} + \text{FN} + \text{FP})} \quad (4)$$

Figure 4 compares the accuracy of the proposed classification technique with respect to wpbc and wdbc datasets. Here, the evaluation is performed by calculating the number of attributes, extracted features, and selected features. This analysis stated that the accuracy of proposed classifier is increased by using visibility graph construction and ACO techniques. The number of features extracted and selected is illustrated in Fig. 5. Then, the existing and proposed classification techniques such as Support Vector Machine (SVM), Neural Network (NN), Random Forest (RF), and K-Nearest Neighbor (K-NN) techniques are compared in Fig. 6. This result stated that the proposed classification technique shows good results when compared with other methods. Classification accuracy of two different datasets is indicated in Table 1 and the accuracy of different classifiers is indicated in Table 2 (Fig. 3).

Fig. 4 Features extracted and selected

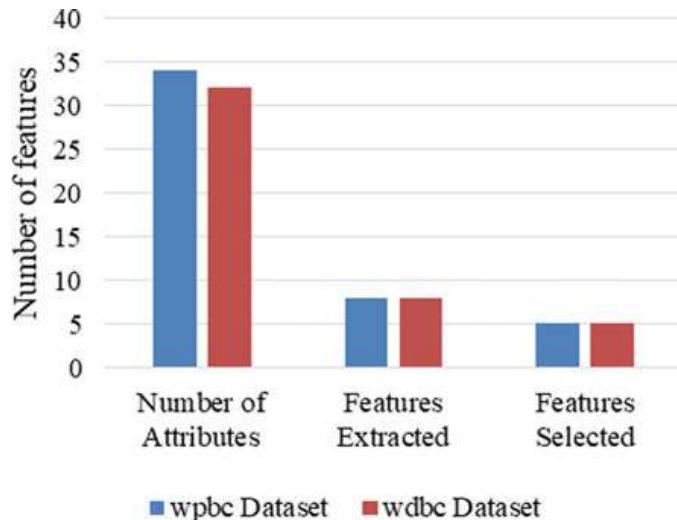


Fig. 5 Accuracy of existing and proposed classification techniques

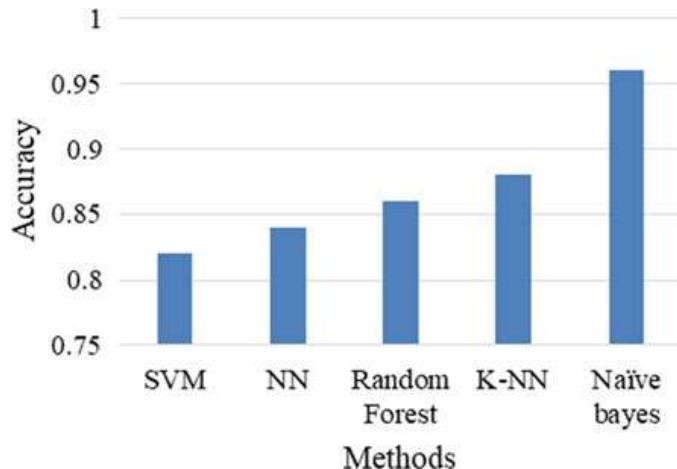
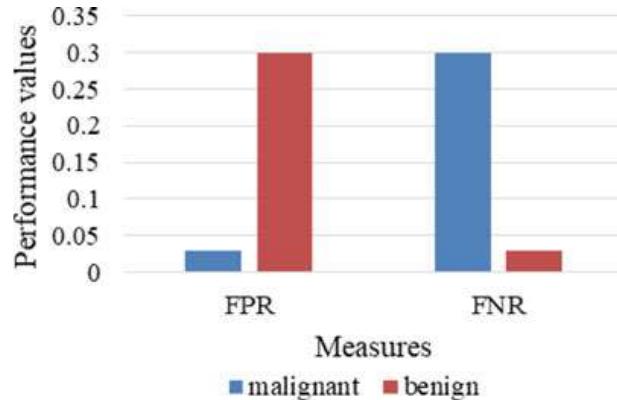
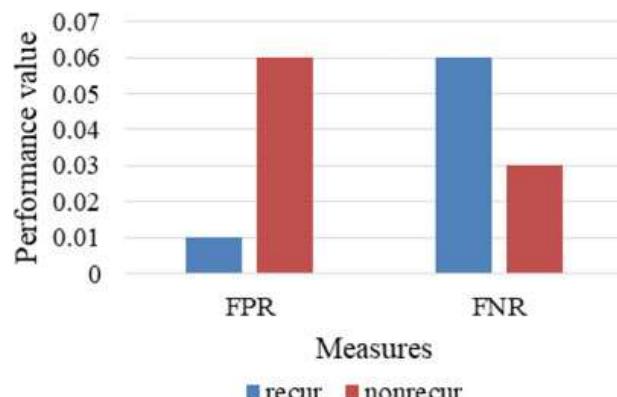


Fig. 6 **a.**False positive and false negative rate for wdbc dataset. **b** False positive and false negative rate for wpbc dataset



(a). False positive and false negative rate for wdbc dataset



(b). False positive and false negative rate for wpbc dataset

Table 1 Classification accuracy of wpbc dataset and wdbc dataset

Name of dataset	Number of attributes	Features extracted	Features selected	Classification accuracy (%)
wpbc dataset	34	8	5	95
wdbc dataset	32	8	5	83

Table 2 Classification accuracy of different classifiers

Name of classifier	Accuracy
SVM	0.82
NN	0.84
Random Forest	0.86
K-NN	0.88
Naïve Bayes	0.96

4.3 False Positive Rate(FPR) and False Negative Rate(FNR)

False positive is examined for the disease identification, in which a positive result is obtained even when the person does not have any disease. The negative results obtained here are represented as positive. It is termed as false positive error. The FPR is decreased when compared with the existing technique. So the accuracy rate is improved in classification. The analysis shows the negative results when the person is affected by the diseases. In the proposed work, the false negative rate is minimized when compared to existing methods. So it improves the classification accuracy. Figure 6a, b represents the FPR and FNR using the two dataset. This result shows that the proposed classification technique provides better and more accurate results for both diagnoses.

5 Conclusion

The proposed work projects a novel technique, VGACO, for feature extraction and feature selection. High-dimensional dataset is reformed into the weighted visibility graph for feature extraction. Extracted features are selected by the ant colony optimization technique. Selected features are classified by Naïve Bayes classifier. The efficiency of this method analyzed using the performance measures of precision, recall, accuracy, FPR, and FNR assures improved classification accuracy.

References

1. Kashef, S., Nezamabadi, P.H.: Introducing a new version of binary ant colony algorithm to solve the problem of feature selection. *Sci. Inf. Database* **12**(2), 127–134 (2015)
2. Dadaneh, B.Z., Markid, H.Y., Zakerolhosseini, A.: Unsupervised probabilistic feature selection using ant colony optimization. *Expert Syst. Appl.* **53**, 27–42 (2016)
3. Wan, Y., Wang, M., Ye, Z., Lai, X.: A feature selection method based on modified binary coded ant colony optimization algorithm. *Appl. Soft Comput.* **49**, 248–258 (2016)
4. Akay, M.F.: Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Syst. Appl.* **36**, 3240–3247 (2009)
5. Deriche, M.: Feature selection using ant colony optimization. In: 6th International Multi-conference Systems, Signals and Devices. SSD'09, pp. 1–4 (2009)
6. Moradi, P., Rostami, M.: Integration of graph clustering with ant colony optimization for feature selection. *Knowl. Based Syst.* **84**, 144–161 (2015)
7. Dorigo, M., Stutzle, T.: Ant colony optimization. Encyclopedia of Machine Learning (2010)
8. Ariyasingha, I., Fernando, T.: Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem. *Swarm Evol. Comput.* **23**, 11–26 (2015)
9. Abd-Alsabour, N., Randall, M.: Feature selection for classification using an ant colony system. In: Sixth IEEE International Conference on e-Science Workshops (2010)

10. Mohammed, S.K., Deeba, F., Bui, F.M., Wahid, K.A.: Feature selection using modified ant colony optimization for wireless capsule endoscopy. In: Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual, pp. 1–4 (2016)
11. Menéndez, H.D., Otero, F.E., Camacho, D.: Medoid-based clustering using ant colony optimization. *Swarm Intell.* **10**, 123–145 (2016)
12. Balasaraswathi, V.R., Sugumaran, M., Hamid, Y.: Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. *J. Commun. Inf. Netw.* **2**, 107–119 (2017)
13. Schiezaro, M., Pedrini, H.: Data feature selection based on Artificial Bee Colony algorithm. *EURASIP J. Image Video Process.* **47** (2013)
14. Li, Y., Wang, G., Chen, H., Shi, L., Qin, L.: An ant colony optimization based dimension reduction method for high-dimensional datasets. *J. Bionic Eng.* **10**, 231–241 (2013)
15. Moosa, J.M., Shakur, R., Kaykobad, M., Rahman, M.S.: Gene selection for cancer classification with the help of bees. *BMC Med. Genomics* **9** (2016)
16. Shunmugapriya, P., Kanmani, S.: A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm Evol. Comput.* **36**, 27–36 (2017)
17. Supriya, S., Siuly, S., Wang, H., Cao, J., Zhang, Y.: Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE Access* **4**, 6554–6566 (2016)
18. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuno, J.C.: From time series to complex networks: the visibility graph. *Proc. Natl. Acad. Sci. U. S. A.* **105**(13), 4972–4975 (2008)
19. Dash, M., Liu, H.: Feature selection for classification. *Intell. Data Anal.* **1**(3), 131–156 (1997)
20. Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., Gao, Z.: A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* **256**, 56–62 (2017)
21. Zorarpaci, E., Özal, S.A.: A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Syst. Appl.* **62**(C), 91–103 (2016)
22. Bhateja, V., et al.: Ant colony optimization based anisotropic diffusion for despeckling of SAR images. In: International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision making, pp. 389–396. Springer, Cham (2016)

From Generic to Custom: A Survey on Role of Machine Learning in Pharmacogenomics, Its Applications and Challenges



Sana Aimani and Kiran Kumari Patil

Abstract Current advancements in medical sciences and pharmacogenomics are focusing on efficient, faster, and economic ways of drug delivery. On the other hand, big data analytics and machine learning are pushing the boundaries of human intelligence. Our aim is to bridge this gap between medical science and engineering by providing solutions that adhere to the requirements. Fields like remote robotic operations or artificial intelligence (AI) system for disease diagnosis and precision medicine are few that bridge this gap. Our proposed work in the field of precision medicine is an effort to contribute for making society healthier and more sustainable by reducing costs as well as reducing iatrogenic diseases by adopting technology advancements. The main focus of this survey paper is to understand the trends in field of biology, particularly in pharmacogenomics for better treatment of diseases by effective medication considering diabetes as an example. The paper also discusses issues related to application of machine learning in genomic data.

Keywords Pharmacogenomics · Diabetes · Precision medicine · Gene drug · Machine learning

1 Introduction

In the field of medicine, diabetes is one of the most widely spread chronic diseases. It is a condition that causes high blood sugar levels which has long-term effect. In India, about one in five Indian adults suffer from diabetes. But in coming few years the rate is expected to skyrocket to as many as one in two and hence India is known to be the capital of diabetes in the world [15]. Precision medicine, on the other hand, is

S. Aimani (✉) · K. K. Patil
School of Computing and Information Technology, REVA University, Bangalore, India
e-mail: sana.myin@gmail.com

K. K. Patil
e-mail: kirankumari@reva.edu.in

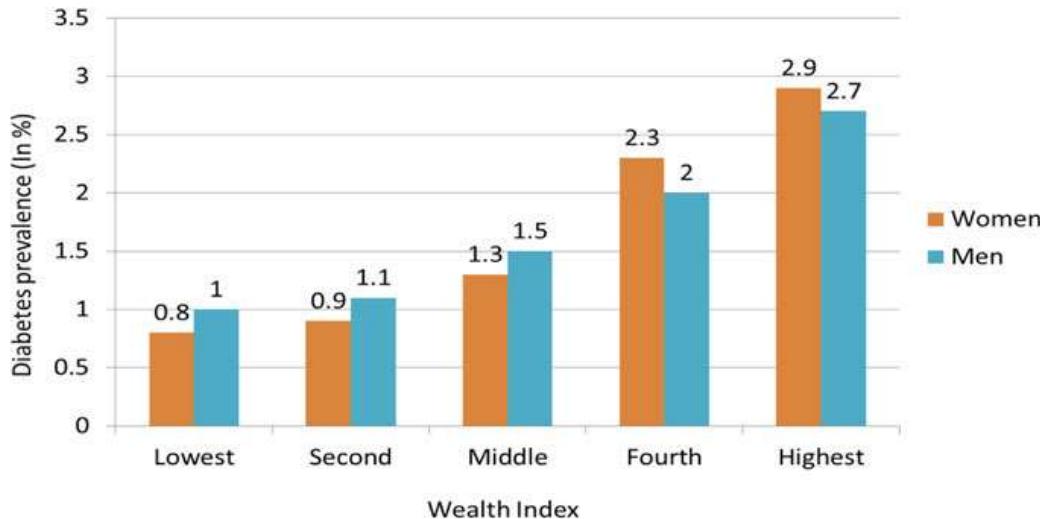


Fig. 1 Diabetes prevalence in India, By Wealth Brackets

an emerging approach in treating diseases by analyzing the gene sequences of individual along with their lifestyle and environmental changes. It does not comply with one-size-fits-all approach where the prevention strategies and treatment procedure are based on the average individual without any much consideration to individual difference [22]. Genome-wide association study (GWAS) has now transformed the medical practice. It has drifted to provide patient-centered care by providing right diagnostics tailored to the individual along with different therapeutic strategies [4] (Fig. 1).

Precision medicine has two facets, one is prediction of the disease occurrence and preventing it and the second is the treatment of the disease by effective medication, which is the primary focus of the survey. The survey also discusses about the purpose of machine learning in precision medicine. The paper is further sub-categorized into genomics, pharmacogenomics, and trends in applications of machine learning in precision medicine where the genotype of the person is assessed to give effective medication.

2 Genomics

2.1 Understanding the Disease

The genetic information is found to be in DNA molecules which are further categorized into distinct genes which describe how larger set of molecules could be created. The genes could be further classified into exons (coding) and introns (non-coding) genes. The exons help in describing how the amino acid (protein) molecules are built while introns describe the building of RNA molecules [12]. Many of these genes are

very crucial for life and well-being but some could be removed completely without causing any harm to the body [2].

Genomics has been applied to studying of diseases ranging from depression to cancer to diabetes. Typically, diabetes was known to have two variants, that is, Type 1 and Type 2, but now we know many more variants like maturity onset diabetes of the young (MODY) which respond very differently to the available drugs and their complication rates are very different due to the genetic variants. If we take leukemia as an example, it was understood to have one or two diseases but after molecular understanding the tumors, it was found that there are 75 distinct diseases and all of which has separate treatments. Since these identified subsets of the disease were tackled differently and efficiently, leukemia has reached very high survival rates. Similarly, the cancer patients, who were earlier treated by their anatomical cancer type like bone, blood, lung, etc., are now treated by what is called precision oncology. The genome sequences of cancer patients are compared to identify the mutations present in the tumorous cells which are not present in the healthier part of the individual's genome. This makes it easier for identifying the drugs which are specific to these affected genes. The technology of DNA sequencing is now faster and less expensive after the completion of Human Genome Project. This genomic-driven research model can be used for other diseases as well [23].

2.2 Challenges

The difficulties in genomic literacy stretch out from the laboratory to examining room. The doctors not only have limited knowledge about the tests to be ordered and the procedure to order but they also do not know how to interpret the test's result. The results just seem to be meaningless and insane. The physicians will need good training and support to interpret the data. Besides, genomics is actually the real big data when compared to astronomy, weather simulations, and social media. With this multiplying genomic information, there is a need for rich electronic framework to allow the sharing of genomics data among the research centers or researches (institutions, physicians, and individuals) so that better clinical care can be rendered. The future clinical decision support system can become an authoritative database to get the right answer.

EMRs are used only by clinicians and hospital industry which are unlike the electronic health record (EHR) that contains the patient's profile which can be shared among all participant hospitals and clinics therefore ensuring interoperability which is critical for big data in precision medicine [9]. Another challenge is storing of variants present in individual and family members in EMR [11]. The EMR should

1. Store variants such as CNVs and SNPs in discrete and computable format.
2. Be interoperable to enable easy data transfer and should be updated frequently.
3. Provide efficient decision support engines.
4. Incorporate visual components for easier interpretation [3].

The Fast Health Interoperability Resources (FHIR) standard reduces the issue addressed [24].

2.3 *Data Sharing*

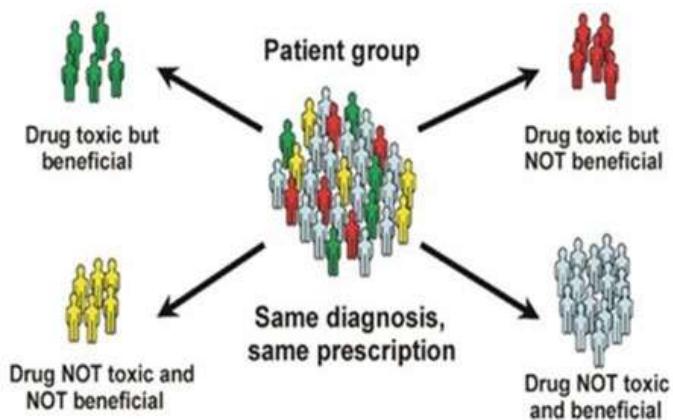
The genomic data requires proper structuring of the information. Earlier, the amount of data dealt by doctors and the hospital administer and payers was way beyond than what can be comprehended now. Presently due to availability of online information, it is easier to tackle the problems that we dealt for decades [28]. Precision medicine and data analytics extract useful information from the pool for improving the quality of health care. This is possible because of advancement and high throughput in –omic and EHR data. Though this seems perfectly right, there are a lot of challenges in –omic and EHR big data analytics. The first reason is the quality of data and the frequency of its occurrence. Second reason is the diversity lying in data types and issues related to dimensionality [28]. The medical health practitioners typically use EMR for clinical decision support. However, it is now important to incorporate –omics data into EMR to provide better support. The genomic variants that cause phenotypes which are based on EMR are identified and integrated with phenotype-to-genotype associations into an EMR system by a network consortium called eMERGE() [11, 20]. The reason for investing in healthcare information is that extensive use of EHR will majorly reduce the medical error, patient’s health complications, and mortality [3, 26]. This will also eventually decrease the healthcare cost. A working platform called 2bPrecise is used to map phenotypic information to genomic information. This platform helps in achieving more accurate diagnosis and treatments because it facilitates access to data sources (e.g., genomic labs) which were previously inaccessible. The mapping could be done either by checking the genomic sequence of a diabetic or cancer patient after learning the disease or when a mutated gene may not have expressed in any way but perhaps might have been associated with something we were unaware of. This will be a very vital part of discovery, from a research perspective.

3 Pharmacogenomics

In the current scenario, when a cluster of people report to a doctor for some disease or illness who undergo same diagnostic tests, it is observed that they react very differently to the same prescribed drug with same composition. The drug maybe beneficial to some while toxic to others. And for some, it may simply make no difference at all. See Fig. 2.

The reason could be the genetic makeup of an individual which varies from person to person. Pharmacogenomics is a part of precision medicine which is a blend of two fields of science, i.e., science of drugs (or pharmacology) and science of genes and its functions (genomics). It helps in developing effective and safe medications that

Fig. 2 Showing how a group of persons with the same diagnosis may react differently to the same treatment [14]



are customized in accordance to the variations present in the individual's gene. The mutations occurring in the coding region (exon) delete the entire amino acid sequence as a principle rule due to the introduction of stop codon in the protein sequence of the mutated gene. The problem now is predicting whether mutation in the sequence is going to damage the structure of the protein or if it is going to affect the stability of the final protein. Recently, we have known that the mutations in the intron region actually cause diseases. This signals that the tools used for analysis which uses only exons are insufficient and that we also require tools for analyzing the intron region as well. Unlike the native cognitive tasks such as speech recognition and visual processing, humans cannot naturally understand and analyze the genomic sequence, pathways, and its interactions. An extraordinary computational ability is required to analyze the entire genome which is another open challenge.

4 Machine Learning in Pharmacogenomics

Machine learning solves several important problems in genomic medicine. It is seen that genomics and precision medicine are two problem domains which serve as a good opportunity for AI scientists to add their expertise in this domain. The two major goals of precision medicine are to (i) learn by what means the DNA variations of an individual can lead to a disease and (ii) determine the reason for the occurrence of a phenotype so that the targeted therapies can be formulated. New techniques are being developed for interpreting and understanding the whole genome. Currently, a lot of resources are spent on new computational approach than data collection [19]. For example, the researchers working on The Cancer Genome Atlas were rethinking if they should shift the focus from sequencing to analysis after spending \$1 billion [17].

A computer system reads the genomic texts and uses it to support genomic medicine. With the advancement of gene editing, the scientists are working on altering the gene by either removing adverse mutations or by inserting new sequences in the

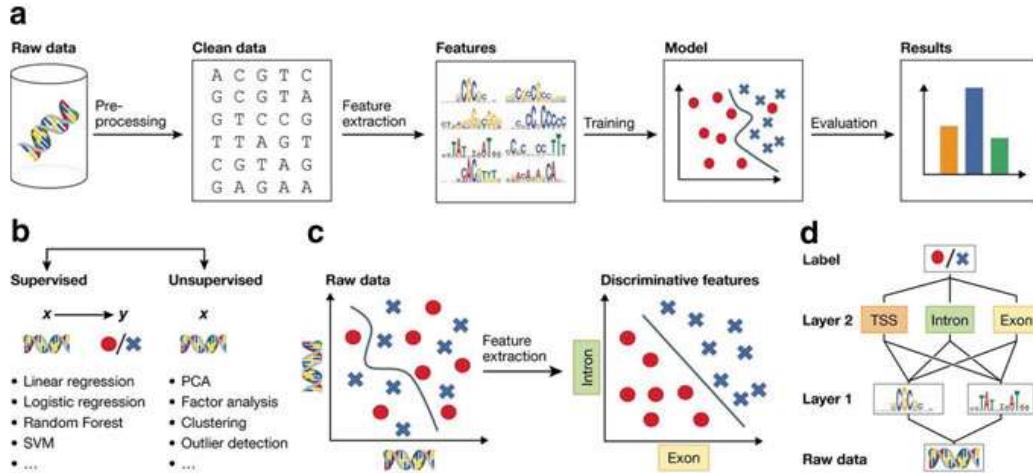


Fig. 3 Overview of different prediction approaches in pharmacogenomics [7]

appropriate predetermined genomic location with great efficacy. Genome editing technology makes it very important to predict the effects of these edits conducted [6].

The next question is---how do we interpret human genome using machine learning? Predicting the phenotypes or traits or disease risks from genome is typically a supervised machine learning problem. The input to the training model is a DNA sequence which is a genotype and the outcome is a phenotype (a character/behavior/trait). Figure 3a gives an overview of the entire genotype--phenotype process. Figure 3b-d shows different approaches for predicting the phenotype from the given DNA sequence. For more complex phenotypes and diseases, this approach is not very ideal. Firstly, because of the tremendous complexity lying between phenotype-to-genotype relationships, which include several tiers of biological processes and various regulatory mechanisms present within a single cell. The attempt is to infer the outcome of these control procedures by only observing the phenotype-to-genotype relations and by not considering the intermediary steps. Secondly, even if we understand and interpret the models that predict disease risk, there is a possibility that the hidden variables of the model may not actually correspond to the right biological process.

And so, it is believed that advanced machine learning, particularly deep learning will have a major play in computational biology. Comparative genomics uses two approaches that map genetic variants and disease risks. One is GWAS and the other is evolutionary conservation.

4.1 *Genome-Wide Association Study (GWAS)*

The purpose of GWAS is the detection of traits within a population and relating the variants to specific genomic location. They use microarray technology for understanding the association between human traits (diseases) and single-nucleotide polymorphism (SNP). However, modern GWAS analysis uses whole genome sequencing data, and not just a subset of variants. The major drawback of an association-based technique like GWAS is that they only point out the correlations between the variables and give no information about the reason behind it. The hidden variables caused by crossover or the differences present in the sub-population caused by migration may result in picking up an SNP which may be incorrect [27]. The GWAS often ignores the causal variant. Another prominent impediment in GWAS is stratifying the population. Computational models for predicting the disease can be created using the GWAS data which takes SNP profiles as input. Since the SNPs are high dimensional in nature, a large proportion of it is noisy and irrelevant to the disease in study. The causal variants can be prioritized by using PolyPhen [1], SIFT, and SPANR [32]. An exceptionally good review on current trends in GWAS can be read here [31].

4.2 *Evolutionary Conservation*

A powerful way to understand and interpret functional genomic sequences is by comparative genomics. This is achieved by comparing the genomes and conserving the distinct sequences that are “identical” across species (orthologous sequences) or by comparing similar sequences within a genome (paralogous sequences). The conservation-based techniques are used as an input for developing predictive models of disease. The mutation rates are very low in the conserved sequences despite the external forces. Also, conservation gives information only about deleterious [8, 18].

4.3 *Applications of Genomics in Diabetes*

GWAS has tested several millions of low-frequency variants in exons and introns region of genome. Nearly 50 genetic locations are identified with various glycemic traits [29]. The data from many genetic variants is combined to form a genetic risk score (GRS) for predicting the susceptibility of diseases (here, Type 2 diabetes). The GRS could predict only genetic risk factor and could not predict the non-genetic factors [21] and this holds good even for an upgraded GRS as well which comprised of 65 variants [30]. When there are effective and safe prevention measures which include behavioral interventions or drug therapies, risk prediction tools can be used. However, adults have greater risk of Type 2 diabetes; the Diabetes Prevention Program (DPP) can curtail the risk by nearly 50% [16]. A few researchers have found little evidence

for the fact that the genetic variant that is related to risk of diabetes has modified the effectiveness of the DPP program. A few other researchers contradicted the previous claim and said that communicating genetic information on disease risk might help to motivate healthy behaviors, but current facts do not prove this statement [13]. If the genetic tests could not predict diabetes or even prevent it, they can still majorly help in differentiating between their types (here Type I and Type II). Due to the outbreak of obesity [5], it is very tough to differentiate between the types of diabetes because some of the adults and children with Type 1 diabetes could also be flabby [10]. A major problem is encountered when the type of diabetes is misclassified. For example, a wrong diagnosis of Type 2 diabetes would result in incorrect treatment of diabetes by prescribing drugs that reduce the sugar levels. On the other hand, a wrong diagnosis of Type 1 will result in prescribing unnecessary insulin dosage. Recently, while studying about Type 1 and Type 2 diabetes, researchers [25] have found out 31 genetic locations for TYPE 2 diabetes by assessing the GRS which includes risky HLA genotypes. By including the clinical factors and autoimmune antibody tests, the GRS enhanced the distinction between Type 1 and Type 2 diabetes. It also helps in predicting who will need insulin treatment within 3 years of diagnosis. A major plus point of genotype-based diagnostic tool is that the results are persistent. However, before advising the clinical use of genetic testing, higher research has to be carried out for differentiating between both types of diabetes efficiently so that better treatment can be rendered for lowering glucose.

5 Conclusion

The world today expects a great deal of contribution of machine learning and deep learning techniques in the field of computational biology. Since the genomes cannot be interpreted by humans, it puts a pressure on machine learning researchers to develop models that can handle large amount of high dimensional datasets. Due to the existence of several layers of biophysical processes in single cell, powerful computational techniques need to be developed for better understanding and interpretation of genotype-to-phenotype relationship. Modeling the genome to disease relationship is all the more difficult and complicated as it is limited by the number of inputs and missing values for features.

Additionally, several environmental factors affect the bioprocesses in the cell and these result in variations in the genome of their offspring. Since these variations differ from person to person (holds good for look-alike and twins as well), only the genotype of an individual is not enough for determining the phenotype/disease on the whole. The genome of an individual along with their diagnostics test results and clinical observation together can be modeled for better results. The computational models however cannot fully replace the clinical examinations but they can considerably minimize the time and effort required for analysis and validation. Lack of thorough knowledge in field of biology for machine learning researchers poses problem in

preprocessing the data and feature extraction. The problem could be solved by joint collaboration biologists and machine learning experts.

The future work can be directed along 3 axes---(i) machine learning researchers can use genes sequence for analyzing the data, identify the patterns from large datasets. (ii) predicting the susceptibility of diseases from the DNA sequence. (iii) Another area of research is genome editing where the mutated genes can be altered by genomic medicine.

References

1. Adzhubei, I.A., et al.: A method and server for predicting damaging missense mutations. *Nat. Methods* **7**, 248 EP (2010)
2. Albert, F.W., Kruglyak, L.: The role of regulatory variation in complex traits and disease. *Nat. Rev. Genetics* **16**, 197 EP (2015)
3. Amarasingham, R., et al.: Clinical information technologies and inpatient outcomes: a multiple hospital study. *Arch. Intern. Med.* **169**(2), 108–114 (2009)
4. Bland, J.S., et al.: Translating emerging science into individualized wellness. *Adv. Med.* **2017**, 1–5 (2017)
5. Cheung, P.C., et al.: Childhood obesity incidence in the United States: a systematic review. *Childhood Obes. (Print)* **12**(1), 1–11 (2016)
6. Cong, L., et al.: Multiplex genome engineering using crispr/cas systems. *Science* **339**(6121), 819–823 (2013)
7. Cook, K.: Deep learning algorithms already hitting its limitations? (2019). <https://www.hou seofbots.com/news-detail/4463-1-deep-learning-algorithms-already-hitting-its-limitations>
8. Dermitzakis, E.T., et al.: Evolutionary discrimination of mammalian conserved non-genic sequences (cngs). *Science* **302**(5647), 1033–1035 (2003)
9. Evans, R.S.: Electronic health records: then, now, and in the future. *Yearb. Med. Inform.* **25**(Suppl 1), S48–S61 (2016)
10. Farmer, A., Fox, R.: Diagnosis, classification, and treatment of diabetes. *BMJ* **342** (2011)
11. Gottesman, O., Kuivaniemi, H., Tromp, G., et al.: The electronic medical records and genomics (eMERGE) network: past, present, and future. *Genet. Med.* **15**(10), 761–771 (2013)
12. Harrow, J., Frankish, A., Gonzalez, E.A.: GENCODE: the reference human genome annotation for the ENCODE project. *Genome Res.* **22**(9), 1760–1774 (2012)
13. Hollands, G.J., et al.: The impact of communicating genetic risks of disease on risk-reducing health behaviour: systematic review with meta-analysis. *BMJ (Clin. Res. Ed.)* **352**, i1102–i1102 (2016)
14. Huang, H., et al.: Predicting drug efficacy based on the integrated breast cancer pathway model. In: 2011 IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS), pp. 42–45, Dec 2011
15. Kaveeshwar, S.A., Cornwall, J.: The current state of diabetes mellitus in India. *Australas. Med. J.* **7**(1), 45–48 (2014)
16. William, C., Knowler, M.D., et al.: Reduction in the incidence of type 2 diabetes with lifestyle intervention or metformin. *N. Engl. J. Med.* **346**(6), 393–403 (2002)
17. Ledford, H.: End of cancer atlas prompts rethink: Geneticists debate whether focus should shift from sequencing genomes to analysing function. *Nature* **157**, 128–129 (2015)
18. Lindblad-Toh, K., Garber, M., Zuk, O., Lin, M.F., Parker, B.J., et al.: A high-resolution map of human evolutionary constraint using 29 mammals. *Nature* **478**(7370), 476–482 (2011)
19. Marx, V.: The big challenges of big data. *Nature* **498**, 255 EP (2013)
20. McCarty, C.A., et al.: The eMERGE Network: a consortium of biorepositories linked to electronic medical records data for conducting genomic studies. *BMC Med. Genomics* **4**(1), 13 (2011)

21. Meigs, J.B., et al.: Genotype score in addition to common risk factors for prediction of type 2 diabetes. *N. Engl. J. Med.* **359**(21), 2208–2219 (2008)
22. Mudasir, A.W., et al.: Big data: issues, challenges and techniques in business intelligence (2016)
23. Musunuru, K., et al.: Interdisciplinary models for research and clinical endeavors in genomic medicine: a scientific statement from the American Heart Association. *Circ. Genomic Precis. Med.* **11**(6) (2018)
24. Ohno-Machado, L., et al.: Genomics and electronic health record systems. *Hum. Mol. Genet.* **27**(R1), R48–R55 (2018)
25. Oram, R.A., et al.: A type 1 diabetes genetic risk score can aid discrimination between type 1 and type 2 diabetes in young adults. *Diab. Care* **39**(3), 337–344 (2016)
26. Parente, S.T., McCullough, J.S.: Perspective: health information technology and patient safety: evidence from panel data. *Health Aff.* **28**(2), 357–360 (2009)
27. Pritchard, J.K., Przeworski, M.: Linkage disequilibrium in humans: models and data. *Am. J. Hum. Genetics* **69**(1), 1–14 (2001)
28. Rajkomar, A., et al.: Scalable and accurate deep learning for electronic health records. *NPJ Digit. Med.* **1**, 1–10 (2018)
29. Scott, R.A., et al.: Large-scale association analyses identify new loci influencing glycemic traits and provide insight into the underlying biological pathways. *Nat. Genetics* **44**, 991 EP (2012)
30. Vassy, J.L., et al.: Polygenic type 2 diabetes prediction at the limit of common variant detection. *Diabetes* **63**(6), 2172–2182 (2014)
31. Visscher, P.M., et al.: Biology, function, and translation. *Am. J. Hum. Genetics* **101**(1), 5–22 (2017)
32. Xiong, H.Y., et al.: RNA splicing. The human splicing code reveals new insights into the genetic determinants of disease. *Science* **347**(6218), 1254806–1254806 (2015) (New York, N.Y.)

Personalised Structure Balance Theory-Based Movie Recommendation System



Aishwarya Sivakumar, Nidheesha Amedapu, Vasudha Avuthu, and M. Brindha

Abstract Recommending appropriate products has become the key to success of E-commerce systems. Collaborative filtering (CF) is the most widely used technology for recommender systems. However, it has several problems such as cold start problem that occurs due to insufficient data, lack of personalisation and scalability. The proposed method is a personalised structure balance theory-based hybrid method (personalised SBT), which attempts to solve the mentioned drawbacks of the traditional CF based system. The problem caused by the lack of data is handled by the application of rules of SBT. The system develops models of user preferences and product features, which are used for personalised recommendation. As it is a model-based system, it is highly scalable to handle Big Data. The process is implemented on Movie Lens-1M dataset, and the rating data of the dataset is used to develop the models that are used for recommendation. Mean Average Error (MAE) is used to evaluate the accuracy of the predicted rating, and Recall is used to evaluate the efficiency of the recommendation.

Keywords Recommendation system · Structure balance theory · User-based personalisation · Product features

A. Sivakumar · N. Amedapu · V. Avuthu · M. Brindha (✉)

National Institute of Technology Tiruchirappalli, Tiruchirappalli, Tamil Nadu, India
e-mail: brindham@nitt.edu

A. Sivakumar
e-mail: shiva.aishwarya97@gmail.com

N. Amedapu
e-mail: nidheeshardy@gmail.com

V. Avuthu
e-mail: vasudha97.rdy@gmail.com

1 Introduction

Recommender systems are used in various areas including recommendation of books, music, movies and products in general, for a target user. There are various recommender systems [1], and they typically are of two types—content-based filtering and collaborative filtering [2, 3]. In order to overcome the disadvantages of the two techniques, several hybrid approaches comprising of both techniques were proposed [4]. Cold start problem [5] caused by lack of rating data, scalability, time-based recommendation [6] and changing user preferences [7–9] are some of the major problems that have to be handled to create an efficient recommendation system. In this paper, the personalised SBT system effectively handles many of the problems faced by traditional recommendation systems. The problems caused by lack of rating data is solved by using the SBT rules, ‘enemy’s enemy is a friend’ and ‘enemy’s friend is an enemy’, for recommendation. The system uses the target user’s preference of each genre to find similar users, and the genre weightage of all the movies watched by the target user to find the similar movies, thus giving a more accurate recommendation.

2 Related Work

SBT based recommendation as implemented in paper [5] attempts to overcome the problem of data sparsity in collaborative filtering. Although, the paper combines both user-based and product-based recommendations to enhance the quality of prediction, it analyses neither the user preferences nor the movie characteristics to provide a personalised recommendation.

The paper [8] uses the user’s psychological profile to enhance recommendations. It analyses movie scores and user’s watch history for prediction enhancement but has cold start limitation and also does not make use of movie feature information.

The paper [9] uses a hybrid collaborative filtering algorithm based on the user preferences and item features. The recommendation results are better than the traditional collaborative filtering algorithm but it does not handle data sparsity well as it does not make use of the structural balance information in the user-product purchase network.

3 Proposed Model

The system comprises of four modules, namely the pre-processing module, user-based recommendation module, product-based recommendation module and the prediction module. Figure 1 shows the flow between modules. Movie Lens-1M dataset is used as input with 6040 users, 3952 movies and 1,000,209 ratings. The

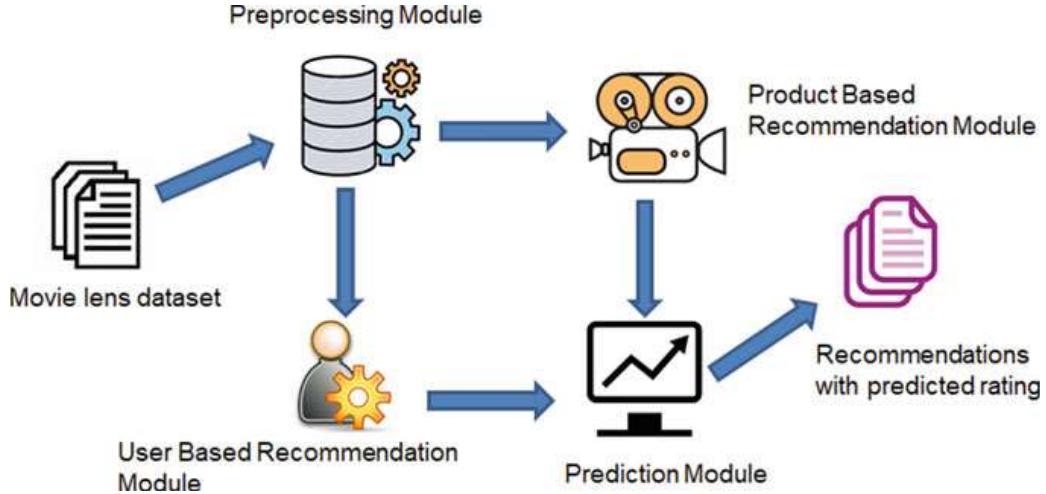


Fig. 1 Block diagram of the modules showing flow of input and output

dataset consists of movie information in the form MovieID::Title::Genres, user information in the form UserID::Gender::Age::Occupation::Zip-code and rating information in the form UserID::MovieID::Rating::Timestamp. The ratings are made on a five-star scale (only whole ratings), and each user has at least 20 different ratings.

3.1 Pre-processing Module

The input data is processed such that *USERS* consists of all the UserIDs, *MOVIES* consists of all the MovieIDs, *RATING* consists of UserID::MovieID::Genres::Rating and *GENRE* is the list of all genres. The *RATING* data is randomly split into two parts of 80 and 20%. 80% of the data is used for training and building models and 20% for testing. It forms the user preference and product feature models. The user preference model shows the preference of each genre for each user. For each movie, with genres as the features, the product feature model shows the contribution of each genre in the movie. These models are formed by using term frequency and inverse document frequency (TF-IDF). IDF of genre g (IDF_g) is found using Eq. (1).

$$IDF_g = \log \frac{\text{total no. of movies}}{\text{no. of movies with genre } g} \quad (1)$$

Term frequency of genre g and user u (TF_{ug}) is the weighted average of the occurrence of the genre g in the movies watched by user u with user rating for the movie m R_{um} taken as the weight as shown in (2).

$$\text{TF}_{ug} = \frac{\sum_m^{\text{movies_}u_g} R_{um}}{\sum_m^{\text{movies_}u} \sum_{k=1}^{|\text{genres}|} g_k R_{um}} \quad (2)$$

where value of g_k is 1 if the movie m watched by user u has genre g_k else 0, $\text{movies_}u$ is the set of movies that have been rated by user u , $\text{movies_}u_g$ is the set of movies having genre g and rated by user u and $|\text{genres}|$ is the total number of genres in the dataset. User genre preference for a user u and genre g , $P(u, g)$, is the product of TF_{ug} and IDF_g as in (3). The user genre preference model is a matrix P of size $|\text{users}| \times |\text{genres}|$. It is normalised for each user u as in (4).

$$P(u, g) = \text{TF}_{ug} * \text{IDF}_g \quad (3)$$

$$\sum_g^{\text{genres}} P(u, g) = 1 \quad (4)$$

Term frequency of genre g of movie m (TF_{mg}) as shown in (5) is the weighted average of ratings given by users who have watched the movie m , with weights as preference of the genre g by the users, i.e., $P(u, g)$.

$$\text{TF}_{mg} = \frac{\sum_u^{\text{users_}m} R_{um} * P(u, g)}{\sum_u^{\text{users_}m} \sum_g^{\text{genres_}m} R_{um} * P(u, g)} \quad (5)$$

where $\text{users_}m$ is the set of users who have rated movie m , R_{um} is the rating given by user u for movie m and $\text{genres_}m$ is the set of genres present in movie m . The contribution of genre g for a movie m , $Q(m, g)$ is a product of TF_{mg} and IDF_g as shown in (6). The product feature model is a matrix Q of size $|\text{movies}| \times |\text{genres}|$. It is normalised for each movie as shown in (7). The algorithm for this module is shown in Algorithm 1.

$$Q(m, g) = \text{TF}_{mg} * \text{IDF}_g \quad (6)$$

$$\sum_g^{\text{genres}} Q(m, g) = 1 \quad (7)$$

3.2 Product-Based Recommendation Module

The set of movies watched by the target user is compared with every other movie in the total set of movies. Only movies that are friends to more than half of the target user movies are considered as recommendation candidates. As similarities are found based on genre weightage in movies and every movie in target set is compared with every other movie, the SBT concept is not necessary here. Similarity returns a value from -1 to 1 . In this module, a movie with similarity value of >0.2 is considered a friend movie. The similarity between movies is found using (8).

$$\text{sim}(m1, m2) = \frac{\sum_g^{\text{genres_}m1m2} (Q(m1, g)) - \text{Avg}(Q(m1)) * (Q(m2, g)) - \text{Avg}(Q(m2))}{\sqrt{\sum_g^{\text{genres_}m1} (Q(m1, g)) - \text{Avg}(Q(m1))^2 * \sum_g^{\text{genres_}m2} (Q(m2, g)) - \text{Avg}(Q(m2))^2}} \quad (8)$$

Algorithm 1: Preprocessing module

```

for each  $g \in \text{GENRE}$  do
    Calculate( $\text{IDF}_g$ )
end for
for each  $u \in \text{USER}$  do
    for each  $g \in \text{GENRE}$  do
        Calculate ( $\text{TF}_{ug}$ )
         $P(u, g) = \text{TF}_{ug} * \text{IDF}_g$ 
    end for
end for
for each  $m \in \text{MOVIES}$  do
    for each  $g \in \text{GENRE}$  do
        Calculate( $\text{TF}_{mg}$ )
         $Q(m, g) = \text{TF}_{mg} * \text{IDF}_g$ 
    end for
end for
```

Here, $\text{genres_}m1m2$ is the set of genres found in both movies $m1$ and $m2$, $\text{genres_}m1$ is the set of genres found in movie $m1$, $\text{genres_}m2$ is the set of genres found in movie $m2$ and $\text{Avg}(Q(m))$ is the average value of all the q values for the different genres for that movie m . This module predicts rating for the movies it recommends using (9).

$$\text{prod_rat}(M) = \text{AvgRating}(M) + \frac{\sum_m^{\text{movies_U}} [\text{sim}(M, m) * (\text{orig_rating}[m] - \text{avgrat}[m])] }{\sum_m^{\text{movies_U}} [\text{sim}(M, m)]} \quad (9)$$

Here $\text{orig_rating}[m]$ is the rating on movie m by target user, $\text{avgrat}[m]$ is the average rating of movie m , $\text{sim}(M, m)$ is the similarity between movies m and M and movies_U is the set of movies watched by target user. The algorithm is as shown in Algorithm 2.

3.3 User-Based Recommendation Module

The target user is compared with every other user to find initial set of enemies. The initial set of enemies is compared with every other user to find possible friends of enemies. The union of the above two sets gives possible enemies. Now in the last step, every user in this set is compared with every other user to find enemies of enemies, which is possible friends. We use the similarity values < 0.2 for enemies and > 0.8 for friends, and similarity between users is found using (10). Movies watched by possible friend users with ratings above 3 are considered as recommendation candidates.

$$\text{sim}(u1, u2) = \frac{\sum_g^{\text{genres_u1u2}} (P(u1, g) - \text{Avg}(P(u1)) * (P(u2, g) - \text{Avg}(P(u2)))}{\sqrt{\sum_g^{\text{genres_u1}} (P(u1, g) - \text{Avg}(P(u1))^2 * \sum_g^{\text{genres_u2}} (P(u2, g) - \text{Avg}(P(u2))^2}} \quad (10)$$

Algorithm 2: Product based recommendation module

```

for each movie  $m_i \in \text{MOVIES}$  do
    set counter=0
    for each movie  $m_{\text{target}}$  watched by  $u_{\text{target}}$  do
        if  $\text{sim}(m_{\text{target}}, m_i) \geq 0.2$  then
            increment counter
        end if
    end for
    if counter  $> (0.5 * \text{no of movies watched by } u_{\text{target}})$  then
        do put movie  $m_i$  in  $\text{Product\_Rec\_Set}$ 
        calculate  $\text{prod\_rat}(m_i)$ 
    end if
end for

```

Here, genres_{u1u2} is the set of genres watched by both users $u1$ and $u2$, genres_{u1} is the set of genres watched by user $u1$, genres_{u2} is the set of genres watched by user $u2$ and Avg($P(u)$) is the average of all P values for the different genres for the user u . This module predicts a rating for the movies it recommends using (11).

$$\text{user_rat}(M) = \text{AvgRating}(u_t) + \frac{\sum_{u \in \text{users_}M} [\text{sim}(u, u_t) * (R_{uM} - \text{avgrat}[u])]}{\sum_{u \in \text{users_}M} \text{sim}(u, u_t)} \quad (11)$$

AvgRating[u] is the average rating of user u , users _{M} is the set of all users who have watched movie M , sim(u, u_t) is the similarity value of user u and target user u_t and R_{uM} is the rating given by user u on movie M . The algorithm for this module is shown in Algorithm 3.

3.4 Prediction Module

A union of the movies recommended in user-based (User_Rec_Set) and product-based (Product_Rec_Set) module is found. The rating for each of the movies in the union is predicted by combining the predicted rating from both the above modules. The product-based module is given a weightage of 0.7 and user based a weightage of 0.3 in (12). For movies present in just one of User_Rec_Set or Product_Rec_Set, the rating is taken as it is.

$$\text{PredictedRating}[M] = \alpha * \text{prod_rat}(M) + (1 - \alpha) * \text{user_rat}(M) \quad (12)$$

Evaluation metrics, Mean Average Error (MAE) and Recall percentage, are evaluated on the final set of movies and ratings using (13) and (14). The lower the MAE value, the better the accuracy of prediction. The higher the Recall value, the better the efficiency in recommendation. In (13), testing_movies is the set of movies used to test the system. Rated_movies is the set of movies rated by target user.

$$\text{MAE} = \frac{\sum_m^{\text{testing_movies}} \text{abs}(\text{original_rating}(m) - \text{pred_rating}(m))}{|\text{testing_movies}|} \quad (13)$$

$$\text{RECALL} = \frac{|\text{Recommended_movies} \cap \text{Rated_movies}|}{|\text{Rated_movies}|} \quad (14)$$

Algorithm 3: User based recommendation module

```

for each user  $u_i \in \text{USER}$  do
  if  $\text{sim}(u_{\text{target}}, u_i) \leq 0.2$  then
    do put user  $u_i$  in  $\text{Enemy1}(u_{\text{target}})$ 
  end if
end for
for each user  $u_i \in \text{Enemy1}(u_{\text{target}})$  do
  for each user  $u_j \in \text{USER}$  do
    if  $\text{sim}(u_i, u_j) \geq 0.8$  then
      do put user  $u_j$  in  $\text{Enemy2}(u_{\text{target}})$ 
    end if
  end for
end for
Enemy( $u_{\text{target}}$ ) =  $\text{Enemy1}(u_{\text{target}}) \cup \text{Enemy2}(u_{\text{target}})$ 
for each user  $u_i \in \text{Enemy}(u_{\text{target}})$  do
  for each user  $u_j \in \text{USER}$  do
    if  $\text{sim}(u_i, u_j) \leq 0.2$  then
      do put user  $u_i$  in  $\text{Possible\_Friends}(u_{\text{target}})$ 
    for each  $m_i \in \text{MOVIES}$  do
      if  $R(u_j, m_i) \in \{4*, 5*\}$  then
        put  $m_i$  in  $\text{User\_Rec\_Set}$ 
        do calculate  $\text{user\_rat}(m_i)$ 
      end if
    end for
  end if
  end for
end for
end for

```

4 Results

The experimental configuration of the system used is a Dell computer (2.8 GHz, 4.0 GB memory), Windows operating system and PYTHON 3 programming language. Each experimental case is executed ten times, and the average of execution results is considered. Three evaluation metrics are considered, and the results are discussed. Initial assumptions include that u is the available count of users in the dataset and m is the available count of movies in the dataset. u and m values are 1000 in all the three experiments.

In the first experiment, MAE of personalised SBT-Rec is calculated and compared with SBT-Rec [5], Web Services Recommendation (WSRec) [10, 11], Monte Carlo Complete Path Recommendation (MCCP) [10] and SBT-Service Recommendation (SBT-SR) [12, 13]. User-product rating matrix density changes from 1 to 4%. Values of MAE w.r.t. density(r) are plotted in Fig. 2. MAE is a measure of the error in the performance. The lower the MAE value, the better the algorithm.

As average rating is considered for missing rating data in WSRec, variation of MAE with r is not so obvious. As the method of finding friends in MCCP is not proper, MAE of MCCP is not less. MAE values of SBT-Rec are lesser than the MAE

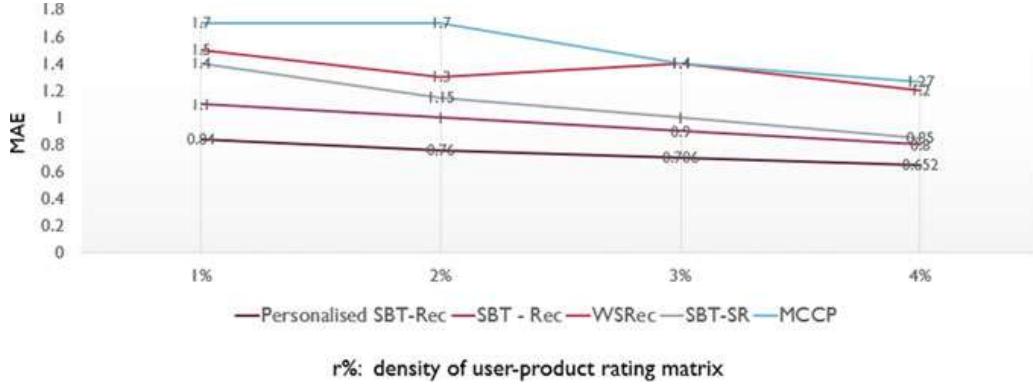


Fig. 2 MAE comparison of personalised SBT-Rec and other methods

values of SBT-SR because SBT-Rec uses both the rules of structural balance theory, whereas SBT-SR uses only one rule. However, personalised SBT-Rec outperforms SBT-Rec as genre preference of users and genre weightage of movies are included in the first one but not the latter. We can say that predicted ratings of personalised SBT-Rec are accurate. In the second experiment, Recall of personalised SBT-Rec is calculated and compared with SBT-Rec [5], WSRec [10, 11], MCCP [14] and SBT-SR [12, 13]. User-product rating matrix density changes from 1 to 4%. Values of Recall w.r.t. density(r) are plotted in Fig. 3. Recall is a measure of the efficiency of the algorithm. The higher the Recall value, the better the algorithm.

As average rating is considered for missing rating data in WSRec and MCCP, Recall is less. Recall of SBT-Rec is better than the Recall of SBT-SR because SBT-Rec uses both the rules of structural balance theory, whereas SBT-SR uses only one rule. However, personalised SBT-Rec outperforms SBT-Rec as genre preference of users and genre weightage of movies are included in the first one but not the latter. We can say that personalised SBT-Rec can make more precise recommendation. In the next experiment, execution times of four methods of recommendation are calculated and compared w.r.t. u and m , and the values obtained are plotted in Figs. 4 and 5, respectively. Execution time increases with increase in rating matrix

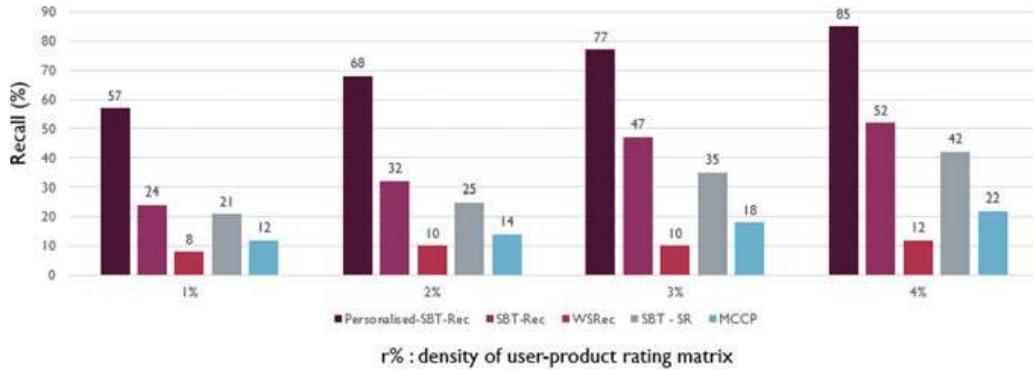


Fig. 3 Recall comparison of personalised SBT-Rec and other methods

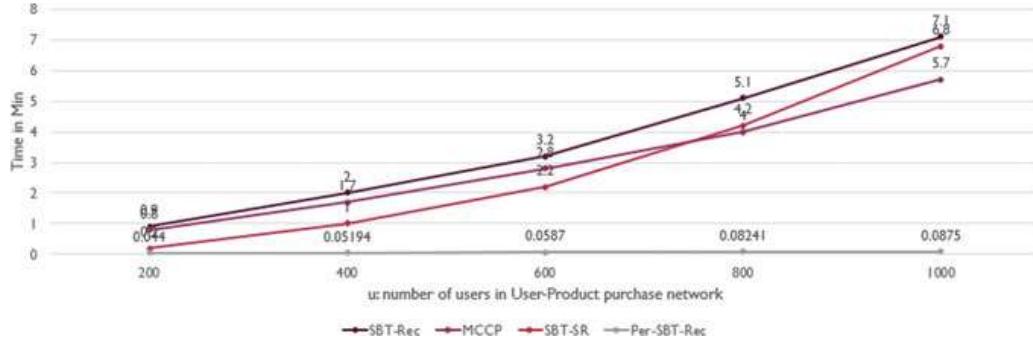


Fig. 4 Execution time versus available users (u)

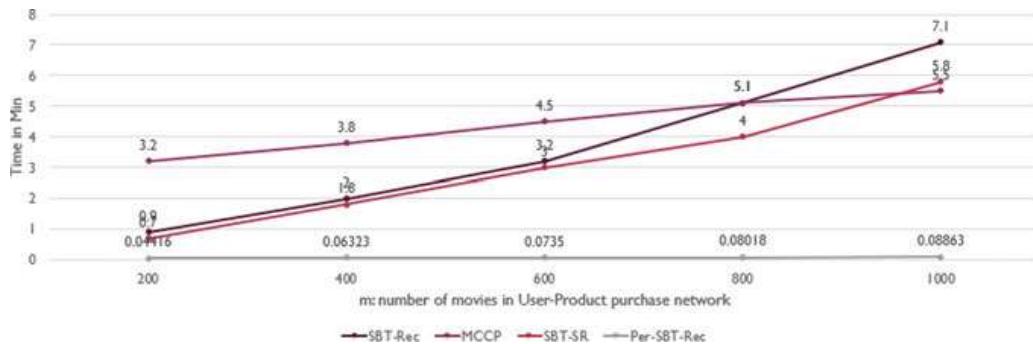


Fig. 5 Execution time versus available movies (m)

density. Execution time is high for SBT-Rec because SBT-Rec uses structural balance theory information. But, execution time of personalised SBT-Rec is less compared to all other methods because a model is developed from the pre-processing module containing P, Q matrices having user preference for each genre and genre weightage for each movie, and this is used for computation of similarity instead of entire dataset which reduces the execution time.

5 Conclusion

This proposed system attempts to overcome almost all the drawbacks of traditional collaborative filtering such as cold start problem, personalised recommendation and scalability. It uses the SBT rules to effectively handle the cold start problem. A model of user preferences and product features is developed by using the target users' preference for each genre, and the genre weightage of all the movies watched by the target user, for finding similar users and movies, thus giving correct set of possible friends and precise recommendation. As the system is model based, it is highly scalable to handle Big Data. Therefore, for a target user, predicted rating values and movies recommended are accurate. Although the paper solves a few

limitations of collaborative filtering, it can still improve in certain aspects. Even though model-based recommendation system increases computation efficiency, it is not open to changes in the trained data. The model has to be reconstructed to accommodate the changes. The systems accuracy is minimal for the user who has watched a lot of movies without any genre preferences. In the further research, automated similarity threshold setting method and time-aware recommendation can be analysed and incorporated.

Acknowledgements This research work was funded by the Department of Science and Technology, India, under the project ‘Design and Development of ICT-Enabled Cloud based mobile application for the self-promotion of products developed by Self Help Groups’.

References

1. Shah, K., Salunke, A., Dongare, S., Antala, K.: Recommender systems: An overview of different approaches to recommendations. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, pp. 1–4 (2017)
2. Rodrigues, C.M., Rathi, S., Patil, G.: An efficient system using item & user-based CF techniques to improve recommendation. In: 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, pp. 569–574 (2016)
3. Sharma, R., Gopalani, D., Meena, Y.: Collaborative filtering-based recommender system: approaches and research challenges. In: 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, pp. 1–6 (2017)
4. Virk, H.K., Singh, E.M.: Analysis and design of hybrid online movie recommender system. *Int. J. Innov. Eng. Technol. (IJIET)* **5**(2) (2015)
5. Qi, L., et al.: Structural balance theory-based e-commerce recommendation over big rating data. *IEEE Trans. Big Data* **4**(3), 301–312 (2018)
6. Maldhure, V.N., Deshmukh, V.M., Dandge, S.S.: Time based collaborative recommendation system by using data mining techniques. *Int. J. Recent Innov. Trends Comput. Commun.* **6**(5) (2018). ISSN: 2321-8169
7. Cai, Y., Leung, H., Li, Q., Min, H., Tang, J., Li, J.: Typicality-based collaborative filtering recommendation. *IEEE Trans. Knowl. Data Eng.* **26**(3), 766–779 (2014)
8. Chiru, C., Preda, C., Dinu, V., Macri, M.: Movie recommender system using the user’s psychological profile. In: 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, pp. 93–99 (2015)
9. Hu, L., Song, G., Xie, Z., Zhao, K.: Personalized recommendation algorithm based on preference features. *Tsinghua Sci Technol* **19**(3), 293–299 (2014)
10. Zheng, Z., Ma, H., Lyu, M.R., King, I.: WSRec: a collaborative filtering based web service recommender system. In: 2009 IEEE International Conference on Web Services, Los Angeles, CA, pp 437–444 (2009)
11. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware web service recommendation by collaborative filtering. *IEEE Trans. Serv. Comput.* **4**(2), 140–152 (2011)
12. Mayer, D., Butler, D.: Statistical validation. *Ecol. Model.* **68**(1), 21–32 (1993)
13. Qi, L., Zhang, X., Wen, Y., Zhou, Y.: A social balance theory based service recommendation approach. In: 9th Asia-Pacific Services Computing Conference, pp. 48–60, Dec 2015
14. Rong, Y., Wen, X., Cheng, H.: A Monte Carlo algorithm for cold start recommendation. In: 23rd International Conference World Wide Web, pp. 327–336, Apr 2014

Machine Learning Techniques for the Investigation of Phishing Websites



Ajaykumar K. B. and Bhawana Rudra

Abstract Phishing is ordinarily acquainted with increase a position in an organization or administrative systems as a zone of a greater assault, similar to an advanced tireless risk (APT) occasion. An association surrendering to such a partner degree assault generally continues serious money related misfortunes furthermore to declining piece of the pie, notoriety, and customer trust. Depending on scope, a phishing attempt may step up into a security episode from that a business can have an inconvenient time recuperating. So as to locate this kind of assault, we endeavored to make a machine learning model that advises the client that it is suspicious or genuine. Phishing sites contain various indications among their substance also, web program-based information. The motivation behind this investigation is to perform different AI-based order for 30 features incorporating Phishing Websites Data in the UC Irvine AI Repository database. For results appraisal, random forest (RF) was contrasted and elective machine learning ways like linear regression (LR), support vector machine (SVM), Naive Bayes (NB), gradient boosting classifier (GBM), artificial neural network (ANN) and recognized to have the most noteworthy exactness of 97.39.

Keywords Machine learning · Cyber security · Phishing

1 Introduction

Phishing is a sort of social building assault more often than not familiar with take client data, just as login accreditation's and ace card numbers. It happens once partner degree aggressor, taking on the appearance of a dependable element, tricks a injured

Ajaykumar K. B. (✉) · B. Rudra

Department of Information Technology, National Institute of Technology Karnataka Surathkal,
Mangaluru, Karnataka 575025, India
e-mail: ajaykumarkb92@gmail.com

B. Rudra

e-mail: bhawanarudra@nitk.edu.in

individual into hole partner degree email, text, or content message. The beneficiary is then fooled into clicking a malignant interface, which may bring about the establishment of malware, the stage change of the framework as a piece of a ransomware assault or the noteworthy of delicate information. An assault can have decimating results. For individuals, this incorporates unapproved buys, the taking of assets, or decide theft. Phishing assault regularly alludes to an expansive assault went for an enormous number of clients (or targets). This can be thought of as an amount over quality approach, requiring negligible planning by the aggressor, with the desire that in any event a couple of the objectives will fall unfortunate casualty to it (trying appealing even in spite of the fact that the normal addition for the aggressor isn't typically all that huge). These reason financial harm as referenced in 2007 [1]. Phishing assaults commonly connect with the client with a message proposed to request a particular reaction (normally a mouse click).

Assailants have improved on phishing assaults over a long time, thinking of varieties that require more straightforward exertion by the assailant yet result in either a higher pace of exploited people or a higher worth payout per injured individual (or both!). Maybe a couple of the phishing techniques are Spear Phishing [1, 2], Link Spoofing [1, 3, 4], Website Spoofing [5], Malicious and Covert Redirects [2, 4, 6].

As phishing attacks are increasing day by day that ends up in a security risk toward user information [1]. This was recorded and conferred within the type of statistics by the APWG and a few of the attacks are Kaspersky research lab that got exaggerated by 47.48 more than by all alternative attacks in 2016 [3]. Several techniques were instructed by the researchers to unravel the issue. Some used universal resource locator and compared it with existing blacklist that has a set of malicious websites et al. used these blacklist with the whitelist of legitimate website comparison. They used a heuristic approach that uses the signature info of the best-known attacks that match the heuristic pattern to decide whether or not the positioning is phishing website or not. For checking the phishing websites, researchers have thought-about measure the website traffic victimization Alexa [5] and some others use machine learning. Machine Learning is Associate in Nursing rising space will be wont to perform tasks that are capable of learning Associate in Nursingd act in an intelligent way. It uses supervised learning and unattended learning to coach a model with a given set of options. Supervised is employed to coach employing a set of measured options associated with the target label. Once trained it generates a brand new target label with the unknown knowledge set. Associate in Nursing unattended formula relies on generating a brand new dataset while not giving any target label for coaching the system [7]. During this paper, we have thought-about an oversized dataset and compared it with all the machine learning techniques and that we received high accuracy, sturdy and smart performance victimization random forest.

2 Literature Review

Phishing can be performed either by sending emails that lead to a fake site or by making the users access the links that are linked with a phishing site. The attackers target human vulnerabilities rather than by software mistakes. These scams are leading to the economic crises of the user [4]. In the early 90s, the users were provided a web portal and made it available online by the service provider. The company which allowed the fake transactions by the users is the American Online Company (AOL). Once the company was known about the phishing portal that is exploiting its services strengthened its system in the mid-90s for the prevention of phishing. Once the attackers were known about the strengthening of the system, they started a new technique by becoming the employee of AOL and started stealing valid accounts. The attackers have requested the user's password for the accounts for security reasons and they used email or message services to retrieve the required information [2]. The studies to solve phishing have categorized into blacklist, heuristic, content analysis, and machine learning techniques. The blacklist technique has become inefficient as the number of phishing websites is increasing day by day and leads to delay and this delay can lead to zero-day attacks [7]. The heuristic approach will use a signature database of the known attacks to form the heuristic pattern. This failed to detect novel attacks that bypass the signatures through obfuscation. Updation of the database of slow that can lead to zero-day attacks [5]. In the content-based approach, the authors used the well-known algorithms to detect phishing sites in terms of Term Frequency-Inverse Document Frequency (TF-IDF) and evaluated the CANTINA. This method analysis the text-based content to decide whether the site is a phishing site or not. They took some heuristics which can be applied for the reduction of false positives. The TF-IDF resulted to catch 97% of phishing websites and near about 6% false positives. With the heuristic approach, they could catch 90% of the phishing sites and near to 1% of false positives. The other researchers have implemented a desktop application for the detection of phishing websites with the help of heuristic- based URLs and web content [4]. They used PhisShield, along with the copyright, null footer links, and some other features and achieved 96.57% accuracy with a false positive of 0.035%. They used even the whitelist of the websites for the detection of phishing websites [4, 5]. URL approach was proposed for the detection of phishing sites by considering various components derived from the URL and the 3 metric for each and every component. Page Ranking is used to achieve the metrics and allows us to take a decision on whether the websites are phishing or not. It is found that 97% of the accuracy of the phishing websites using this method [5]. A prediction system.

The AI strategy learns the attributes of the phishing destinations and after that predicts the new phishing locales. A portion of the methods are Naive Bayes (NB), support vector machines (SVM), RF, artificial neural network (ANN), and Bayesian Net (BN). The precision will fluctuate starting with one calculation then onto the next. A portion of the work done by the analysts utilizing various strategies to recognize phishing sites are as per the following. The creators [8] incorporated some new highlights to the heuristic hunt highlights of CANTINA. To improve the proficiency

and lift the identification exactness by 15 and 20% regarding f-measure and error rate utilizing the six AI calculations. A redesign form to the CATINA will be CATINA + which included more highlights that have applied AI calculations and accomplished an exactness of 92% genuine positive and 0.4% false positive [9]. Mohammad et al proposed an enemy of phishing apparatus for the expectation of phishing assaults utilizing neural networks(NN) at a time scale [10]. To test the apparatus they considered 600 real URLs and 800 Phishing destinations to quantify the exhibition. The testing precision was 92.48% when utilized 17 highlights and 500 ages. Pradeepthi et al. thought about 4500 URLs and arranged them into four datasets. They tried on ten different AI calculations. They have inferred that the tree-based classifiers are increasingly reasonable for the phishing assault classification [11]. PhishStorm was proposed for the location of phishing URLs. They thought about 12 features and applied on the dataset of 96,018 phishing and real URLs utilizing administered order and brought about 94.91% exactness. The false positive is low of 1.44% that can ascertain the hazard score of the URLs on the testing dataset. The testing dataset exactness is of 92.22% for the authentic client and 83.97% for phishing URLs [12].

3 Materials and Method

Websites have various characteristics and patterns that can be considered as features. Here, we cover all phishing features that have been used in the earlier research as possible. Furthermore, we noticed some of the features are a subset of the other feature, and data-imbalance problem, both have been taken care of. The total features are 30. We divided them into four main types as shown with features in Table 1.

We initially used the available dataset [13] and build the model using various machine learning techniques like Naive Bayes, support vector machine, logistic regression, gradient boosting classifier, artificial neural network, random forest and achieved 97.39% accuracy using the random forest. As the existing dataset is having less URLs, We created our own dataset which contains 0.1 million legitimate URLs and 10,460 phishing URLs. Basically, we have written a python script which uses python third-party modules like urllib [14], beautifulsoap [15], whois [16], favicon

Table 1 Accuracy of the model using different algorithms

Classification algorithms	Training accuracy (%)	Test accuracy (%)
Naive Bayes	58.84	58.27
Support vector machine	89.35	82.52
Logistic regression	90.01	89.93
Gradient boosting classifier	79.96	76.34
Artificial neural network	93.45	92.64
Random forest	96.39	95.26

[17], socket [18], ssl [19], and takes each URL to extract features which are mentioned below. For example, consider a feature where based on the age we can say URL is legitimate or not. To find the age we need https certificate of a particular URL, here we use python module ssl [19] to get the certificate, then we find the difference between current date and the certificate mentioned date. Once we find the age we apply the rule which we mentioned in page 6 to find whether URL is legitimate or not. In our dataset 0 indicates suspicious, 1 indicates legitimate and -1 indicates phishing.

We have more number of legitimate URLs when we train our model, it will be biased. To overcome this we removed the imbalance in a dataset by using the resampling technique. There are various techniques of resampling, currently we used random undersampling and random oversampling [20]. Random undersampling is a technique which decreases the majority class randomly and uniformly, similarly random oversampling increases the minority class randomly and uniformly.

As we have 30 features in a URL, some of them are a subset of other, so removing such kind of features does not affect the overall accuracy and reduces the time complexity. The rules to be followed after extraction of the features are

- If Domain name as IP Address then it is treated as Phishing else it is a legitimate URL.
- URL Domain Length If URL length <54 features then is legitimate else if URL length is lesser than or equal to 54 and greater than or equal to 75 then the URL is treated as suspicious else phished URL.
- If the URL is tiny then it is phishing else legitimate user.
- If the number of hyphens is zero then legitimate else phishing.
- If the @Symbol exists in the Domain then it is a phishing URL else No.
- If // position exists in the last occurrence is less than 7 in the URL then Phishing Else NO.
- If the Domain contains Dots in part is equal to 1 then it is legitimate and the dots exist in part 2 then suspicious else it is phished URL.
- If HTTPS and the issuer is trusted along with the age of the certificate is more than 1 year then the URL is legitimate, if the HTTPS is used and the issuer is not trusted, suspicious else phishing.
- If the Domain expiry is lesser than or equal to 1 year then it is phished Domain else No.
- If Favicon is loaded from the external domain then it is a phishing favicon else no.
- The use of the port in the preferred status or the use of HTTP token the domain leads to phishing else no.
- If the percentage of URL request is lesser than 22% then the URL is legitimate and if it is greater than 61% leads to phishing and percentage present between these two is a suspicious URL.
- If the percentage of request of URL of Anchor is lesser than 31% is legitimate else phishing.

- If the percentage of <Meta >, <Script >, <Link >tags are less than 17 % then the site is legitimate and the percentage of tags are lesser than or equal to 82% is phishing and in between these are suspicious sites.
- If the Server Form Handler is blank or empty, phishing else legitimate. If the SFH refers to other domains then it is suspicious.
- If the hostname is not included in the URL, user information to be submitted by email or a status bar changes on Mouse Over or the right-click is disabled, a popup window contains text field, If no records exist of the domain, if IFrame redirection is used then it is a phishing URL else legitimate.
- If the website Rank is greater than 100,000 is suspicious and if it is less than 100,000 is legitimate, else it is phishing.
- If the page rank is lesser than 0.2 and webpage not indexed by google, link points to a webpage is zero is phishing else no.
- If the host belongs to the Top phishing IPs or its domains are phishing sites else No.

Based on these rules, the URLs are traced and compared with all the rules to find whether the URL is phishing URL or not and fetch accurate results.

3.1 Dataset

We have collected 0.1 million legitimate URLs from Alexa [13] and phishing websites consist of 10460 phishing URLs that have been collected from Phishtank [21]. We have also used the existing dataset which contains 11,000 URLs where 30 features extracted based on the features of websites in the UCI database [13].

3.2 Data Preparation and Preprocessing

Extracting features from the URL using python modules like selenium, urllib, etc. and creating our dataset. We have collected 0.1 million legitimate URLs and 10000 phishing URLs. As we have more legitimate URLs our model tends to learn more about legitimate URLs. To overcome this we balance the dataset, using random undersampling and random oversampling methods. We have gone through 30 features, and we found that some of the features are a subset of the other. For example, in Fig. 1 if we can see input feature 1.1.8, input feature 1.1.12 is a subset, that is, if we can find the existence of HTTPS, then we can also find whether it is in domain part or not. In order to reduce the computation by removing such features, we used genetic algorithm particle swarm optimization. After preprocessing the data, we will train our model with our custom dataset. Model is implanted using machine learning algorithms like Naive Bayes, Logistic Regression, Random forest, etc (Fig.2).

Input(Features)	Output(Class)
<p>1.1. Address Bar based Features</p> <ul style="list-style-type: none"> 1.1.1. Using the IP Address 1.1.2. Long URL to Hide the Suspicious Part 1.1.3. Using URL Shortening Services "TinyURL" 1.1.4. URL's having "@" symbol 1.1.5. Redirecting using "//" 1.1.6. Adding Prefix or Suffix Separated by (-) to the domain 1.1.7. Sub Domains and Multi Sub Domains 1.1.8. HTTPS 1.1.9. Domain Registration Length 1.1.10. Favicon 1.1.11. Using Non-Standard Port 1.1.12. The Existence of "HTTPS" Token in the Domain Part of URL <p>1.2. Abnormal Based Features</p> <ul style="list-style-type: none"> 1.2.1. Request URL 1.2.2. URL of Anchor 1.2.3. Links in <Meta>, <Script>, and <Link> tags 1.2.4. Submitting Information to Email 1.2.5. Abnormal URL <p>1.3. HTML and Java Script based Features</p> <ul style="list-style-type: none"> 1.3.1. Website Forwarding 1.3.2. Status Bar Customization 1.3.3. Disabling Right Click 1.3.4. Using Pop-up Window 1.3.5. IFrame Redirection <p>1.4 Domain based Features</p> <ul style="list-style-type: none"> 1.4.1. Age of Domain 1.4.2. DNS Record 1.4.3. Website Traffic 1.4.4. PageRank 1.4.5. Google Index 1.4.6. Number of Links Pointing to Page 1.4.7. Statistical-Reports Based Feature 	-1 Phishing 1 Legitimate

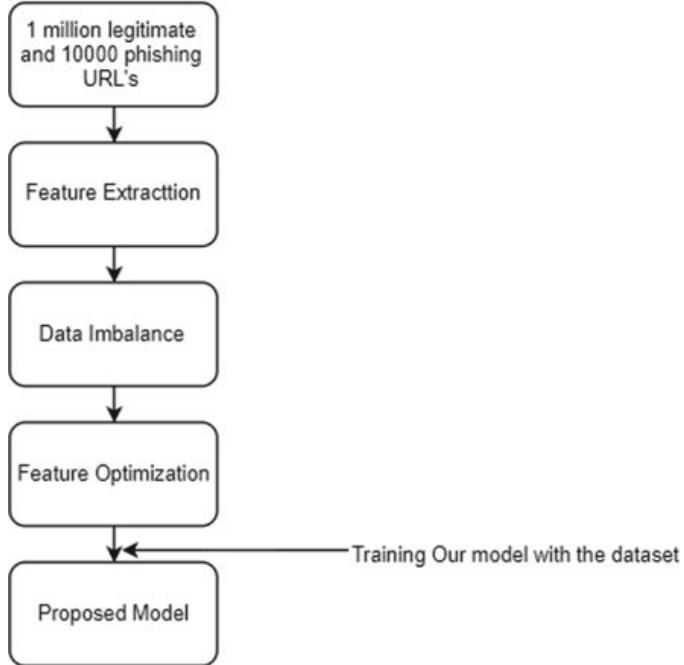
Fig. 1 Categories of features extraction

When we give URL as an input, we extract all the features like domain age, length of the URL, number of external images, etc. from the URL and convert that to the dataset having 30 features. Now this dataset (input data) is input for our model. Our model will predict whether the URL is legitimate or phishing. By using this information, we can stop phishing attacks. We developed the classifier using the RF technique as in the following steps:

- Divide data into training and test dataset, which we take 80, 20% for training and testing respectively.

Fig. 2 Different phases for training

Different Phases to train our model



- Train and test the combinations of 30 features dataset to get the features that arise the accuracy of detection.
- After second step, we have a number of features that go to the final stage of training and testing.
- Execute the final classifier.

4 Experimental Results and Discussion

In our approach, we have trained our model with different algorithms like the random forest, linear regression, and achieved accuracy accordingly. We have used our dataset to train the model, Table 1 is shown.

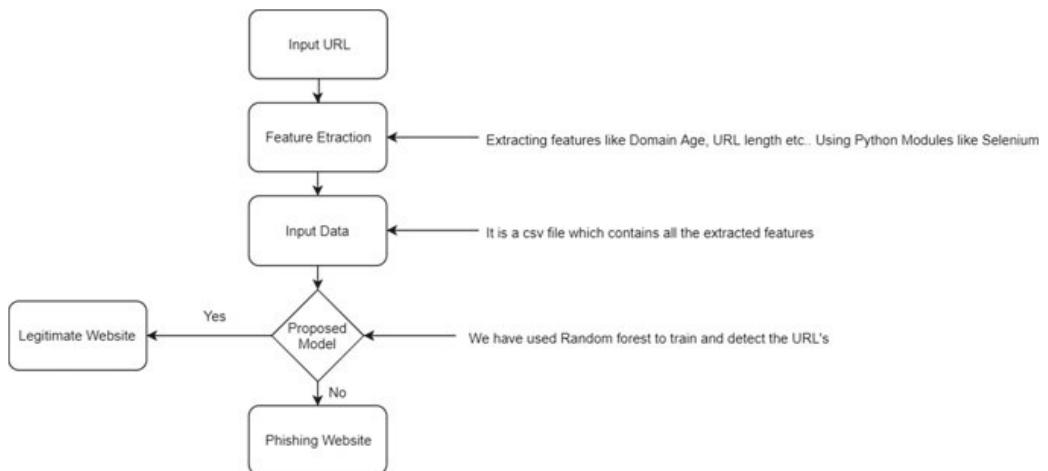
From Table 2 we can see that except random forest other classifiers have not produced good results because of their nature. Considering Support Vector Machine, it works well in higher dimension, but it does not work well with large dataset, similarly Naive Bayes classifier usually assumes that all dimensions are independent. Like Naive Bayes, Logistic Regression also works well if we can identify important independent variables. Coming to gradient boosting classifier, it has produced better results but random forest is more accurate because it uses ensemble learning technique where it creates as many trees as the subset of data and combine the output of all the trees and it also reduces variance.

Table 2 Accuracy of the model using different algorithms

Classification algorithms	Training accuracy (%)	Test accuracy (%)
Naive Bayes	60.40	60.48
Support vector machine	96.43	95.78
Logistic regression	92.81	92.66
Gradient boosting classifier	91.71	91.71
Artificial neural network	97.03	95.73
Random forest	98.99	97.39

5 Conclusion and Future Work

We have considered 30 features so as to decrease time and increase accuracy, with minimal combination of the features. Nonetheless, as a result of time deficiency and less resource availability, we chose features randomly to process it. We finished up after some perception that the combination of features is processed to take the state of a typical conveyance bend, it begins with least combination of features with low likelihood of time and combination devouring, at that point grabs in like manner, at that point goes down as it comes to the final number of 30 features, as appeared in Fig. 3. Subsequently, we chose them for our anti-phishing expansion program as final features that are utilized for the augmentation. The final accuracy which has been gotten is 97.39% by random forest for the standard dataset. For custom dataset at present, we have considered 0.1 million real URLs from Alexa which comprises of 1 million and accomplished 95.26% exactness. In the future, we are intending to deal with various combinations such has more phishing URLs and legitimate URLs.

**Fig. 3** Procedure to execute the system

References

1. AO Kaspersky Lab: The Dangers of Phishing: Help employees avoid the lure of cybercrime (2017). [Online] <https://go.kaspersky.com/Dangers-Phishing-Landing-Page-Soc.html>, 30 Oct 2017
2. Jakobsson, E., Myers, E.: Phishing and Counter-Measures: Understanding the Increasing Problem of Electronic Identity Theft. Wiley, 2006, p. 23
3. A Financial threats in 2016: Every Second Phishing Attack Aims to Steal Your Money Internet: <https://www.kaspersky.com/about/pressreleases/2017nancial-threats-in2016>. 22 Feb 2017 [30 Oct 2017]
4. Blasi, M.: Techniques for Detecting Zero Day Phishing Websites. M.A. thesis, Iowa State University, USA (2009)
5. Nguyen, L.A.T., To, B.L., Nguyen, H.K., Nguyen, M.H.: Detecting phishing web sites: a heuristic URL-based approach. In: 2013 International Conference on Advanced Technologies for Communications (ATC 2013), pp. 597–602 (2013)
6. Rao, R.S., Ali, S.T.: PhishShield: A desktop application to detect Phishing webpages through heuristic approach. Procedia Comput. Sci. **54**(Supplement C), 147–156 (2015)
7. VanderPlas, J.: Python Data Science Handbook, 1st edn, 1005 Gravenstein Highway North, Sebastopol, CA 95472.: O'Reilly Media, Inc., p. 331515 (2016)
8. Sanglerdsinlapachai, N., Rungsawang, A.: Web Phishing Detection Using Classifier Ensemble, pp. 210–215. NY, USA, New York (2010)
9. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: CANTINA+: a FeatureRich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. **14**(2), 21:1–21:28 (2011)
10. Mohammad, R.M., Thabtah, F., McCluskey, L.: Predicting phishing websites based on self-structuring neural network. Neural Comput. Appl. **25**(2), 443–458 (2014)
11. Pradeepthi, K.V., Kannan, A.: Performance study of classification techniques for phishing URL detection. In: Sixth International Conference on Advanced Computing (ICoAC), pp. 135–139 (2014)
12. Marchal, S., Franois, J., State, R., Engel, T.: PhishStorm: detecting phishing with streaming analytics. IEEE Trans. Netw. Serv. Manage. **11**(4), 458–471 (2014)
13. PhishTank Join the ght against phishing. [Online]. Available: <https://www.phishtank.com/>
14. <https://docs.python.org/3/library/urllib.html>
15. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
16. <https://pypi.org/project/python-whois/>
17. <https://pypi.org/project/favicon/>
18. <https://docs.python.org/3/library/socket.html>
19. <https://docs.python.org/3/library/ssl.html>
20. <https://imbalanced-learn.readthedocs.io/en/stable/oversampling.html>
21. Alswailem, A., Alabdullah, B., Alrumeayh, N., Alsedrani, A.: Detecting Phishing Websites Using Machine Learning. 978-1-7281-0108-8/19/\$31.00 (2019). IEEE

Feature Extraction and Classification of Gestures from Myo-Electric Data Using a Neural Network Classifier



Praahas Amin , Airani Mohammad Khan, Akshay Ram Bhat, and Gautham Rao

Abstract The information about intended hand gestures can be extracted by processing surface electromyography signals using non-invasive commercial off the shelf surface electromyography data acquisition devices. Surface electromyography signals have a great potential for use in multi-functional prosthetic controllers. The objective of this study is the implementation of a classifier that can be used to classify gestures from Myo-electric data obtained from the Myo-armband. This study describes in detail a method for data acquisition, feature extraction, and offline gesture classification using Artificial Neural Network. The performance is then compared with a Support Vector Machine Classifier. The proposed approach results in an accuracy greater than 94% for validation data set for classification of five distinct hand gestures. It could be concluded that this technique could be used in the human-machine interfaces with five distinct control signals including rest. A significant observation in this study was that a single artificial neural network taking inputs from all sensors simultaneously gives inferences with better accuracy compared to a system with a separate neural network for each sensor with a majority voting to decide the classification of the gesture.

Keywords Human machine interface · Prostheses · Artificial neural networks · Pattern recognition · Machine learning · Feature extraction · Electromyography

P. Amin  · A. M. Khan

Department of Electronics, Mangalore University, Mangaluru, Karnataka 574199, India

e-mail: praahas1234@gmail.com

A. M. Khan

e-mail: asifabc@gmail.com

A. R. Bhat · G. Rao

Flashflow Technologies (OPC) Private Limited, Mangaluru, Karnataka 575003, India

e-mail: akshay.bhat981222@gmail.com

G. Rao

e-mail: graogautham@gmail.com

1 Introduction

Robotic assistive devices have been in use for some time now for the rehabilitation of amputees and people affected by paralysis. These assistive devices have a human-machine interface using which the user can seamlessly interact with the device. The technique used for this purpose is surface electromyography (SEMG). Myo-electric control systems are used in prosthetic systems to rehabilitate amputees as well as in assistive exoskeleton suits. When a human being intends to move a limb, the intention originates in the brain results in the contraction of muscle fibers which cause a movement in the limbs. The signals that cause the contraction, known as electromyography (EMG) signals, can be detected by non-invasive techniques from the skin surface using commercial off the shelf devices. Human-machine interfaces (HMI) can be developed around such Myo-electric control systems. Rapid growth of miniaturized and efficient electronic systems has opened up interesting application scenarios in the field of HMI such as powered exoskeleton suits and prostheses that could be used for rehabilitation of amputees or stroke survivors to regain previous dexterity and aid in activities of daily living (ADL). A simple contraction-detection system can be used for prosthetic systems that can perform only two gestures, extension and contraction of the fingers. By using machine learning techniques for pattern recognition, it is possible to find distinct pattern in the SEMG signal corresponding to different gestures. However, the SEMG signal is a time-varying signal, and on observation, it can be seen that the signal amplitudes cannot be used as a discriminating feature for classifying gestures. There are features called Time-Domain Features, Frequency-Domain Features, Time-Frequency Features [1, 2], and Wavelet-Based Features [3] which can be used to identify patterns in the signal. In this paper, the focus is on Time-Domain Features. Using machine learning strategies to learn from these features, a Myo-electric control system for multi-function prostheses can be designed. In [4], a real-time control scheme for a Myo-electric control system using pattern recognition was proposed. Different intended gestures will have distinct patterns in the signal. Therefore, pattern recognition is a critical part of the Myo-electric control systems. This paper proposes a complete end-to-end methodology for data acquisition, feature extraction, and classification of gestures using artificial neural network (ANN) for a single user Myo-electric system using a commercial off the shelf device for data acquisition.

2 Background

The use of pattern recognition in Myo-electric systems was first introduced in one of the earliest works for multi-function prosthesis is presented in [5]. The authors developed a Myo-electric control system for upper limb prosthesis that could classify four movements with an accuracy of 90% and an average error or 9.25%. The

features considered were Mean Absolute Value (MAV), MAV Slope, Zero Crossings (ZC), Slope Sign Change (SSC), and Waveform Length (WL). These are the Time-Domain features which are referred to as Hudgin's Time-Domain Features. Implementation of Myo-electric systems for use with real-time systems is proposed in [4] where the authors specify that Hudgin's Time-Domain Features outperforms Wavelet Features. Their study showed that for Time-Domain features the average signal processing and classification times were approximately 16 ms for a 256 ms data set. This is important from a control perspective because it means that relatively complex feature extraction can be performed without introducing substantial delays into the controller. Again, this work was focused on the real-time implementation of the classifier and the advantages of majority voting. The effects of electrode displacement on classification accuracy are presented in [6]. The authors used Hudgin's Time-Domain Features, Auto-Regression, and a combination of the two feature sets with a Linear Discriminant Analysis (LDA) Classifier. The benefits of pattern-recognition-based Myo-electric systems in comparison with conventional Myo-electric systems is discussed in [7], where the authors concluded that the benefits of intramuscular electrodes maybe useful for conventional Myo-electric controllers, where signals from each electrode could be used to independently control actuators for multi-function, unlike pattern-recognition-based multi-function Myo-electric systems which functions with acceptable accuracy and is more accepted by users due to its non-invasive nature in contrast with the conventional systems.

The usage of a Support Vector Machine (SVM)-based classification scheme for Myo-electric control applied to upper limb is investigated in [8]. Their work presented a method to adjust SVM parameters before classification, and studies overlapped segmentation and majority voting as two schemes to improve controller performance. An SVM, as the core of classification in Myo-electric control, is compared with two commonly used classifiers: LDA and multi-layer perceptron (MLP) neural networks. It demonstrates exceptional accuracy, robust performance, and low computational load.

A system that uses SVM to perform hand gesture recognition based on SEMG signals is proposed in [9] where the study demonstrates the application of EMG signals for controlling home devices. To implement this, a commercial off the shelf Myo-armband that has an array of eight SEMG sensors around the forearm was used. Two components were presented in it. First, an analysis of the SEMG data delivered by the Myo-armband, involving gesture recognition using the SVM Classifier and second, implementation of a gesture capture and recognition system using Myo-armband and SVM, achieving gesture control for home devices using IR communication protocol. 15 different hand gestures were studied to create a dictionary of gesture control. These gestures represent multiple hand and finger movements. Gesture recognition was achieved using SVM classification. Radial, polynomial, and sigmoid kernels were tested to achieve the best conditions for learning and recognition of hand gestures as well as an accurate determination of these movements. To demonstrate the applicability of the results, a gesture control system was implemented as an embedded system by the authors. The system also enables Bluetooth

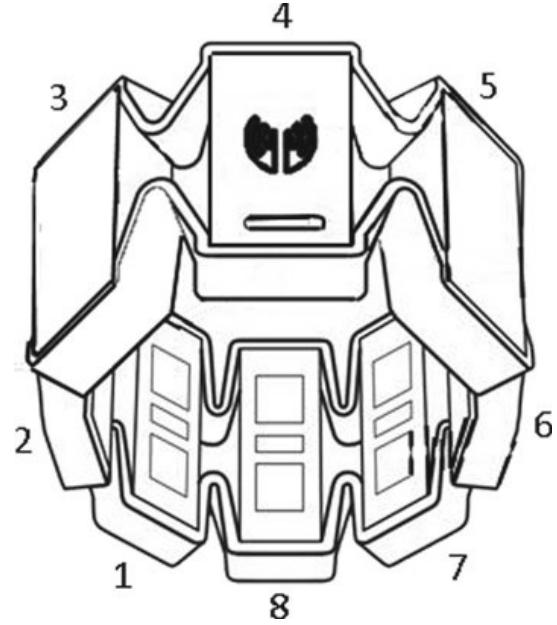
communication with the armband to send gesture control commands to household devices using IR protocol.

The performance of the Myo-armband was tested in [10], and the results showed that the quality of SEMG signal obtained by Myo-armband in terms of frequency spectrum properties was close to the ideal values. They concluded that the Myo-armband is more robust for class-separability than the conventional EMG system. Moreover, the Myo-armband is more comfortable to use and more readily accepted by the users than the conventional EMG system. The Myo-armband performance was evaluated with the LDA Classifier in [11]. The effects of electrode displacement and the solution for this problem are discussed in [6, 12, 13] where the authors suggest to include data points corresponding to gestures with displaced electrodes as well, which would compensate for the effects of electrode displacement. In [14], an investigation was done on the fluctuation of EMG signals and its effect on pattern recognition algorithms over a period of time, indicating that the Myo-electric systems may have to be retrained after a period of usage. In [15], the authors presented an EMG based finger movement recognition for prosthetic arm. Two-channel EMG sensor was used, and a comparative study was done on the performances of K-Nearest Neighbors (KNN) algorithm and ANN. One of the earliest studies on using neural networks to perform classification on Myo-electric signal is [16] where the authors used a two-layer perceptron to classify elbow extension and flexion and wrist pronation and supination. In [17], the focus is on classification of pinching with one of three fingers. Baseline wander can occur due to electrode displacement, and the technique for baseline wander correction is discussed in [18]. In [19], the authors discuss a solution for additive white Gaussian noise whose spectral components coincide with the spectral components of EMG signals, making it a problem for EMG signal analysis. In [20], the authors describe a technique of feature fusion using canonical correlation analysis, so that the number of features is minimized. The reviewed literature helped in designing the methodology for data acquisition.

3 Methodology

In this section we outline the methodology used for acquisition of data, training, evaluating, and testing the ANN. This process can be applied to any arm gesture. In this paper, focus is on five gestures for a single user personalized human-machine interface control system. They are Closed Fist (CF), Index Finger Extension (IE), Cylindrical Grip (CG), Middle Finger Extension (ME), and Rest (RE). Myo-armband(TM) by Thalmic Labs was used for acquiring the SEMG data. The Myo-armband contains eight surface electromyography sensors. The numbering of the sensors is denoted in Fig. 1, and it can be seen that the sensors are equally spaced. Data was acquired from ten consenting healthy individuals. The Myo-armband samples data at the rate of 200 Hz implying that every 5 ms, a reading from each of the eight sensors is received, which is stored in a CSV file.

Fig. 1 Myo-armband used for data acquisition



The gestures to be classified were selected in such a way that the muscle groups used to generate these signals were very similar, leaving it to be quite a challenge to classify. The muscles mainly used for flexion of the muscle were flexor digitorum superficialis and the flexor digitorum profundus. Flexor digitorum superficialis is the superficial muscle that helps in the movement of the fingers, but the main muscle in use on the anterior side is the flexor digitorum profundus. On the posterior side, we have the extensors, most important of which is the extensor digitorum profundus, a deep-seated muscle [21].

The SEMG signals picked up by the Myo-armband are primarily from the superficial muscles, with the deep-seated muscle contraction for particular gestures causing certain signal changes leading to slightly different patterns. In this paper, we aim to make these classes of signals as distinctly distinguishable from one another as possible.

The experimental setup comprises of a Myo-armband which is connected to a computer using a Bluetooth dongle. The test subject wears the Myo-armband on their forearm. To collect the data, a custom Python application was developed which takes the subject's name, gesture type, and batch number as input and then prompts the subject as to when to flex or extend the gesture and relax. The subject then follows the on-screen prompts. Data is acquired for the gestures shown in Fig. 2. The acquired raw data will be stored in a .CSV file which is then used for feature extraction in the application. The representation of the data in the feature space is also available as a .CSV file in which each row represents a point in the feature space. These features are then used for training and validating the Artificial Neural Network. The experimental procedure is discussed in the following sections.

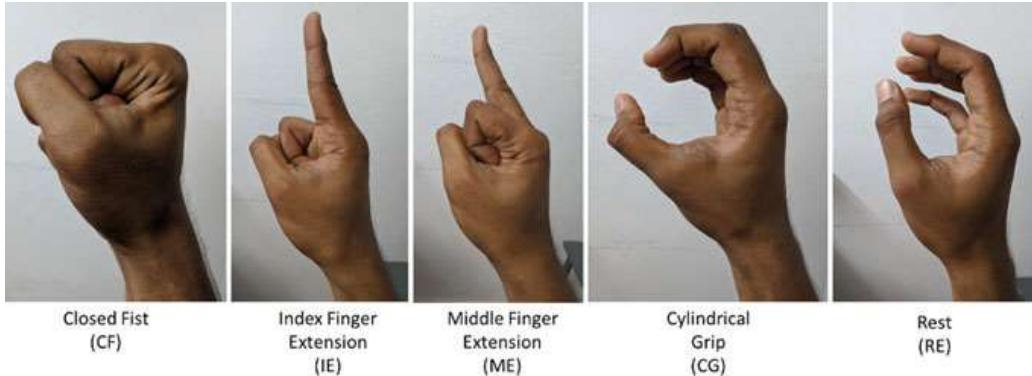


Fig. 2 Five gestures used in the study

3.1 Data Acquisition

The data set used for this study is a new data set acquired using the methodology described below. The data set consists of data from ten users. The Python application is used for data acquisition. Each subject wears the Myo-armband and performs five different gestures including rest. 100 instances were collected for each subject for each of the five gestures. One instance would include performing the gesture within a window of 2 s followed by a rest window of 2 s. This pattern of flexion or extension followed by relaxation is repeated in order to acquire the SEMG data. Taking into consideration the fatigue that accumulates over time, each session is restricted to a batch of 25 gesture instances per session. The window size considered is of 180 samples over which features are extracted. The feature set considered are ten Time-Domain Features [22], namely Integrated EMG (IEMG), Mean Absolute Value (MAV), Simple Square Integral (SSI), Variance (VAR), Root Mean Square (RMS), Waveform Length (WL), Log Detector (LOG), Willison Amplitude (WAmp), Slope Sign Change (SSC), and the number of Zero Crossings (ZC) within the window.

3.2 Data Segmentation

Myo-electric systems are meant to be used with real-time systems, and therefore, the data is segmented in SEMG analysis. Segment or window is the number of samples being considered for analysis and feature extraction. Data segmentation can be done using two techniques, i.e., disjoint window technique and overlapping window technique. In this paper, the technique used for the data collection and feature computation is the disjoint window technique. A threshold value is set to identify the onset of gestures to ensure that a processing segment is not erroneously initiated. When the signal amplitude crosses the threshold when a gesture is made, the segment for computation of the features for the window size is initiated. This happens across

all the eight channels. The threshold values were chosen after studying the SEMG signals of each user as the threshold levels for each user may not be the same.

3.3 Feature Extraction and Classification

The chosen features must be capable of distinguishing the characteristics of the five chosen gestures. The raw SEMG signal is processed to identify relevant patterns in the data. The features considered are described below.

Integrated EMG (IEMG) is calculated as the summation of the signal's absolute value and is often used as an active segment detector.

$$\text{IEMG} = \sum_{n=1}^N |x_n| \quad (1)$$

where x_n is an EMG signal and N is the length of that signal.

Mean Absolute Value (MAV) is used to mark the onset of a gesture.

$$\text{MAV} = \frac{1}{N} \sum_{n=1}^N |x_n| \quad (2)$$

Simple Square Integral (SSI) is a representation of the energy in EMG signal.

$$\text{SSI} = \sum_{n=1}^N |x_n|^2 \quad (3)$$

Variance of EMG (VAR) is an estimate of the power content of the signal.

$$\text{VAR} = \frac{1}{N-1} \sum_{n=1}^N (x_n - \bar{x}) \quad (4)$$

where \bar{x} is the signal mean.

Root Mean Square (RMS) is an estimate of the standard deviation of the signal and also estimates the power content of the signal.

$$\text{RMS} = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (x_n - \bar{x})^2} \quad (5)$$

Waveform Length (WL) is the cumulative length of the signal that contains information about signal frequency, amplitude, and duration.

$$\text{WL} = \sum_{n=1}^N |x_{n+1} - x_n| \quad (6)$$

Willison Amplitude (WAMP) measures the motor unit activity while ignoring noise as defined by the threshold level.

$$\text{WAmp} = \sum_{n=1}^N f(|x_{n+1} - x_n|), \quad \text{where } f = \begin{cases} 1, & \text{if } x_{n+1} - x_n \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Log Detector (LOG) provides an estimate of the force exerted by a muscle.

$$\text{LOG} = e^{\frac{1}{N} \sum_{n=1}^N \log|x_n|} \quad (8)$$

Slope Sign Change (SSC) estimates the frequency content of a signal and ignores noise as defined by threshold value.

$$\text{SSC} = \sum_{n=2}^N f((x_n - x_{n-1})(x_n - x_{n+1})), \quad f = \begin{cases} 1, & x \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Zero Crossing (ZC) estimates the frequency content of the signal and ignores noise as defined by threshold value

$$\text{ZC} = \sum_{n=1}^N \text{sgn}(-x_n \times x_{n+1}), \quad \text{sgn}(x) = \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

The extracted features for every window are then stored in a .CSV file. This file holds the data represented in the feature space. This data can now be used to train a classifier and generate the classifier model. The data set is split into training and validation sets in the ratio of 70% and 30%, respectively. The experiment had two approaches. In the first approach, the data from the eight sensors were used to train eight different Artificial Neural Networks. A majority voting was performed on the output of the eight neural networks. However, the accuracy of the classifier failed to improve beyond 89%. This was compared with an SVM Classifier which gave an accuracy of 87%. The second experiment made use of a single Artificial Neural Network which took as input the ten features of each of the eight sensors. For this study, we opted to avoid performing any technique of dimensionality reduction and continued to work with ten features for every sensor. We scaled the data to be in the range $(-1, 1)$ with a mean 0 and unit variance. Thus, the input vector consisted of 80 features with 10 features from each sensor. These 80 features made up one data point. After we presented the data in this format, we use it to train a neural network model. The performance of the ANN Classifier is compared with an SVM Classifier

[23]. The training work flow of the model is best represented by Fig. 3 and the testing work flow by Fig. 4. The results are discussed in the upcoming sections.

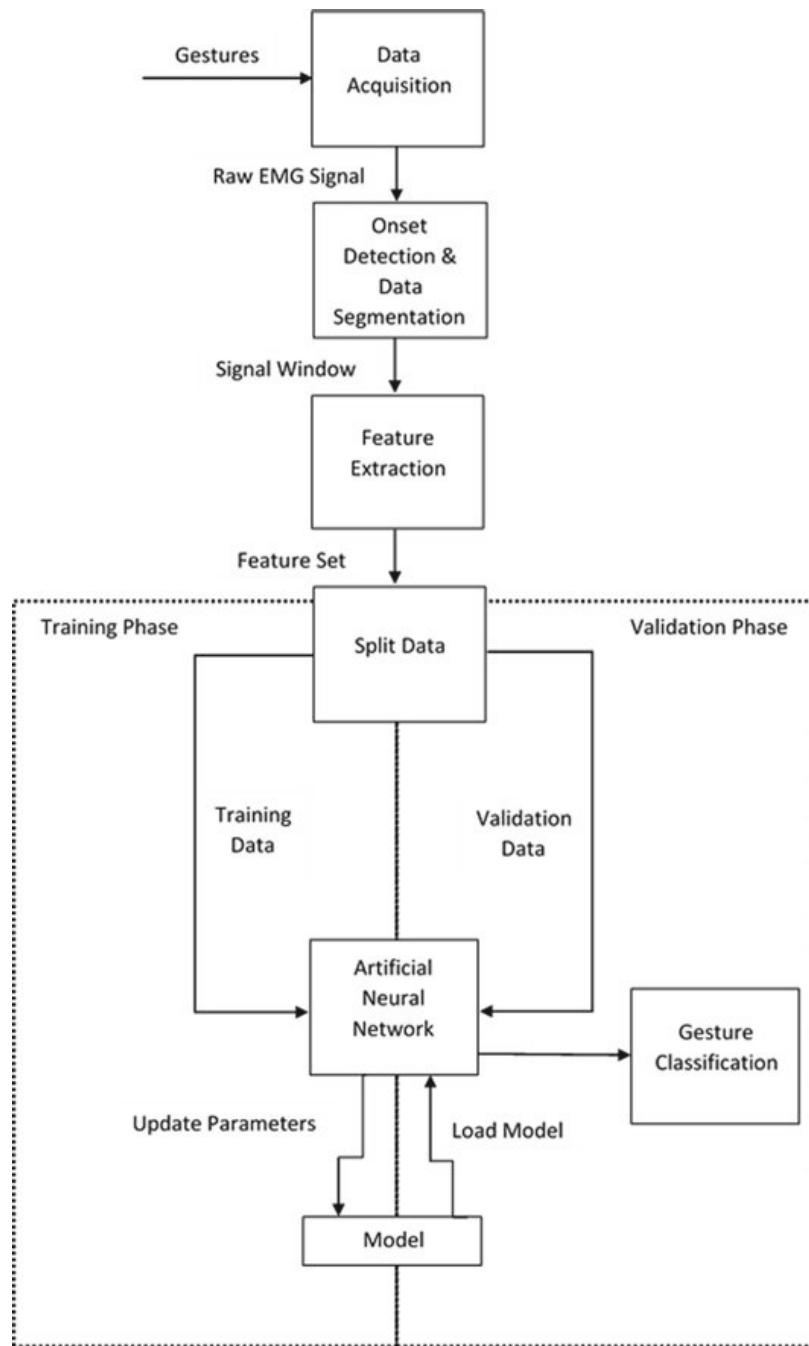
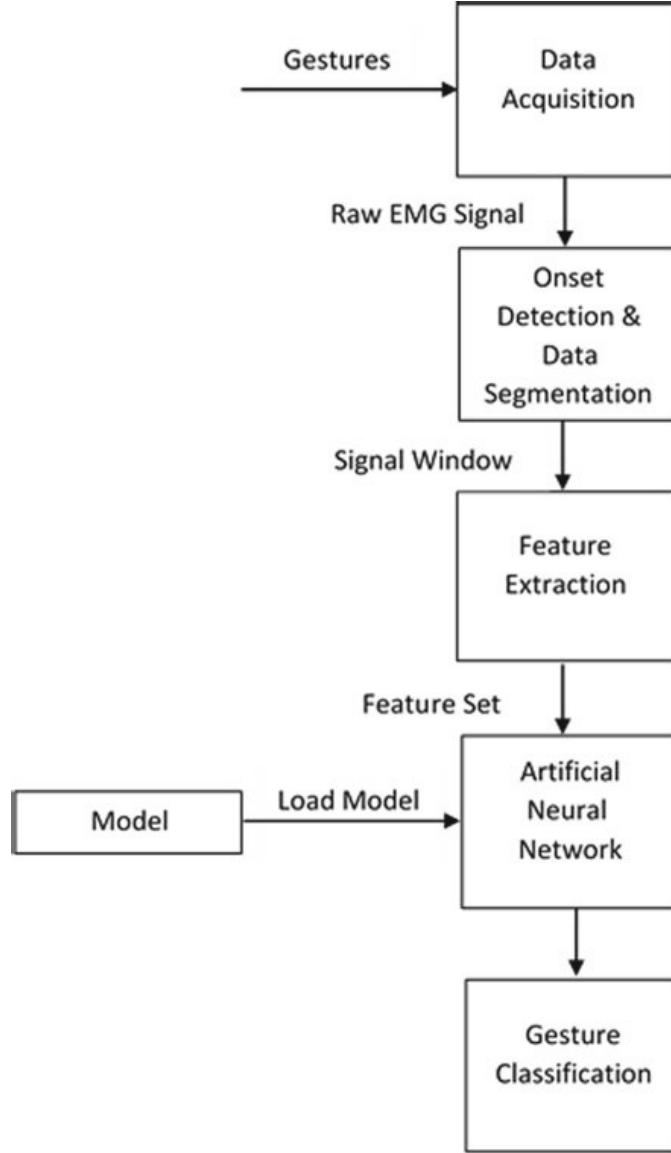


Fig. 3 Training and validation work flow

Fig. 4 Testing work flow

4 Discussion and Results

The experiments and trials showed promising results for an offline classification system. A good score for accuracy was achieved by using the second approach as mentioned in Sect. 3. This can be attributed to the idea that neural networks require a larger pool of features as well as data in order to predict robustly.

The first approach had a smaller subset of features with only ten features for every neural network, the networks refused to converge and showed a weaker validation score with the same number of data points. The second method was comparatively more effective as the amount of knowledge being learned by the Artificial Neural Network was higher.

The case of User 5 from the set of ten users has been selected at random, and the plot for the training and validation accuracy for User 5 is shown in Fig. 5, and the training and validation loss is shown in Fig. 6.

The performance metrics for the Artificial Neural Network Classifier and SVM can be seen in Tables 1 and 3, respectively. The performance metrics considered are Accuracy (ACC), Precision (PRE), Recall (REC), and F1-Score (F1). The calculation of the performance metrics are as given and can be calculated from the confusion matrix.

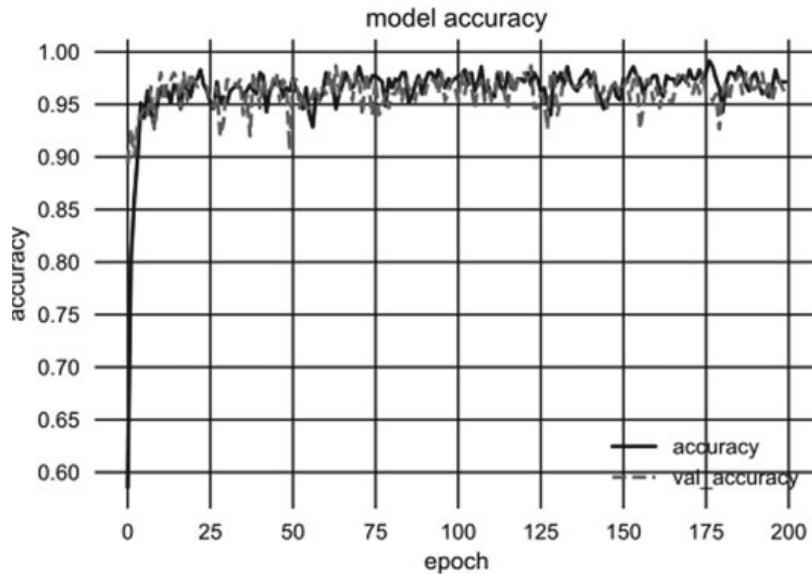


Fig. 5 Model accuracy during training for artificial neural network

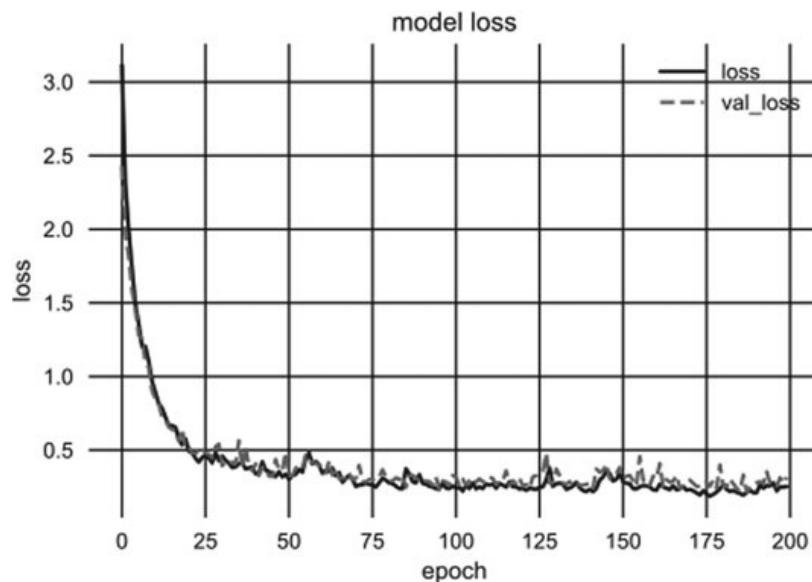


Fig. 6 Model loss during training for artificial neural network

Table 1 Gesture accuracy of neural network classifier for ten users

User	Gesture accuracy (%)				
	CF	IE	ME	CG	RE
1	96.6	96.6	100	93.3	100
2	100	93.3	100	90	100
3	90	100	90	100	100
4	100	100	86.6	100	100
5	93.3	93.3	93.3	100	100
6	100	100	96.6	96.6	100
7	93.3	100	100	96.6	100
8	93.3	90	96.6	93.3	100
9	100	80	100	90	100
10	100	100	100	96.6	100

Table 2 Performance metrics of neural network classifier for ten users

User	Classifier metrics (neural network) (%)			
	ACC	PRE	REC	F1
1	97.3	97.37	97.3	97.33
2	96.6	96.8	96.6	96.66
3	96	95.97	96	95.97
4	97.3	97.53	97.3	97.29
5	96	96.08	96	95.98
6	98.6	98.75	98.6	98.67
7	98	98.1	98	97.99
8	94.6	94.93	94.6	94.75
9	94	94.27	94	93.94
10	99.3	99.3	99.3	99.33

$$\text{Precision(PR)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (11)$$

$$\text{Recall(RE)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (12)$$

$$\text{F1-Score(F1)} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

$$\text{Accuracy(ACC)} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (14)$$

Table 3 Gesture accuracy of SVM classifier for ten users

User	Gesture accuracy (%)				
	CF	IE	ME	CG	RE
1	96.6	96.6	100	90	100
2	96.6	100	100	93.3	100
3	96.6	96.6	96.6	93.3	100
4	100	100	90	100	100
5	90	96.6	96.6	96.6	96.6
6	100	100	100	96.6	100
7	100	100	100	96.6	100
8	93.3	96.6	96.6	86.6	100
9	100	66.6	96.6	90	100
10	100	100	96.6	96.6	100

After training, validation of the model as seen by Tables 1 and 3 indicates the percentage of accuracy of every gesture for each of the ten users for Artificial Neural Network and SVM, respectively. As given in Tables 2 and 4, a validation accuracy of 96% could be observed for Artificial Neural Network, and an accuracy of 95.3% for SVM, respectively, for User 5. The result observed in this study is comparable with results of other contemporary work which show similar results in terms of accuracy in the range of 90% or higher. The confusion matrix for Artificial Neural Network and SVM are given in Tables 5 and 6, respectively.

Table 4 Performance metrics of SVM classifier for ten users

User	Classifier metrics (SVM) (%)			
	ACC	PRE	REC	F1
1	96.6	96.68	96.66	96.64
2	98	98.18	98	98.01
3	96.6	96.68	96.66	96.66
4	98	98.18	98	97.99
5	95.3	95.9	95.33	95.44
6	99.3	99.35	99.33	99.33
7	99.3	99.35	99.33	99.33
8	94	94.49	94	94.06
9	90.6	91.62	90.66	90.42
10	98.6	98.7	98.6	98.6

Table 5 Confusion matrix for testing data set for User 5 using neural network classifier

	CF	IE	CG	ME	RE
CF	40	0	0	0	0
IE	0	23	4	0	0
CG	0	3	21	0	0
ME	0	2	0	26	0
RE	0	0	0	0	30

Table 6 Confusion matrix for testing data set for User 5 using SVM classifier

	CF	IE	CG	ME	RE
CF	37	0	0	3	0
IE	1	17	9	0	0
CG	0	1	22	1	0
ME	0	2	0	26	0
RE	0	2	0	0	28

5 Conclusion

This paper describes in detail an end-to-end methodology including data acquisition, data segmentation, feature extraction, and classification of hand gestures and investigated the performance of the Artificial Neural Network Classifier with a commercial off the shelf surface electromyography device for a personalized human-machine interface. The significant insight that this study showed was that when a single Artificial Neural Network was used for classifying gestures taking as input ten features for each of the eight sensors gave a significantly better performance than eight separate Artificial Neural Networks which took ten features each as input, with a majority voting for classification at the output. The results were consistent when observed across samples from ten different users. The results indicates that there is scope for implementing online classification system with the same methodology with the identification of suitable hardware and optimized implementation which could be used in Myo-electric control systems for human-machine interface systems such as a human assistive devices or exoskeletons. There is also scope for implementing overlapping window for data segmentation.

Acknowledgements The authors would like to thank Department of Electronics, Mangalore University, and Flashflow Technologies (OPC) Private Limited, for their support during the research work by providing access to a variety of journals which has been tremendously helpful in guiding this work and for all the technical infrastructure and equipment provided for establishing the experimental setup which are immensely critical for this work.

Declaration We have taken permission from competent authorities to use the data as given in the paper. In case of any dispute in the future, we shall be wholly responsible.

References

1. Englehart, K., Hudgins, B., Parker, P.A., Stevenson, M.: Classification of the myoelectric signal using time-frequency based representations. *Med. Eng. Phys.* **21**(6–7), 431–438 (1999)
2. Englehart, K., Hudgins, B., Parker, P.A.: Time frequency based classification of the myoelectric signal: static vs. dynamic contractions. In: Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No.00CH37143), Chicago, IL, USA, vol. 1, pp. 317–320. IEEE (2000)
3. Englehart, K., Hudgins, B., Parker, P.A.: A wavelet based continuous classification scheme for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* **48**(3), 302–311 (2001)
4. Englehart, K., Hudgins, B.: A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* **50**(7), 848–854 (2003)
5. Hudgins, B., Parker, P.A., Scott, R.N.: A new strategy for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* **40**(1), 82–94 (1993)
6. Hargrove, L.J., Englehart, K., Hudgins, B.: The effect of electrode displacements on pattern recognition based myoelectric control. In: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, USA, pp. 2203–2206. IEEE (2006)
7. Hargrove, L.J., Englehart, K., Hudgins, B.: A comparison of surface and intramuscular myoelectric signal classification. *IEEE Trans. Biomed. Eng.* **54**(5), 847–853 (2007)
8. Oskoei, M.A., Hu, H.: Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Trans. Biomed. Eng.* **55**(8), 1956–1965 (2008)
9. Junez, G.P., Terriza, J.H.: Hand gesture recognition based on sEMG signals using Support Vector Machines. In: 2016 IEEE 6th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, pp. 174–178. IEEE (2016)
10. Sueaseenak, D., Khawdee, C., Pakornsirikul, N., Sukjamsri, C.: A performance of modern gesture control device with application in pattern classification. In: 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, pp. 428–431. IEEE (2017)
11. Mendez, I., Hansen, B.W., Grabow, C.M., Smedegaard, E.J.L., Skogberg, N.B., Uth, X.J., Bruhn, A., Geng, B., Kamavuako, E.N.: Evaluation of the Myo armband for the classification of hand motions. In: 2017 International Conference on Rehabilitation Robotics (ICORR). London, pp. 1211–1214. IEEE (2017)
12. Hargrove, L.J., Englehart, K., Hudgins, B.: A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control. *Biomed. Sig. Process. Control* **3**(2), 175–180 (2008)
13. Young, A.J., Hargrove, L.J., Kuiken, T.A.: Improving myoelectric pattern recognition robustness to electrode shift by changing inter electrode distance and electrode configuration. *IEEE Trans. Biomed. Eng.* **59**(3), 645–652 (2012)
14. Kaufmann, P., Englehart, K., Platzner, M.: Fluctuating EMG signals: investigating long-term effects of pattern matching algorithms. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, pp. 6357–6360. IEEE (2010)
15. Haris, M., Chakraborty, P., Rao, B.V.: EMG signal based finger movement recognition for prosthetic hand control. In: 2015 Communication, Control and Intelligent Systems (CCIS), Mathura, India, pp. 194–198. IEEE (2015)
16. Kelly, M.F., Parker, P.A., Scott, R.N.: The application of neural networks to myoelectric signal analysis: a preliminary study. *IEEE Trans. Biomed. Eng.* **37**(3), 221–230 (1990)
17. Saponas, T.S., Tan, D.S., Morris, D., Turner, J., Landay, J.A.: Making muscle-computer interfaces more practical. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems—CHI ’10. Atlanta, Georgia, USA, pp. 851–854. ACM Press (2010)
18. Tiwari, D.K., Bhateja, V., Anand, D., Srivastava, A., Omar, Z.: Combination of EEMD and morphological filtering for baseline wander correction in EMG signals. In: Proceedings of 2nd International Conference on Micro-electronics, Electromagnetics and Telecommunications. Singapore, pp. 365–373. Springer (2018)

19. Srivastava, A., Bhateja, V., Tiwari, D.K., Anand, D.: AWGN suppression algorithm in EMG signals using ensemble empirical mode decomposition. In: Intelligent Computing and Information and Communication. Singapore, pp. 515–524. Springer (2018)
20. Mishra, A., Bhateja, V., Gupta, A., Mishra, A., Satapathy, S.C.: Feature fusion and classification of EEG/EOG signals. In: Soft Computing and Signal Processing, Singapore, pp. 793–799. Springer (2019)
21. Kizirian, A.: Muscles of the Forearm. <https://antranik.org/muscles-of-the-forearm>. Accessed 23 Jan 2019
22. Arief, Z., Sulistijono, I.A., Ardiantsyah, R.A.: Comparison of five time series EMG features extractions using Myo Armband. In: 2015 International Electronics Symposium (IES), Surabaya, Indonesia, pp. 11–14. IEEE (2015)
23. Amirabdollahian, F., Walters, M.L.: Application of support vector machines in detecting hand grasp gestures using a commercially off the shelf wireless myoelectric armband. In: 2017 International Conference on Rehabilitation Robotics (ICORR), London, pp. 111–115. IEEE (2017)

Text-Convolutional Neural Networks for Fake News Detection in Tweets



Harsh Sinha, Sakshi, and Yashvardhan Sharma

Abstract With the widespread use of online social networking websites, user-generated stories and social network platform have become critical in news propagation. The Web portals are being used to mislead users for political gains. Unreliable information is being shared without any fact-checking. Therefore, there is a dire need for automatic news verification system which can help journalists and the common users from misleading content. In this work, the task is defined as being able to classify a tweet as real or fake. The complexity of natural language constructs along with variegated languages makes this task very challenging. In this work, a deep learning model to learn semantic word embeddings is proposed to handle this complexity. The evaluations on the benchmark dataset (VMU 2015) show that deep learning methods are superior to traditional natural language processing algorithms.

Keywords Social media · Twitter · Fake news

1 Introduction

Since social interactions are rising significantly on online social networks, deceptive practices that misuse the system have increased. Rapid dissemination of misleading opinions can also have devastating effects, especially during natural calamities like hurricanes or terrorist attacks.

Fake news refers to any multimedia content which contains misleading information about the event it is associated with. For example, a user may post an image out of context. It is also seen that users share pictures which are concerned with

H. Sinha · Sakshi (✉) · Y. Sharma
Birla Institute of Technology and Science, Pilani, India
e-mail: p20180437@pilani.bits-pilani.ac.in

H. Sinha
e-mail: h20130838@pilani.bits-pilani.ac.in

Y. Sharma
e-mail: yash@pilani.bits-pilani.ac.in

some other similar event. A user may manually edit or morph an image as a form of amusement. On the other hand, there are genuine or real news which represent the event truthfully. Such posts are helpful for the community to make the community aware and safe. There are another class of posts which are fake but are propagated in a sarcastic manner.

In this paper, a deep learning approach is employed for supervised learning for benign classification of tweets as real or fake. Fake news detection is arduous as fact-based checking for news is not feasible. To solve the problem, the approach must be able to learn latent representation of fake news data. Therefore, the paper proposes a Convolutional Neural Network (CNN)-based deep learning approach to learn specific latent representation for accurate classification.

Prior works have used techniques such as graphs [2, 7, 9] and anomaly detection [10]. However, this work focuses on extracting higher-level representations from raw input text. With advent of GPUs, there has been significant development in deep learning especially CNNs. The primary cause which lead to proliferation of CNNs across domains is its agility in reducing variations and extracting spatial correlations.

Researchers model text as sequences. Neural architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) are employed for sequence processing. In this work, one-dimensional CNNs are used as CNNs are competent in extracting space-invariant features. RNNs are useful in predicting the areas such as machine translation or image captioning. However, CNNs are superior in classifying a sentence. CNNs can extract latent features pivoted for accurate classification. Moreover, CNNs are superior as they are extremely efficient and fast in comparison with RNNs. This work focuses on learning an optimal CNN that can be successfully applied for text classification.

The rest of the paper is structured as follows. Section 2 describes the prior techniques used for rumor detection. An overview of the proposed methodology is available in Sect. 3. Section 4 details the experimental set-up and preprocessing methods. A discussion of the outcomes and relative performance measures is presented in Sect. 5. Finally, Sect. 6 summarizes the key contributions of the work.

2 Related Work

Rumors over the Internet have been addressed specifically in the domains of Web spam detection. The initial taxonomy was proposed by Gyongyi et al. [5]. Further, to combat Web spam Castillo et al. [2] used a graph-based structure to infer link-based and content-based dependencies between Web pages. Seo et al. [9] developed a way to study how rumors spread on social networks. It was found that misleading information is generated from a few sources which is re-posted by several other users. Similarly, the dissemination of tweets was studied extensively by Mendoza et al. [7]. They present tweets specifically propagated during crisis and emergencies. Researchers have also tried to employ user characteristic information to model a



Fig. 1 Framework for CNN based fake news detection

binary classification problem of users as spammers and non-spammers [1]. A similar study on user information was carried out by Stringhini et al. [10] by specifically studying anomalous behavior.

In this work, a deep learning approach is proposed to learn latent features that credibly differentiates a real news from fake tweet. Similar studies have been conducted by Gupta et al. [4] which detect fake news during Hurricane Sandy. However, the difference lies in obtaining a realistic estimate by conducting experiments on benchmark datasets which are not restricted to a particular crisis event.

3 Proposed Methodology

Figure 1 displays the block diagram of the proposed approach. The major components of the proposed framework include preprocessing, embedding matrix generation, and classification.

3.1 Data Preprocessing

Tweets crawled from Twitter API contains non-ASCII characters which have to be removed for efficient classification. First of all the HTML tags were removed as they cannot be converted to text. The proposed model attempts to learn difference in latent natural language constructs to classify a tweet as fake or real. Thus, it does not add value for text-based classification. Several other preprocessing steps include removal of URLs, UTF-8 BOM, and hashtags. The text corpus was used to learn an embeddings matrix.

3.2 Embedding Matrix

Embedding matrix is a conceptual idea to model text as points in a n -dimensional hyperspace or more conveniently, as a ‘one-hot’ encoded vector. The n -dimensional vector represents different words in the sentence. Assuming one word per dimension can lead to a very high-dimensional space, so the vectors are transformed into lower vector subspace. In the n -dimensional hyperspace, product of one-hot encoded vector

with an embedding matrix results in a word embedding. The embedding matrix can be represented as $W \in \mathbb{R}^{e \times n}$ where e represents the embedding dimensionality and n is size of vocabulary. This reduces the input size and avoids overfitting. Finally, each word can be represented by a e -dimensional vector, and every tweet is composed of several words.

3.3 Classification

The embedding matrix representing a tweet can be used for classification using a CNN. The first layer of proposed CNN represents the input layer which feeds tweets as matrices as explained in Sect. 3.2. These high-dimensional representation is reduced to a subspace using an embedding layer. The output matrix from the embedding layer $I \in \mathbb{R}^{w \times h \times c}$ where w, h and c are matrix width, height, and number of channels, respectively. A filter matrix $K \in \mathbb{R}^{k \times k \times c}$ is convolved with the matrix I , which results in n activation maps. A convolution layer is followed by a global max-pooling layer which subsamples the resultant activation maps of the previous layer. A pooling layer is used to obtain an efficient representation of data, while rejecting unimportant spatial information. A pooling layer with filter size $k \times k$ computed on a matrix of size $I \in \mathbb{R}^{w \times h \times c}$ generates a matrix of size $P \in \mathbb{R}^{\frac{w}{k} \times \frac{h}{k} \times c}$. However, a global max-pooling layer is used in proposed methodology which outputs a matrix of size $P \in \mathbb{R}^{\frac{w}{k} \times \frac{h}{k} \times 1}$. A global max-pooling layer is very helpful in language processing domains. As dense layers are prone to overfitting, global max-pooling layer preserves the spatial information while reducing the number of parameters for accurate classification. In the proposed architecture, three different convolutional layers are used with their respective global max-pooling layers as shown in Fig. 2. Finally, all the matrices are concatenated and fed for classification using two dense

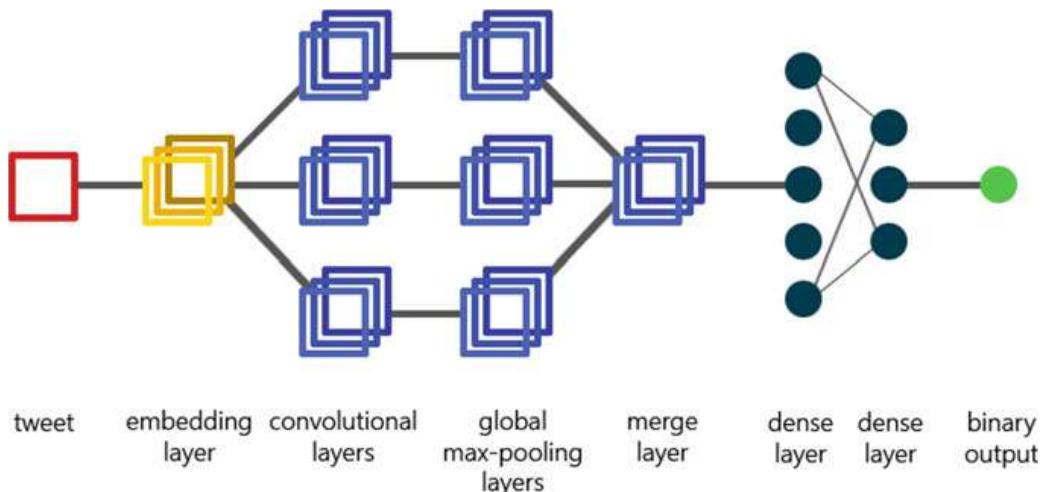


Fig. 2 Proposed CNN architecture

layers. In order to reduce overfitting, every convolutional layer and dense layer is succeeded by a dropout layer. The final softmax loss was replaced with binary-crossentropy loss. The softmax loss is explained in (1).

$$L(x_a, x_b) = -\log \left(\frac{e^{f(x_b)}}{\sum_{j=1}^m e^{f(x_b)}} \right) = \log \left(\sum_{j=1}^m e^{f(x_b)} \right) - f(x_b) \quad (1)$$

where L denotes the softmax loss between two samples x_a and x_b , f denotes the linear transformation of sample ($f(x) = W_i \cdot x + b$), using W as the weight matrix and b as bias).

The binary-crossentropy loss is depicted in (2)

$$L(x_a, x_b) = (y_b \cdot \log(f(x_b)) + (1 - y_b)\log(1 - f(x_b))) \quad (2)$$

where y_b denotes the true label of the sample. Unlike the softmax loss which depends on Boltzmann's distribution, binary-crossentropy loss depends on Shannon's information entropy. The binary-crossentropy loss takes into account each component independently. The goal of proposed deep neural architecture is to learn suitable filters using back-propagation for accurate classification of tweet as fake or real.

Fig. 3 Wordcloud depicting the frequency of words in tweets used for training

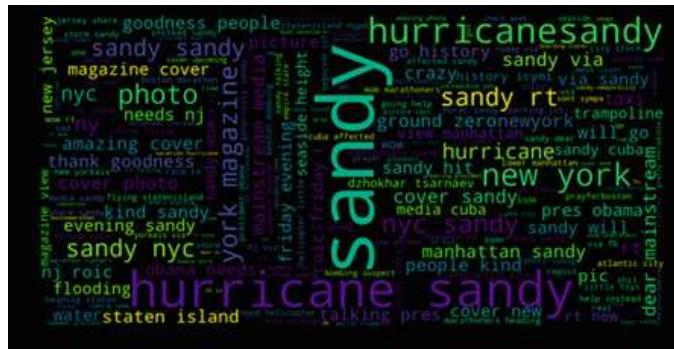
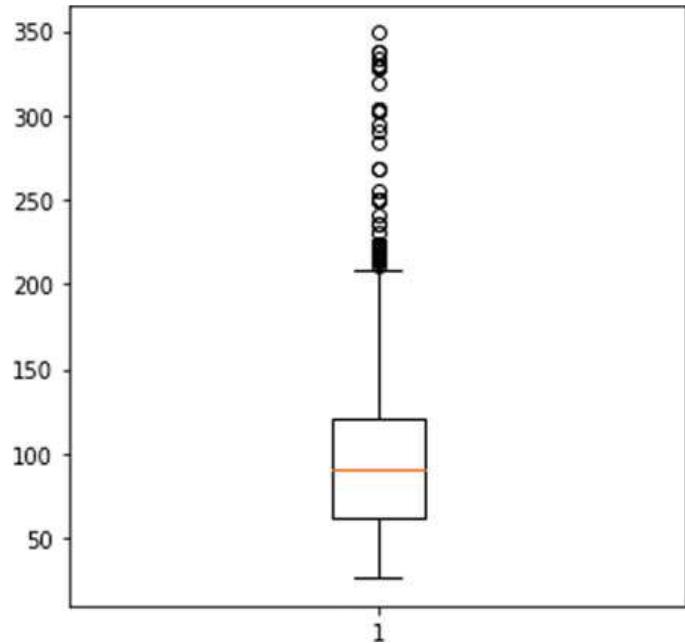


Fig. 4 Wordcloud depicting the frequency of words in tweets used for testing



Fig. 5 Boxplot depicting distribution of length of tweets (number of characters pr tweet)



4 Experiments

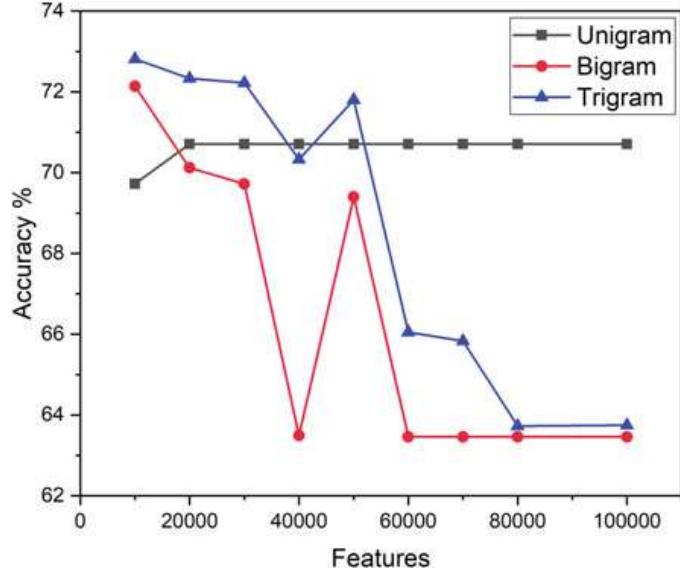
The following section explains various datasets, experiments, and different hyperparameters used for performance evaluation of the proposed methodology.

The proposed CNN is evaluated in terms of average recognition accuracy.

4.1 Dataset

The Verifying Multimedia Use (VMU) 2015 dataset contains a corpus of tweets classified as real and fake with their images shared on the social networking Web site Twitter. The dataset was used in Verifying Multimedia Use Workshop 2015 [3]. The dataset consists of tweet ID, tweet text, user ID, associated image ID, associated username, timestamp, and label as real or fake. The training dataset collects tweets associated with events such as Boston Marathon, Columbian Chemicals, Hurricane Sandy, Malaysia Airlines MH-370, and Sochi Olympics. However, the test dataset contains tweets associated with events such as Garissa Attacks and Nepal Earthquake. The illustration presents the datasets as wordclouds in Figs. 3 and 4.

Fig. 6 Classification accuracy of different n -gram models with Logistic Regression



4.2 Feature Extraction

The distribution of length of tweets is shown in Fig. 5. The distribution is not correct as Twitter's character limit is 140 characters. Thus, the tweet text is cleaned such that it contains ASCII characters only. The several preprocessing steps included removal of HTML tags, UTF-8 BOM, non-english characters, and URLs. Any special symbols such as hashtags and '@' mentions were also removed.

Further, for classification using CNN each tweet is represented by a vector, where each word is represented by a natural number using a tokenizer word index. Each vector representing a tweet is padded with zeros such that all the tweets have equal length.

5 Results and Discussion

In order, to learn the baseline accuracy for classification, the several n -gram models are developed. The classification accuracy by logistic regression in conjunction with n -gram based feature extraction is presented in Fig. 6. The plot depicts classification accuracy for uni-gram, bi-gram, and tri-gram models evaluated w.r.t. feature vector size. The best accuracy is attained by tri-gram model for a feature vector size of 30,000 achieving an accuracy of 72.2%.

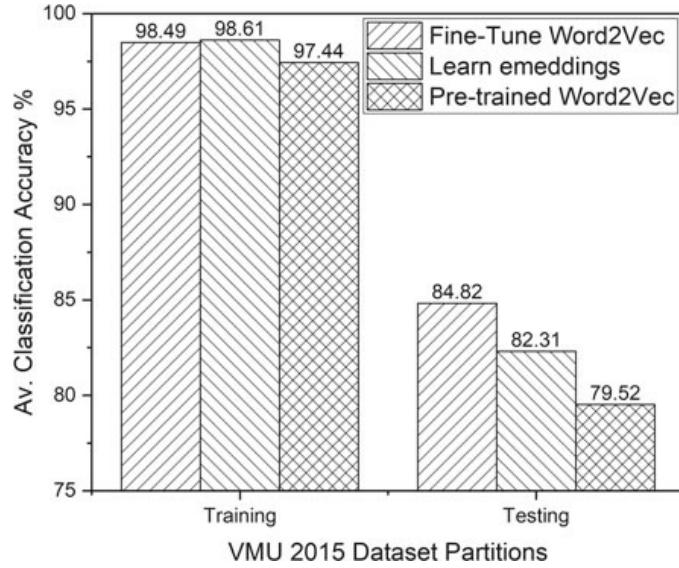
The dataset was also evaluated using Doc2Vec [6] models, namely Distributed Bag of Words (DBOW), Distributed Memory (PV-DMM) Mean, and Distributed Concatenated (PV-DMC). The respective accuracies obtained are given in Table 1. However, the accuracy obtained are inferior as the tweets can barely be considered as documents.

The Doc2Vec model extends the idea of Word2Vec [6]. As the dataset contains tweets which are analogous to sentences, a Word2Vec model can learn representations

Table 1 Average classification accuracy for Doc2Vec models trained on different n -gram features

	Uni-gram (%)	Bi-gram (%)	Tri-gram (%)
DBOW	66.78	67.4	67.91
PV-DMM	64.29	64.98	64.98
PV-DMC	63.46	65.6	66.36

Fig. 7 Classification performance of ANN with Word2Vec

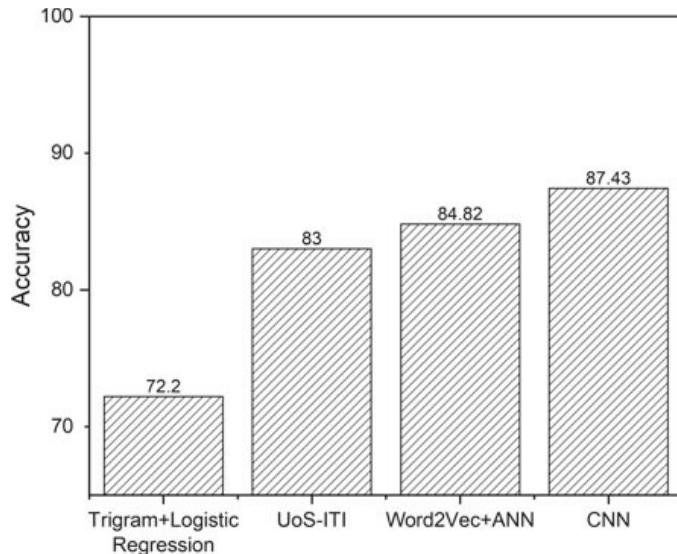


of tweets at the word-level. Further, a two-layer artificial neural network (ANN) is trained on Word2Vec embeddings. The models have three variants as shown in Fig. 7. First of all, the neural network model is trained on pre-trained Word2Vec embedding, restricting re-training of Word2Vec embedding. Secondly, the model is started from Word2Vec model, but it is allowed to fine-tune embedding values to achieve higher classification accuracy. Finally, the ANN model was allowed to learn embeddings from scratch. We observe that fine-tuned Word2Vec model achieves superior test accuracy of 84.82%. Moreover, learning embeddings from scratch has a greater tendency to overfit the training data.

Finally, a CNN is trained on the VMU 2015 dataset to achieve an average accuracy of 87.43% which is better than all the models presented in the study. Although, CNNs are primarily used for image domains, the proposed methodology makes use of one-dimensional convolutions in order to learn a CNN for text data. The three convolutional layers uses kernels of size 2×1 , 3×1 , and 4×1 to simulate n -gram models in a CNN architecture.

The comparative performance of the proposed CNN architecture is presented in Fig. 8. The proposed CNN achieves better accuracy than all other approaches achieving 87.43%. The proposed methodology is also compared by UoS-ITI [8] which use a semi-automatic approach of tokenization, POS tagging, named entity recognition and relational extraction through regex patterns. The proposed approach achieves a

Fig. 8 Comparative Classification performance of proposed methodology



superior accuracy as it aims to learn the latent features which discriminate a fake tweet from a real tweet. Most of the techniques used for text classification are based on handcrafted features such n -gram feature extraction. However, neural networks combine feature extraction and classification in a single algorithm eliminating human biases. This leads to better accuracy as evident by ANN and CNN models which achieve 84.82% and 87.43%, respectively.

6 Conclusion

In this work, a practical and efficient deep learning approach is presented which could discriminate misleading and credible tweets by representing text in n -dimensional vector spaces. The approach achieves an acceptable accuracy of 87.43% which is superior to traditional handcrafted techniques. It establishes that one-dimensional CNNs can be promising in text classification. The high accuracy mends from the fact that CNN is effective in learning a latent representation of the text embedding data by combining feature extraction and classificationn in a single pipeline, pivoted for accurate classification.

References

1. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS), vol. 6, p. 12 (2010)
2. Castillo, C., Donato, D., Gionis, A., Murdock, V., Silvestri, F.: Know your neighbors: web spam detection using the web topology. In: Proceedings of the 30th Annual International ACM

- SIGIR Conference on Research and Development in Information Retrieval, pp. 423–430. ACM (2007)
- 3. Boididou, C., Papadopoulos, S., Zampoglou, M., Apostolidis, L., Papadopoulou, O., Kompatiari, Y.: Detection and visualization of misleading content on Twitter. *Int. J. Multimedia Inf. Retrieval* **7**(1), 71–86 (2018)
 - 4. Gupta, A., Lamba, H., Kumaraguru, P., Joshi, A.: Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 729–736. ACM (2013)
 - 5. Gyongyi, Z., Garcia-Molina, H.: Web spam taxonomy. In: First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005) (2005)
 - 6. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
 - 7. Mendoza, M., Poblete, B., Castillo, C.: Twitter under crisis: Can we trust what we RT? In: Proceedings of the First Workshop on Social Media Analytics, pp. 71–79. ACM (2010)
 - 8. Middleton, S.: Extracting Attributed Verification and Debunking Reports from Social Media: Mediaeval-2015 Trust and Credibility Analysis of Image and Video (2015)
 - 9. Seo, E., Mohapatra, P., Abdelzaher, T.: Identifying rumors and their sources in social networks. In: Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III, vol. 8389, p. 83891I. International Society for Optics and Photonics (2012)
 - 10. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proceedings of the 26th Annual Computer Security Applications Conference, pp. 1–9. ACM (2010)

Effect of Soil and Climatic Attribute on Greenhouse Gas Emission from Agriculture Sector



Pranali K. Kosamkar and Vrushali Y. Kulkarni

Abstract Agriculture sector is a major contributor to global greenhouse gases (GHGs) emission and thus to anthropogenic climate change. In the proposed system, we used soil attributes, i.e., soil type, soil humidity, soil temperature, Ph value, soil moisture, and climatic attributes, i.e., temperature, humidity, wind speed, pressure, and location to analyze and predict the emission of greenhouse gases CO₂ and CH₄ from Pune, India. We used different regression techniques and deep learning model to analyze and predict the emission of CO₂ and CH₄ for different crops and season wise also. The result indicated that the decision tree regressor gives good result, Root Mean Square Error (RMSE) values 0.032930 and 0.026116 for emission of CO₂ and CH₄ as compared to other algorithm used. The deep learning model works best for 4 layers of sequential neural network. The RMSE values for number of epoch 1000 with different layers are 13.18, 7.87, and 10.36. Thus the effect of soil and climate attributes makes the difference in the greenhouse gas emission from agriculture field.

Keywords Greenhouse gases · Agriculture · Deep learning · Soil · Climate

1 Introduction

Agriculture is one of the primary sources of livelihood in India. With having second largest population in the world, India is highly dependent on agriculture. To provide the food to growing population, farmers adapt different practices without knowing the adverse effect on environment and human life. Such practices often lead to greenhouse gas emission which ultimately leads to global warming. The greenhouse gases are in the form of carbon dioxide (CO₂), nitrogen oxide (N₂O), and methane (CH₄). Agriculture activities are one of the major contributors of GHGs

P. K. Kosamkar (✉) · V. Y. Kulkarni
MIT-World Peace University, Pune, Maharashtra 411038, India
e-mail: pranali.kosamkar@mitwpu.edu.in

V. Y. Kulkarni
e-mail: vrushali.kulkarni@mitwpu.edu.in

emission. Due to the rise of greenhouse gases (GHGs) into atmosphere, there is an increase in average global temperature. GHGs emissions from agriculture field are generally associated to different agricultural practices such as Enteric Fermentation, Manure Management, Rice Cultivation, Synthetic Fertilization, Agriculture soil, Crop Residues, Burning Savanna, Land use change, etc. The Intergovernmental Panel on Climate Change (IPCC) produces reports that support the United Nations Framework Convention on Climate Change (UNFCCC), which is the main international accord on climate change [1–3]. The ultimate objective of the UNFCCC is to “stabilize greenhouse gas concentrations in the atmosphere at a level that would prevent dangerous anthropogenic interference with the climate system” [4]. Therefore, the impact of the agricultural management practices is important to study GHGs emission from agriculture sector. The paper is organized into the following sections. Section 1 gives the introductory part and importance of reducing the emission of greenhouse gases from agriculture sector. Section 2 gives the current work done for reducing the GHGs emission using different techniques like deep learning, and machine learning. Section 3 includes proposed methodology for calculating the emission of greenhouse gases. Section 4 discusses the performance analysis and finally Sect. 5 concludes the paper.

2 Literature Survey

We did the survey on how artificial intelligence, machine learning can be used to reduce the emission of GHGs using different soil attributes, climatic attributes, and energy used in the agriculture sector. Panday and Nkongolo [5] have considered the attributes soil air and water to study the effect of these attribute on greenhouse gases emissions in corn/soybean field for a silt loam soil type. They have shown that there is significant correlation between greenhouse gases and soil pore space indices and soil water [5]. Agriculture activities like selection of crops, cropping pattern, and harvesting mostly depend on weather and climate. E. M. Arrieta et al. determined the carbon and energy footprint plus carbon and energy efficiency from soybean and maize crops. They found that climate, particularly mean annual precipitation was the major responsible parameter for the large difference in yield, GHGs emission, and energy efficiencies [6]. Agriculture uses energy in different form such as machinery equipment like tractor, fossil fuel, irrigation equipment, use of fertilizer, and chemicals used in the farm. Benyamin Khoshnevisan et al. predicted the wheat production yield and GHGs emission on the basis of energy input used in the farm. They have used the backpropagation neural network to predict the yield and GHGs emission using the coefficient of determination (R^2). Their result shows that electricity, chemical fertilizer, and water for irrigation are the most influential factors in energy consumption [7]. Similarly, Homa Hosseinzadeh-Bandbafha et al. used artificial intelligence methods, i.e., adaptive neuro-fuzzy inference system and artificial neural network (ANN) for predicting GHGs emission and energy output for calf fattening farms. The result shows that adaptive neuro-fuzzy inference system

predicts energy output and greenhouse gas emissions more accurately than the ANN [8]. Homa Hosseinzadeh-Bandbafha et al. used multi-layer neural networks build on backpropagation algorithm for predicting the amount of energy output and amount of greenhouse gas emission resulting from energy consumption of dairy farms [9].

The effect of climate change is studied using Ricardian approach for greenhouse warming under future climate change. The authors found that both temperature and precipitation significantly determine the farmland values for greenhouse warming [10]. A nitrous oxide emission from agricultural soils was reviewed by Diana Signor et al. They analyzed that nitrous oxide is an important greenhouse gas, due to its high global warming potential. They suggested that for mitigating new strategies to reduce the N₂O emission from agriculture soils we should understand the process of N₂O formation and the influencing factors for GHGs emission [11]. There are various methods to calculate the emission of GHGs. Quantification of nitrous oxide emissions from agricultural soils and management impacts was studied by S. J. Del Grosso and W. J. Parton using eddy covariance and day cent method [12]. Tek B. Sapkota et al. mentioned that Nitrogen use rate, frequency of application, tillage, residue management, and manure application are important from cereal crop for emission of GHGs. In addition to bio-physical, socioeconomic factors such as gender, level of education, training on climate change adaption, and mitigation plus access to information also significantly influenced the adoption of technologies contributing to high-yield low-emission pathways. They have used the tool CCAFS Mitigation Options Tool (CCAFS-MOT) for estimating greenhouse gas emission [13]. Mphethe Tongwanea et al. studied the effect of application of synthetic fertilizer, lime and crop residues retained in the field after harvest during field crop production in South Africa. GHGs emission was compared among different crop production and management practices. GHGs emissions were calculated using Agriculture and Land Use National Greenhouse Gas Inventory Software (ALU) [14].

Different management practices are followed to reduce the GHGs emission like conventional tillage, conservational tillage, and no tillage. The impact of conservation tillage (CT) system to reduce GHGs emission is assessed by M. Abdalla et al. They mentioned that adapting CT practices have reduced CO₂ emission plus it helps to improve in soil organic carbon and soil structure. They also identified the disadvantages of adopting CT practices mainly related to N₂O emission. They suggested that depending on climate and soil condition CT practices should be modified to reduce GHGs emission [15]. Zero tillage practices for climate change mitigation are observed by S. Mangalassery et al. They studied existing work to find out effect of minimizing the tillage operation on climate change mitigation opportunities. Based on their analysis they reported reduction in N₂O emission under long-term zero tillage but significant variability exists in N₂O flux information [16]. Use of nitrogen fertilizer to increase crop yield is one of the major source of GHGs emission from agriculture. A meta-analysis of the effect of N application (nitrogen application) is done by SUN Bin-feng et al. on CH₄ emission in rice paddies, CH₄ uptake in upland fields, and N₂O emission. Their result shows that emission of CH₄ to nitrogen input might depend on CH₄ concentration in rice paddy. They mentioned that nitrogen

fertilizer application rate and control of CH₄ uptake and N₂O emission are the critical factors to affect CH₄ uptake. They concluded that while assessing CH₄ and N₂O emission/uptake the influence of application time, cropping system and measurement frequency should be considered [17]. Reduction in emission of greenhouse gas N₂O without comprising on crop yield is demonstrated by Neville Millar et al. Emission of N₂O gas increases following soil management practices especially irrigation. Irrigation when applied with Nitrogen (N) fertilizer and when N fertilizer inputs exceed crop N requirements. Their result shows that to reduce N₂O emission reduction in rate of N fertilizer could be adapted in irrigated spring wheat in Yaqui Valley without comprising grain yield [18]. The effect of rate and time of fertilizer N application (Nitrogen application) is investigated for 2 years on cornfields for N₂O emission by B. J. ZebARTH et al. They used General Linear Model of SAS for statistical analysis for calculation. They suggested that improved fertilizer N management can reduce nitrate intensity in corn production and it will also reduce N₂O emissions in most cases. They provide the facts that improved fertilizer N management may not result in reduced N₂O emissions under some conditions [19].

3 Proposed Methodology

The main aim is to design the system which will capture the real-time data of agriculture soil attributes, i.e., soil type, soil humidity, soil temperature, Ph value, soil moisture, and climatic attributes, i.e., temperature, humidity, wind speed, pressure and location and greenhouse gases to analyze the emission of GHGs, to analyze and predict the crop-wise GHGs emission and to find the most influential parameter for emission of GHGs from agriculture field. Figure 1 shows the overall system architecture of the proposed system.

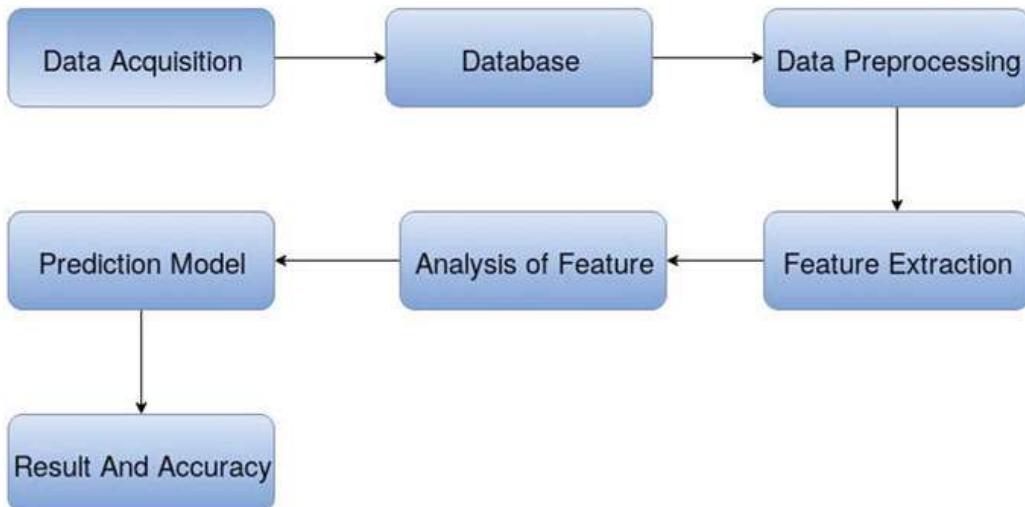


Fig. 1 System architecture

Table 1 Details of experiment site

Particulars	Details
Location	Lavale Phata, Pirangut, Pune, Maharashtra, India
Latitude	18.524633
Longitude	73.691266
Soil type	Clay soil
Crops included in study	Spinach, turnip, and pumpkin

3.1 Data Acquisition

The data for various soil attributes, i.e., soil type, soil humidity, soil temperature, Ph value, soil moisture, and climatic attributes, i.e., temperature, humidity, wind speed, pressure and location and greenhouse gases CO₂ and CH₄ are collected from the agriculture farm the details are mentioned in Table 1. The following sensors are used for collecting the various soils, greenhouse gas, and climatic attributes from experiment site.

DHT 11 Sensor: It measures soil temperature and humidity. The readings are taken after putting the sensor in the pit of 8 cm depth for about 2 min.

2 in 1 Sensor: This sensor displays the value of soil pH and Moisture when its electrodes are inserted into the soil at the depth of 8 cm and values are recorded manually.

MQ-4 and MQ-135 Sensors: MQ-4 sensor is used to measure CH₄ and MQ-135 is used to measure CO₂ emissions from the soil.

Agrinex Solution: It is used for calculating nitrogen, phosphorous, and potassium (NPK) and pH values of the soil.

Google Weather App: Google Weather app provide default readings for climatic attributes, i.e., temperature, humidity, wind speed, pressure, and location.

Table 1 gives the details of experiment site.

3.2 Database Creation

We collected the data from the agriculture farm. The details are mentioned in Table 1. The Raspberry Pi will read the data from different sensors like DHT-11, MQ-4, and MQ-135 Sensors from the experiment site and send it to ThingSpeak. ThingSpeak a cloud server is used for collecting, storing and monitoring the sensor data.

3.3 Data Preprocessing

The categorical field nitrogen, phosphorus, and potassium (NPK) values label is changed into values 0, 1, 2 which replaces the values low, medium, and high which ease the processing of data in algorithm with replace function in python. The dataset is further normalized using MinMax scaler function in python.

3.4 Feature Extraction

All the features of soil attributes and climatic attributes are extracted to find the relationships between the GHGs emission and to analyze which features make the difference in emission of CH₄ and CO₂ for different crops.

3.5 Analysis of Features

In this step soil and climatic attributes are used for analyzing and finding out the most influential factors or attributes for emission of CH₄ and CO₂ using decision tree regressor.

3.6 Prediction Model

Here, various machine learning algorithms such as linear regression, decision tree regressor, and random forest regressor are used to build a prediction model based on database to predict CO₂ and CH₄ emissions from the field.

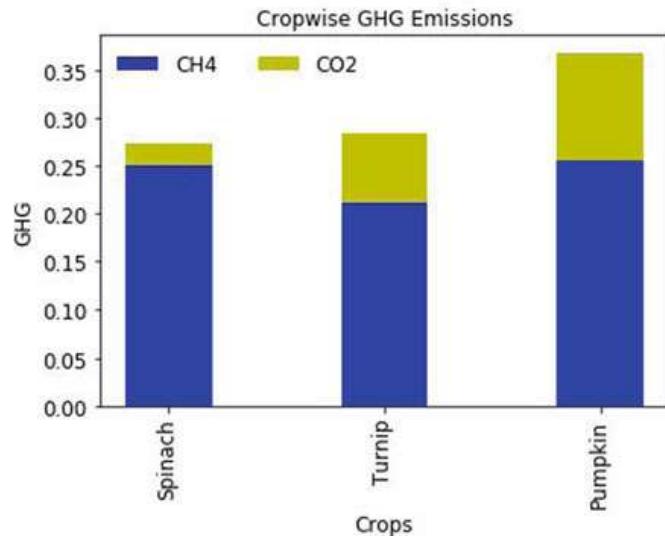
4 Result Analysis

We tried to analyzed and predict CO₂ and CH₄ emission from soil attributes and climatic attributes using machine learning algorithms. We observed that the overall GHGs emission rate was highest for pumpkin other than spinach and turnip. The GHGs emissions were highest in the summer season as compared to winter. The attribute affecting the emission rate of CO₂ is wind speed and for CH₄ is maximum climatic temperature. The performance of the algorithm is evaluated using Root Mean Square Error (RMSE). Table 2 gives the RMSE values for linear regression, decision

Table 2 Comparison of different algorithm using RMSE values for CO₂ and CH₄ greenhouse gas emission

Greenhouse gases/algorithm	Linear regression	Decision tree regressor	Random forest regressor
CO ₂	0.094505	0.032930	0.041039
CH ₄	0.064444	0.026116	0.029275

Fig. 2 Crop-wise CO₂ and CH₄ emission



tree regressor, and random forest regressor. The decision tree regressor gives best RMSE values as compared to linear regression and random forest regressor. For CO₂ and CH₄ gases, the difference between actual and predicted value is best in decision tree regressor 0.032930 and 0.026116, respectively.

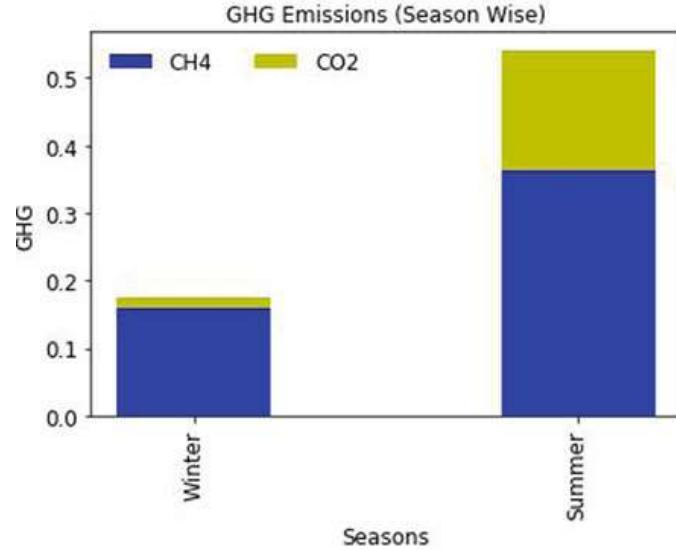
4.1 Crop-Wise GHGs Emission

For comparing crop-wise CO₂ and CH₄ emission, we have considered three crops namely spinach, turnip, and pumpkin. We can observe from the graph of Fig. 2 that the highest CH₄ emission is for spinach and whereas the highest CO₂ emission is for pumpkin. The overall GHGs emission rate was highest for pumpkin.

4.2 Season-Wise GHGs Emission

We can infer from the graph of Fig. 3 that the GHGs emissions were highest in the summer season. This is due to the climatic temperature, as the temperature is higher

Fig. 3 Season-wise CO₂ and CH₄ emission



in summer the emission rates are higher in summer season. As observed from the analysis, we can say that the overall emission rate of CH₄ is more compared to that of CO₂.

4.3 Most Influencing Parameter for Prediction of GHGs Emission

We analyzed the most influencing attribute for CO₂ and CH₄ emission from soil and climatic attributes. The attribute affecting the emission rate of CO₂ mostly is wind speed. The attribute affecting the emission rate of CH₄ most is maximum climatic temperature (Fig. 4).

4.4 Using Deep Learning Approach

Model Accuracy

The graph in Fig. 5 shows the EPOCH values against RMSE values. It gives the idea how the increasing the epoch decreases the RMSE values but the decrement in RMSE slows down after a certain epoch value.

Effect of Number of Layers on Model Accuracy

Figure 6 graph shows the comparison of how number of layers in deep learning model affect the accuracy. The 5 layered approach works better than the 3 layered appeal but performs worse than 4 layered approach.

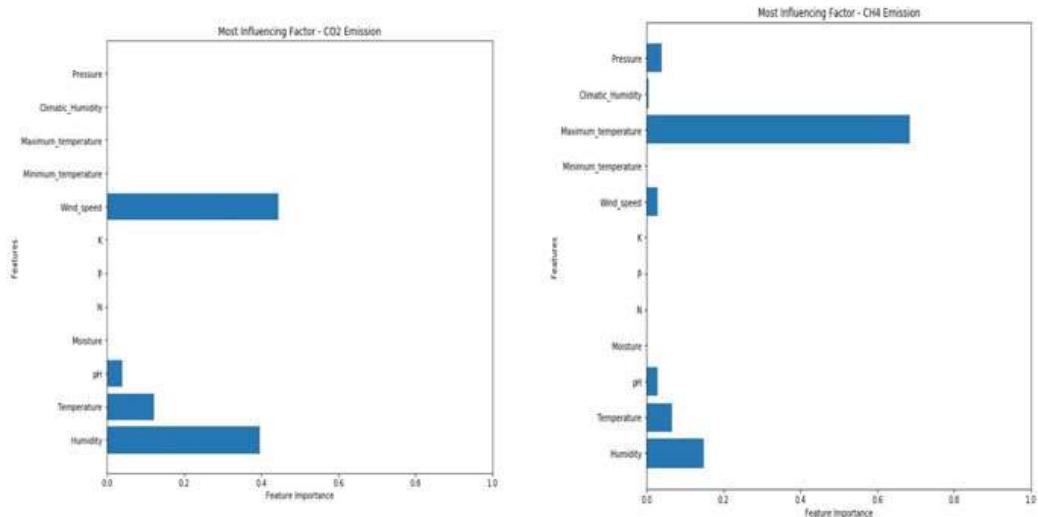


Fig. 4 Most influencing parameter for CO₂ and CH₄ greenhouse gas emission

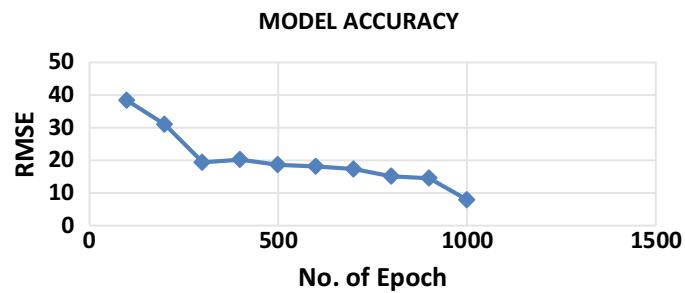


Fig. 5 Model accuracy with number of epoch

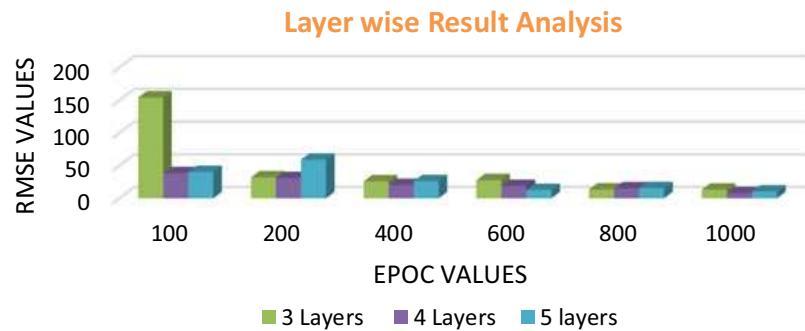


Fig. 6 Layer-wise analysis on model accuracy

5 Conclusion

Dealing with the impact of climate variation on agriculture will require careful management of resources like soil, water, and weather. The proposed system will

capture the real-time data of soil attributes, climatic attributes, and CO₂, CH₄ greenhouse gases using different sensors from the experiment field. We used different algorithms to analyze and predict the greenhouse gases CO₂, CH₄ emission. The result shows that for CO₂ and CH₄ gases the difference between actual and predicted values is best in decision tree regressor 0.032930 and 0.026116, respectively, as compared to linear regression and random forest regressor. The crop-wise CO₂ and CH₄ emission rate was highest for pumpkin as compared to spinach and turnip. We also analyzed the season-wise CO₂ and CH₄ emission the result shows that CO₂ and CH₄ emissions were highest in the summer season. The deep learning model works best for 4 layer of sequential neural network. The RMSE values for number of epoch 1000 are for 3 layers is 13.18, 4 layers is 7.87, 5 layers is 10.36, respectively. Also it shows that very high epoch values should be avoided so that the model does not over fit on the data.

The current need of the hour requires a system which gives the most accurate predictions of GHGs emission which can be then used to provide solutions to control the global warming, greenhouse gas emission. While there has already been a lot of research going on technology aspect, i.e., using deep learning, machine learning, and quantum computing can we solve the global warming, world hunger, and greenhouse gas emission problem instead of using classical computers.

References

1. Introduction to the Convention, UNFCCC, Archived from the original on 8 Jan 2014. Accessed 29 June 2018
2. IPCC: "Principles Governing IPCC Work" (PDF). Approved 1–3 Oct 1998, last amended 14–18 Oct 2013. Accessed 29 June 2018
3. <https://ccafs.cgiar.org/publications/reducing-greenhouse-gasemissions-agriculture-without-compromising-food-security-0#.XEcbKVwzZPY>. Accessed 22 Jan 2019 at 8.14 pm
4. <https://unfccc.int/resource/bigpicture/>. Accessed on 7 June 2019 at 11.20 pm
5. Panday, D., Nkongolo, N.V.: Effect of Soil Air and Water on Greenhouse Gases Emissions in a Corn-Soybean Rotation. Department of Agriculture and Environmental Sciences, Lincoln University, Jefferson City, MO 65101-0029, USA
6. Arrietaa, E.M., Cuchiettia, A., Cabrolb, D., Gonzálezc, A.D.: Greenhouse gas emissions and energy efficiencies for soybeans and maize cultivated in different agronomic zones: a case study of Argentina. *Sci. Total Environ.* **625**, 199–208 (2018)
7. Khoshnevisan, B., Rafiee, S., Omid, M., Yousefi, M., Movahedi, M.: Modeling of energy consumption and GHG (greenhouse gas) emissions in wheat production in Esfahan province of Iran using artificial neural networks. *Energy* **52**, 333–338 (2013). <https://doi.org/10.1016/j.energy.2013.01.028>
8. Hosseinzadeh-Bandbafha, H., Nabavi-Peleesaraci, A., Shamshirband, S.: Investigations of energy consumption and greenhouse gas emissions of fattening farms using artificial intelligence methods. *Environ. Prog. Sustain. Energy.* American Institute of Chemical Engineers (2017). <https://doi.org/10.1002/ep.12604>
9. Hosseinzadeh-Bandbafha, H., Safarzadeh, D., Ahmadi, E.: Modeling output energy and greenhouse gas emissions of dairy farms using neural networks. *Biol. Forum Int. J.* (2015). <https://www.researchgate.net/publication/283571073>
10. Attavanich, W.: The effect of climate change on Thailand's agriculture. MPRA Paper No. 84005, posted 22 January 2018 06:32 UTC. Online at <https://mpra.ub.uni-muenchen.de/84005/>

11. Signor, D., Cerri, C.E.P.: Nitrous oxide emissions in agricultural soils: a review. *Pesq. Agropec. Trop.* **43**(3), 322–338 (2013). e-ISSN 1983-4063. www.agro.ufg.br/pat, jul./set. 2013
12. Del Grosso, S.J., Parton, W.J.: Quantifying nitrous oxide emissions from agricultural soils and management impacts. In: *Understanding Greenhouse Gas Emissions from Agricultural Management ACS Symposium Series*. American Chemical Society, Washington, DC (2011)
13. Sapkota, T.B., Aryal, J.P., Khatri-Chhetri, A., Shirsath, P.B., Arumugam, P., Stirling, C.M.: Identifying high-yield low-emission pathways for the cereal production in South Asia. In: *Mitigation and Adaption Strategies Global Change*. Springer (2018)
14. Tongwanea, M., Mdlambuzi, T., Moeletsia, M., Tsuboa, M., Mliswa, V., Grootboom, L.: Greenhouse gas emissions from different crop production and management practices in South Africa. *Environ. Dev.* **19**, 23–35 (2016)
15. Abdalla, M., Osborne, B., Lanigan, G., Forristal, D., Williams, M., Smith, P., Jones, M.B.: Conservation tillage systems: a review of its consequences for greenhouse gas emissions
16. Mangalassery, S., Sjögersten, S., Sparkes, D.L., Mooney, S.J.: Examining the potential for climate change mitigation from zero tillage. MS received 27 Sept 2013, revised 2 July 2014. Accepted TBC Aug 2014
17. Sun, B., Zhao, H., Lu, Y., Lu, F., Wang, X.: The effects of nitrogen fertilizer application on methane and nitrous oxide emission/uptake in Chinese croplands. *J. Integr. Agric.* **15**(2), 440450 (2016)
18. Millara, N., Urrea, A., Kahmark, K., Shcherbak, I., Philip Robertson, G., Ortiz-Monasterio, I.: Nitrous oxide (N_2O) flux responds exponentially to nitrogen fertilizer in irrigated wheat in the Yaqui Valley, Mexico. *Agric. Ecosyst. Environ.* **261**:125–132 (2018). (<https://doi.org/10.1016/j.agee.2018.04.003>, 0167-8809/ © 2018 The Authors. Published by Elsevier B.V.)
19. ZebARTH, B.J., Rochette, P., Burton, D.L., Price, M.: Effect of fertilizer nitrogen management on N_2O emissions in commercial corn fields. *Can. J. Soil Sci.* 189–195

Optimal Image Feature Ranking and Fusion for Visual Question Answering



Sruthy Manmadhan and Binsu C. Kovoor

Abstract Visual Question Answering (VQA) is a moderately new and challenging multi-modal task, which endeavors to discover an answer for a given pair of an image and a relating question. This AI-complete task gains attraction from numerous researchers from the areas computer vision (CV) and natural language processing (NLP) due to its various potential applications. The general flow of VQA algorithms consists of image feature extraction, question feature extraction and joint comprehension of these two to generate an appropriate answer. Existing VQA systems did not pay attention to input feature extraction, but only celebrated different ways of multi-modal embedding. This paper proposes to improve the task of VQA by feature-level fusion of visual information. The goal of feature fusion is to consolidate relevant information from two or more feature vectors into a solitary one with additional discriminative power. Unlike simple concatenation, this paper uses discriminative correlation analysis (DCA) for fusion, which is the only method that incorporates the class structure into the feature-level fusion. Since the VQA systems are generally modeled as classification systems by treating the correct answers as classes, class-specific DCA suits well here. The newly created fused feature vectors are close to the right answers and thus raise the role of image understanding in VQA. The experimental results show the effectiveness of the new approach on DAQUAR dataset with mutual information (MI) as an evaluation metric.

Keywords Convolutional neural networks · Discriminative correlation analysis · Feature extraction · Mutual information · Visual question answering

S. Manmadhan () · B. C. Kovoor

Division of Information Technology, Cochin University of Science and Technology, Kochi, Kerala, India

e-mail: sruthym.88@gmail.com

B. C. Kovoor

e-mail: binsu.kovoor@gmail.com

1 Introduction

A new task called Visual Question Answering (VQA) [1] has attracted many researchers in recent years. In its simplest form, a VQA model can be considered as one that answers questions asked for an input image. This task is challenging because a considerable amount of image understanding and natural language understanding is fundamental to have the option to foresee the appropriate response to a given image-question pair. Its highlighted difficulty lies with joint comprehension of multi-modal features got from image and question to find the correct answer and being celebrated by VQA research community for past four to five years. Instead of dynamically generating answers, most of the existing VQA systems model it as a classification task and thus infer the correct answer from a fixed set of possible answers. Variants of this task comprise of *binary VQA* [2], *multiple-choice VQA* [3] and *fill in the blank VQA* [4].

Of the two modalities to be handled, natural language question is being explored by natural language processing (NLP) community since 1970s in the form of textual question answering [5, 6]. Understanding of image poses a significant challenge as images are much higher dimensional and typically noisier than text. Some example instances of the task taken from VQA dataset has been shown in Fig. 1. This demonstrates the need for solving various CV tasks like object recognition, object detection, scene classification, etc. within the context of a VQA system. In this regard, transforming an image to a fixed-length, unique, numeric vector (*feature vector*) assumes a significant job in VQA because an image cannot be straightforwardly encouraged into any of the previously mentioned CV frameworks.

A dominant part of VQA research has concentrated on improving the final multi-modal feature fusion, while the impact of individual features has been disregarded, especially the image feature. All the current methodologies utilize a single image feature vector extracted from a profound deep CNN, e.g., VGGNet [7], GoogLeNet [8] or ResNet [9] trained on the ImageNet [10] dataset. It is difficult to accept that these conventional have fine-grained discriminative power required to answer a wide assortment of visual inquiries. Thus, it is desirable to fuse multiple image feature descriptors to improve the image understanding performance because different features can provide complementary information. Also, individual feature extraction can now be simplified by utilizing the benefits of *Transfer Learning*.

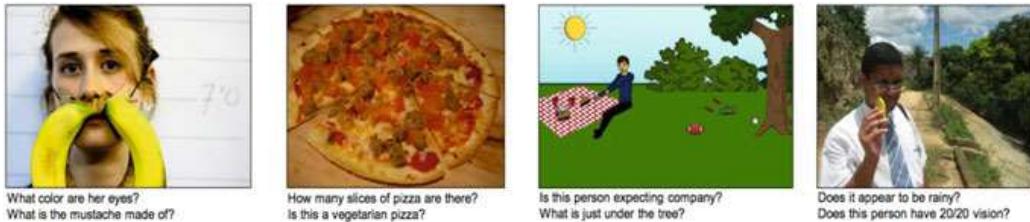


Fig. 1 Sample image-question pairs from VQA dataset [1]

In this paper, we propose a feature ranking followed by DCA-based feature-level fusion of two top-ranked image descriptors. This framework also used clustering and mutual information (MI) metric in various parts of its implementation. In summary, the main contributions of this paper are:

- A generalized ranking and feature fusion (RFF) framework is proposed to aid extraction of optimal image features concerning the task of VQA.
- A novel feature ranking algorithm based on mutual information (MI) is presented to select the top two image features to be fused.
- Efficient use of Discriminative Correlation analysis (DCA)-based feature-level fusion is demonstrated for VQA systems.
- Extensive experiments are carried out on DAQUAR dataset to demonstrate the efficiency and effectiveness of the RFF framework.

2 Related Work

2.1 Visual Question Answering

Highlights from VQA literature disclose that the existing solutions mainly differ in the methodology they have used for combining image and question feature vectors before feeding into answer generation stage. With respect to this, the VQA solutions can be classified into four.

- *Baseline fusion models* which are straight forward techniques like concatenation [11], element-wise addition and element-wise multiplication [1, 2] have been used to map image and textual features to a common space.
- *Encoder-Decoder Models* [3, 12] which may combine CNNs (as encoders of visual features) and long short-term memory (LSTM) networks (as decoders) to generate correct answers forms a firm baseline in VQA literature.
- *End-to-End DNN models* are those which implement a dedicated set of layers for image featurization, question featurization and their joint comprehension. Some well-studied named models from this category include neural module networks (NMN) [13], dynamic parameter prediction network (DPPN) [14], multi-modal residual network (MRN) [15], deep attention neural tensor network (DA-NTN) [16] etc.
- *Attention models* are the celebrities of the current VQA world. The prominent works include word-to-region attention network (WRAN) [17], co-attention network [18] to jointly reason about image and question representation and recently ‘hard attention’ [19] which perform filtering by avoiding unwanted information.

Table 1 Overview of CNN models trained on ImageNet

Successful CNN models	Year	Number of layers	Input dimension	Output dimension	Reported error
AlexNet [10]	2012	8	227×227	4096	16.4
ZFNet [20]	2013	8	227×227	4096	11.7
VGGNet [7]	2014	19	224×224	4096	7.3
GoogleNet [8]	2014	22	229×229	1024	6.7
ResNet [9]	2015	152	224×224	2048	3.57 [better than human]

2.2 *Image Feature Extraction*

VQA requires correct use of two separate information streams to guarantee reliable output; pictures and questions. Studies of ablation have shown regularly that the models using only the natural language question processing performs drastically better than the models using visual information. One can infer the following fact from this; the processing of visual information as part of VQA must further be explored to fill the gap between machines and humans. But, in all of the existing research, this part of VQA model gets frozen to one of the ImageNet winners. Table 1 show details of pre-trained models used for image feature extraction in existing VQA systems. A study done by [21] reveals that it is essential to leverage multiple visual features from distinct sources to truly comprehend an image and answer questions about it.

3 Ranking and Feature Fusion (RFF) Framework

In this section, the novel RFF framework has been presented, which directly addresses a significant step of VQA, the image featurization. First, the overall architecture is presented in Sect. 3.1. Then, its two sub-modules are further detailed in Sect. 3.2 (Feature ranking) and Sect. 3.3 (Feature fusion), respectively.

3.1 *Overall Architecture*

The goal of the proposed RFF framework is to learn more discriminative image features with enriched semantics and contextual details, leading to the correct answer. Figure 2 shows the overall architecture of the proposed system. The input to the framework includes the set of all images contained in the training part of the VQA dataset being examined. For each image, there will be an associated question and corresponding answer. The initial step entails the extraction of naïve features from all images using multiple well-known deep CNN models via transfer learning, aiming to

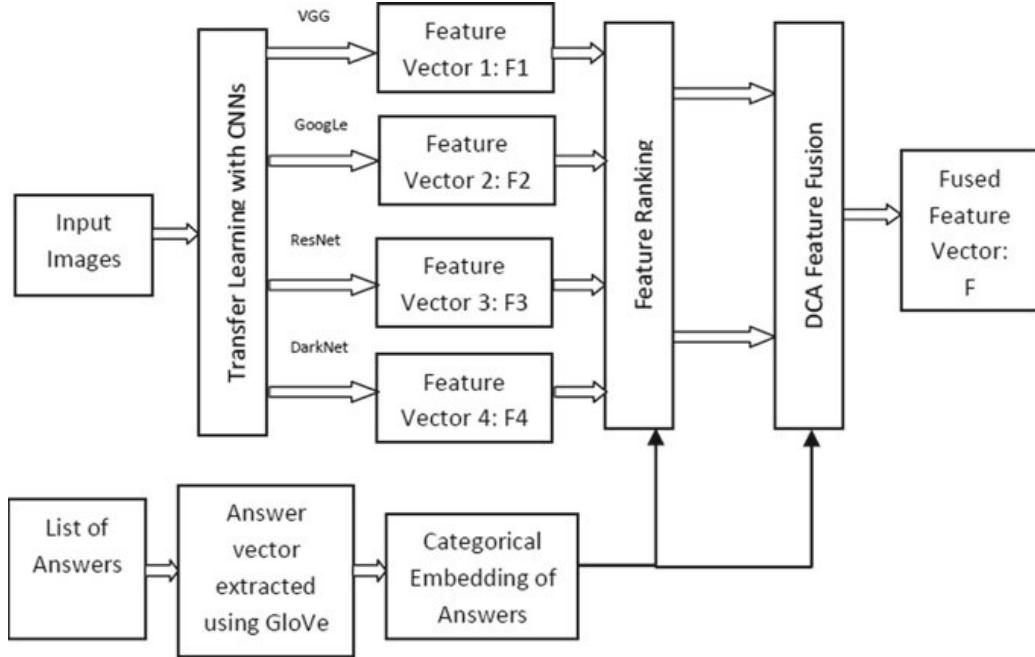


Fig. 2 Overall architecture of RFF framework

capture various visual aspects relevant for different question types. Transfer learning is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks, thereby reducing the huge amount of compute and time resources required to develop neural network models from scratch.

If there are n images in the training set, this step will result in different large numeric matrices as given in Eq. 1, with dimensions with respect to the feature vector length of models used. To avoid memory overflow, it is recommended to project feature matrices into a common lower-dimensional space. In our experiments, we have extracted four different feature vectors for each image, as shown in Fig. 2.

$$F1 \in \mathbb{R}^{n \times p1}; F2 \in \mathbb{R}^{n \times p2}; F3 \in \mathbb{R}^{n \times p3}; F4 \in \mathbb{R}^{n \times p4}; A \in \mathbb{R}^{n \times p3} \quad (1)$$

The lower part of Fig. 2 exhibits another basic input to the system, the list of answers. In an ideal VQA system, we hope that visual features must be able to tell partially about the correct candidate answer. In other words, in VQA systems modeled as classifiers, the answers must be incorporated while ranking and fusing visual features. For this to happen, we need feature vector for each of the candidate answers, which can be done by any well-known NLP word embedding tools like GloVe [22]. This completes the first level feature extraction by contributing another matrix, A , to the community.

3.2 Feature Ranking

Synthesizing the advantages of each feature extractors, Algorithm 1 presents an adaptive weighted ranking scheme. It uses two concepts; categorical embedding of answers and mutual information.

The input A denotes numerical vectors corresponding to all answers. There exist many similar or same answers in the list. So to form a categorical embedding of the same, the first task is to cluster similar answers together. Here, we use Hierarchical DBSCAN (HDBSCAN) [23] clustering algorithm which performs well in clustering data of varying density and varying shape. Another motivation for using HDBSCAN in this context is its speed as the input data size increases.

Algorithm 1 also uses calculated mutual information (MI) [24] between each of the image feature vectors and normalized answer feature. MI is one of many quantities that measure the relationship between two random variables and generally answers the question, ‘How much does one random variable tell about another?’ Here, it gives the information about how much an image feature can tell about the associated correct answer. It is a dimensionless and non-negative quantity. Mathematical definition of MI of two variables X and Y is given by

$$I(X; Y) = \int_y \int_x p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right) \quad (2)$$

Each row of weight matrix stores the maximum MI value obtained by each featurizer for each class. The column with the largest MI value against a class is the best featurizer for that class. So, first rank will be given to the featurizer which is best for more number of classes.

As the dataset becomes large, processing of large feature matrices becomes a primary challenge which could be tackled with upgraded memory or incremental batch-wise processing of the dataset.

Algorithm 1 Adaptive weighted ranking algorithm

Input:	Individual visual feature vectors extracted and projected into a common dimensionality space: $F_i \in \mathbb{R}^{n \times p}; i = \{1, 2, \dots, f\}$ Mutual information between each of the image features and answer vector recorded as $M_i = \mathbb{R}^{n \times f}; i = \{1, 2, \dots, f\}$
Output:	Index of top 2 feature vectors
Step 1:	Perform clustering of answer word vectors and record cluster labels as categorical embedding of answers to be used as class labels. $\text{Answer_Cat} = \mathbb{Z}^n$

(continued)

(continued)

Step 2:	<p>Initialize weight matrix:</p> $\mathbf{w} = \begin{pmatrix} w_{11} & \cdots & w_{1f} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mf} \end{pmatrix}$ <p>$w_{ij} = 0; i = \{1, \dots, m\}; j = \{1, \dots, f\}$</p> <p>m demotes number of unique labels/classes.</p>
Step 3:	<p>for i from 1 to m:</p> <ol style="list-style-type: none"> Pos \Leftarrow Find indices of Answer_Cat where value is i for j from 1 to f: $w_{ij} = \max(M_j[\text{pos}])$
Step 4:	Output top two column indices of \mathbf{w} which hit $\text{argmax}(\mathbf{w}_i)$ mostly.

3.3 Feature Fusion

Here, the feature-level fusion of two top-ranked image features has been done via discriminative correlation analysis (DCA) [25]. DCA eliminates the between class correlations and restricts the correlations to be within classes. DCA is closely related to a widely adopted method of fusion known as canonical correlation analysis (CCA). Both maximize the correlation of corresponding features across the two feature sets. But, the main issue in CCA-based approaches is their ignorance of the class distribution among samples which is vital in pattern recognition tasks. Mathematical explanation of DCA can be found in [25].

Similar to CCA, this approach results in two sets of transformed feature vectors say X and Y those can be either concatenated or added together to have final fused feature vector. The effectiveness of the resulting feature can be evaluated by computing mutual information between the new fused feature vector and the answer vector.

4 Results and Discussions

This section deals with details of experiments and visualizations of results to demonstrate the benefits of the proposed system. All experiments have been conducted on the first significant VQA dataset named DAQUAR [26] with 6794 training and 5674 test question-answer pairs based on images from ‘NYU-Depth V2 Dataset.’

The employed image feature extractors were the profound CNNs pre-trained on ImageNet data such as VGGNet, GoogLeNet, ResNet152 and Darknet53 [27] which is the latest addition to the list of CNNs to aid real-time object detection. Thus, after the first level of feature extraction, we have four image feature vectors as follows:

$$F1 \in \mathbb{R}^{6794 \times 4096}; F2 \in \mathbb{R}^{6794 \times 1024}; F3 \in \mathbb{R}^{6794 \times 2048}; F4 \in \mathbb{R}^{6794 \times 1000} \quad (3)$$

The next module of RFF system is feature ranking which starts with clustering of answers using HDBSCAN. A total of 72 clusters have been obtained by setting the parameters, `min_cluster_size` as 30 and `min_samples` as 15 for DAQUAR answers. After running the ranking algorithm given in Sect. 3.2 (Algorithm 1), GoogLeNet (InceptionV3) was given rank 1, and ResNet152 was given rank 2 with respect to DAQUAR dataset which may change depending on the dataset being used. Out of 72 classes, for 37 classes, GoogLeNet gave maximum mutual information, for 23, ResNet and for 12, VggNet.

Now comes, the exciting results of feature-level fusion using DCA. Quality assessment of visual features was done with computed mutual information (MI) between image and answer. In context of VQA, MI values indicate how much a particular image feature can tell about the corresponding correct answer. For better understanding, Fig. 3 shows the distribution of MI values calculated using four different CNN image featurizers with help of histograms.

Finally, the comparison of individual models and fused vector is presented as a density plot in Fig. 4, and a box plot in Fig. 5, where the readers can quickly identify,

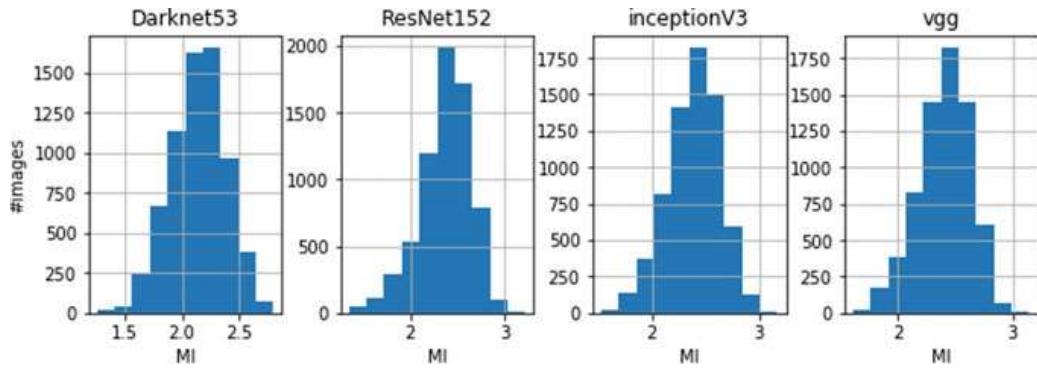


Fig. 3 MI values computed using visual features extracted from individual CNN models

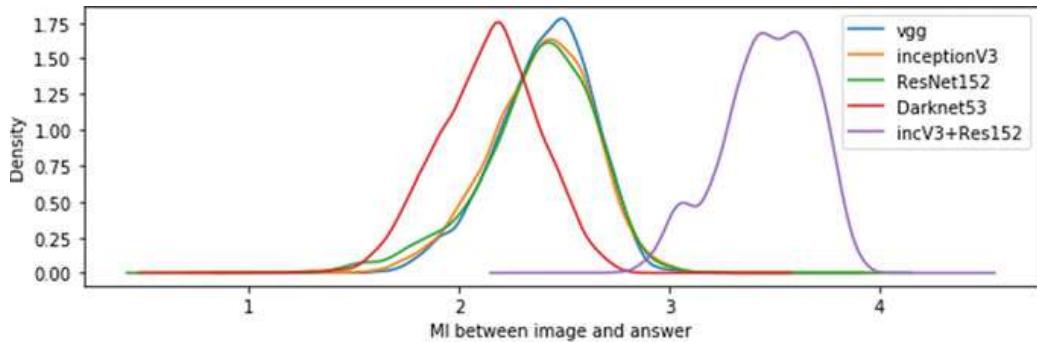


Fig. 4 Density plot showing MI distribution before and after fusion

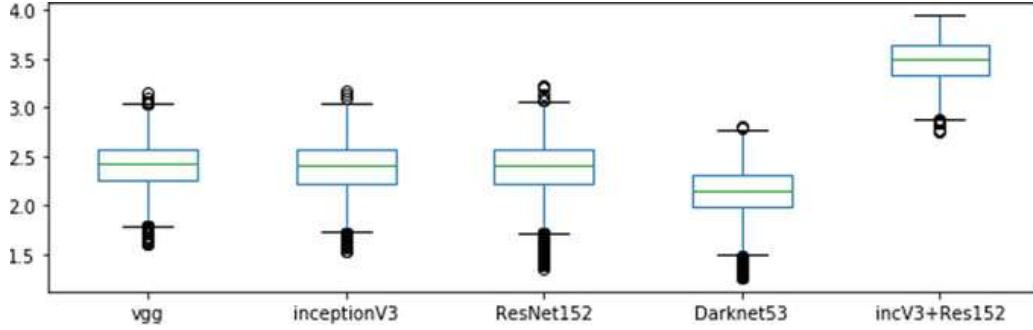


Fig. 5 Box plot showing MI distribution before and after fusion

Table 2 Descriptive statistics of MI values before and after fusion

	vgg	inceptionV3	ResNet152	Darknet53	incV3+Res152
count	6795	6795	6795	6795	6795
Mean	2.405369	2.388787	2.374342	2.141077	3.469251
Std	0.228806	0.244999	0.273675	0.23903	0.218721
Min	1.604141	1.525752	1.353383	1.25422	2.748044
25%	2.260601	2.229616	2.227191	1.983325	3.331878
50%	2.42477	2.407066	2.404343	2.155937	3.486872
75%	2.570092	2.563716	2.567029	2.305979	3.636185
max	3.147144	3.171732	3.218353	2.802623	3.946799

a drastic horizontal shift in the distribution of MI values from the range [0–3] to [2.5–4]. This horizontal scaling implies that, after fusion, the visual features are closer to answers which will help the final decision of classifier after combining with question features.

For those who want to analyze quantitative details of the MI values, refer Table 2. It shows that after fusion (last column of Table 2) the minimum MI value is 2.748044 and maximum is 3.946799, and the standard deviation is minimum compared to other individual distributions. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a broader range of values.

5 Conclusion

This paper takes a simple step toward improving the AI-complete task of Visual Question Answering, explicitly tackling the image featurization step. A general framework named RFF has been presented for optimal image feature ranking and fusion using the concept of discriminative correlation analysis. A series of experiments were

conducted and results were presented to show the benefits on DAQUAR dataset with mutual information as an evaluation metric. RFF creates visual features that are close to answers, thus helps the VQA system to predict the correct answer quickly than using image features from a single CNN model. This must be extended to multiple datasets to identify the different effects of fusion. Also, there exists a possibility to fuse more than two feature vectors at in a hierarchical manner as pointed by [25]. Hence, future research is planned to focus on multi-step fusion and evaluation on multiple datasets which cover different VQA contexts.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
2. Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., Parikh, D.: Yin and Yang: balancing and answering binary visual questions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5014–5022 (2016)
3. Zhu, Y., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7w: grounded question answering in images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4995–5004 (2016)
4. Yu, L., Park, E., Berg, A.C., Berg, T.L.: Visual madlibs: fill in the blank description generation and question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2461–2469 (2015)
5. Fader, A., Zettlemoyer, L., Etzioni, O.: Paraphrase-driven learning for open question answering. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1608–1618 (2013)
6. Xiong, C., Merity, S., Socher, R.: Dynamic memory networks for visual and textual question answering. In: International Conference on Machine Learning, pp. 2397–2406 (2016)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A.: Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Jabri, A., Joulin, A., van der Maaten, L.: Revisiting visual question answering baselines. In: European Conference on Computer Vision, pp. 727–739. Springer, Cham (2016)
12. Wu, Q., Wang, P., Shen, C., Dick, A., van den Hengel, A.: Ask me anything: Free-form visual question answering based on knowledge from external sources. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4622–4630 (2016)
13. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 39–48 (2016)
14. Noh, H., Hongsuck Seo, P., Han, B.: Image question answering using convolutional neural network with dynamic parameter prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 30–38 (2016)

15. Kim, J.H., Lee, S.W., Kwak, D., Heo, M.O., Kim, J., Ha, J.W., Zhang, B.T.: Multimodal residual learning for visual QA. In: Advances in Neural Information Processing Systems, pp. 361–369 (2016)
16. Bai, Y., Fu, J., Zhao, T., Mei, T.: Deep attention neural tensor network for visual question answering. In: Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, vol. 11216, p. 20. Springer, Berlin (2018)
17. Peng, L., Yang, Y., Bin, Y., Xie, N., Shen, F., Ji, Y., Xu, X.: Word-to-region attention network for visual question answering. *Multimedia Tools Appl.* 1–16 (2018)
18. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances in Neural Information Processing Systems, pp. 289–297 (2016)
19. Malinowski, M., Doersch, C., Santoro, A., Battaglia, P.: Learning visual question answering by bootstrapping hard attention. In: Computer Vision—ECCV 2018 Lecture Notes in Computer Science, pp. 3–20 (2018)
20. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer, Cham (2014)
21. Tommasi, T., Mallya, A., Plummer, B., Lazebnik, S., Berg, A.C., Berg, T.L.: Combining multiple cues for visual madlibs question answering. *Int. J. Comput. Vis.* 1–23 (2018)
22. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
23. Campello, R.J., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 160–172. Springer, Berlin (2013)
24. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley (2012)
25. Haghigiat, M., Abdel-Mottaleb, M., Alhalabi, W.: Discriminant correlation analysis: real-time feature level fusion for multimodal biometric recognition. *IEEE Trans. Inf. Forensics Secur.* **11**(9), 1984–1996 (2016)
26. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. In: Advances in Neural Information Processing Systems, pp. 1682–1690 (2014)
27. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement (2018). arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)

An Investigation on Indoor Navigation Systems



**J. Akilandeswari, A. Naveenkumar, R. S. Sabeanian, P. Iyyanar,
M. E. Paramasivam, and G. Jothi**

Abstract The development of navigation system for visually impaired people in an indoor environment has been a challenge to researchers for more than a decade. This paper presents a review of the literature on recent techniques and methodologies for indoor navigation system. A detailed review on each phase in the navigation system along with the various techniques used, cost involved, feasibility level along with the performance of path selection algorithms has been analyzed. The paper also presents the demerits of the existing approaches for navigation in indoors.

Keywords Indoor navigation system · Location finding · Path selection · Object detection

J. Akilandeswari (✉) · A. Naveenkumar · R. S. Sabeanian · P. Iyyanar · M. E. Paramasivam ·

G. Jothi

Sona College of Technology (Autonomous), Salem, Tamil Nadu, India

e-mail: akilandeswari@sonatech.ac.in

A. Naveenkumar

e-mail: naveenkumar@sonatech.ac.in

R. S. Sabeanian

e-mail: sabeanian@sonatech.ac.in

P. Iyyanar

e-mail: iyyanar.p@sonatech.ac.in

M. E. Paramasivam

e-mail: sivam@sonatech.ac.in

G. Jothi

e-mail: jothig@sonatech.ac.in

1 Introduction

A survey by the World Health Organization (WHO) shows that approximately 1.3 billion people are affected with visual defects [1]. Incidentally, around 15 million in the above-mentioned population live in India. Researchers have been working for more than a decade to help such visually impaired people to move around without co-human intervention and efforts. The goal of an indoor navigation system is to develop an effective system for navigating visually impaired in a closed environment. The one-stop solution for positioning and navigation of human's with electronic gadgets is to utilize Global Positioning System (GPS). However, this technology does not suit for indoor navigation due to the non-transduction of signals in concrete buildings. Hence, the utilization of GPS technology for indoor environment becomes void. In an indoor environment, the choice of the best and dynamic path has been a challenge for researchers. The presence of smaller object viz (table, chair, wall, partitions, etc.,) in an indoor environment encourages the utilization of multiple sensors along with WiFi signals for any location.

The development of an indoor navigation system contains two phases. In the first phase, the shortest root between the present location and destination is identified and navigates them. In the second phase, objects/obstacles are detected for proper navigation. This paper presents the literature which contributed to the development of such navigation systems in all the two phases or any of the two phases. We propose the taxonomies of methodologies which propose image processing-based solutions to navigate visually challenged/impaired person.

The rest of the paper is structured as follows: Section 2 describes various pathfinding algorithms. Section 3 explains the methods for object detection. The conclusion is presented in Sect. 4.

2 Path Selection Taxonomy

Generally, searching algorithms are categorized into uninformed and informed (heuristic) search based on information needed to reach out the destination. Uninformed search reaches the goal without any additional information. However, the approach computes the exact path in reaching the destination. On the other hand, informed search requires additional information about path to reach the destination.

2.1 *Uninformed Search Algorithm*

Uninformed search algorithm is based on the pre-established rules to reach the destination and is used to obtain the exact path between the source and destination. There are two types of algorithms available.

2.1.1 Dijkstra's Algorithm

Dijkstra's algorithm is one among the classical approaches which computes the path toward the destination, based on the shortest weight of the next node. Xu et al. [2] implemented the indoor optical path planning based on the Dijkstra's approach. Eshu [3] implemented the Dijkstra's algorithm with the help of Global Positioning System (GPS) for planning the traffic in a populated city.

2.1.2 Breadth-First Search Algorithm

Breadth-first search (BFS) algorithm searches in a horizontal fashion to get the shortest path. In this algorithm, the visited nodes are maintained as a binary array, and thereof, the shortest path is calculated. Akram et al. [4] implemented the BFS concept on a wireless network to find the shortest route and build a bridge. Subramanian et al. [5] proposed the path planning method for mobile robot agent on dynamic environment based on the BFS.

2.2 *Informed Search Algorithm*

The informed search algorithm requires additional information such as direction of the destination to plan routes with pre-established rules. This algorithm forecasts the multiple routes between the nodes, and finally, one best route is chosen.

2.2.1 A* Algorithm

A* algorithm is heuristic algorithm to detect the routing on dynamic environment. In this algorithm, the heuristic function should be calculated on each node. Based on the heuristic values, the optimal path is predicted. Martinez-Sala et al. [6] implemented the indoor navigation system for visually impaired people by using the A* algorithm for routing purpose. Zhang et al. [7] proposed the improved A* algorithm to predict a safe path for mobile robot on complex environment.

2.2.2 IDA* Algorithm

Iterative-deepening algorithm (IDA*) is a path search algorithm which identifies the shortest path between the nodes. It is a depth-first search algorithm that inherits the properties of heuristic function in A* algorithm. The main advantage of IDA* algorithm is that it consumes less memory when compared to A* algorithm. Kurkovsky [8] experimented the IDA* concept on the heterogeneous mobile network for optimizing the space. Potts et al. [9] implemented the iterative expansion A* algorithm

based on the IDA*. This method minimizes the redundant node as in case of the depth search algorithm in each iteration.

2.2.3 Jump Point Search Algorithm

Jump point search (JPS) algorithm works with an assumption to pruning the neighbor nodes on horizontal, vertical or diagonally on an uniformed graph. It is an optimized A* algorithm. Aversa et al. [10] introduced inventory-driven path planning based on the concept of JPS.

2.2.4 Orthogonal Jump Point Search Algorithm

Orthogonal jump point search algorithm is advanced version of JPS algorithm. In this algorithm, the pruning operations are carrying on all eight directions to obtain the routing between the nodes [11].

The performance of various path selection algorithms is analyzed using the tool pathfinding [12]. A screenshot of the tool utilized for each of the methods is shown in Fig. 1. In Fig. 1, the starting node is denoted in green color box, destination node is denoted in red color box, obstacles are represented in gray color box and the path is represented in yellow color line.

Table 1 represents the comparison of path selection algorithm based on the experimental setup of pathfinding tool. The experimental results infer that A* algorithm performed better when compared to other path selection algorithms based on computation time and complexity metrics.

3 Object or Obstacle Detection Taxonomy

Object or obstacle detection plays a vital role in navigation system. Object detection is the process of identifying and locating the objects in an image or scene. The literatures show that object detection was achieved through using approaches such as histogram of oriented gradients (HOG) [13], scale-invariant feature transform (SIFT) [14] and speeded-up robust feature (SURF) [15]. However, the existing approaches are found to be not suitable for context-aware environment. With the technological advancements, image detection is done using concepts of neural networks such as CNN, RCNN, Fast RCNN, Faster RCNN and YOLO. The object detection algorithms utilizing neural network approaches detect obstacles, pedestrian and objects.

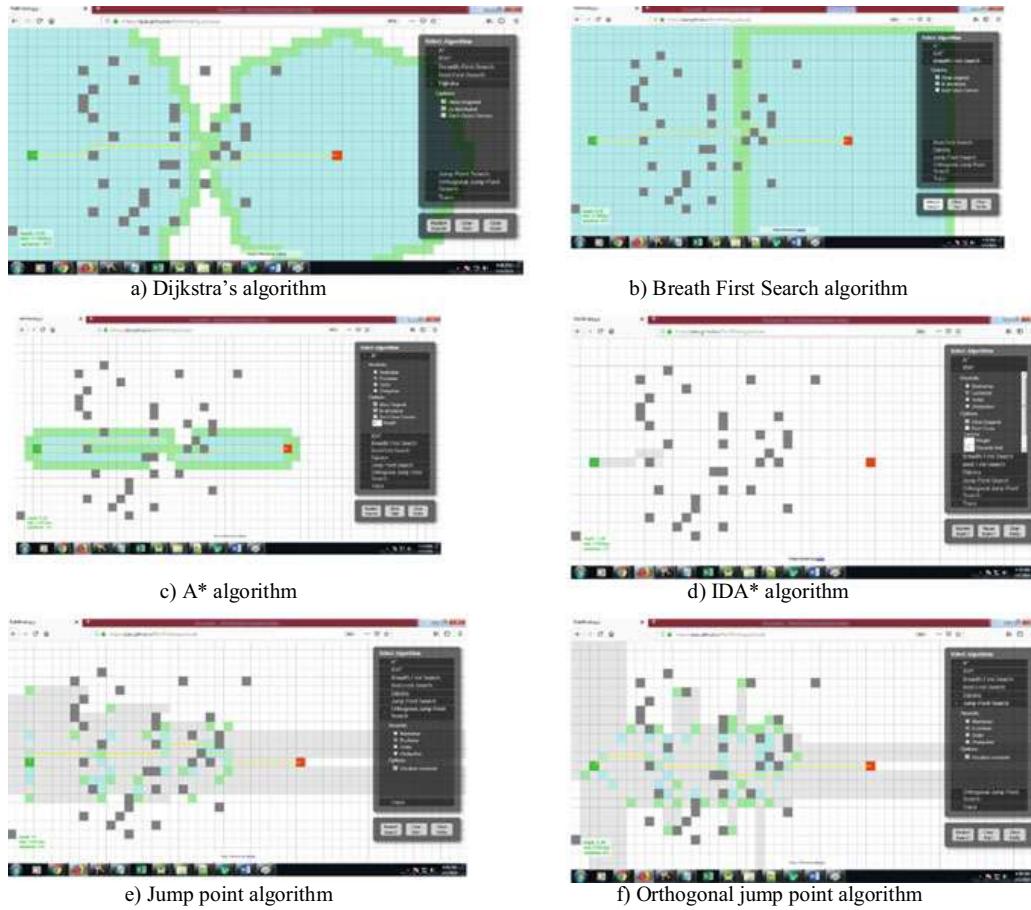


Fig. 1 Output results of various path selection algorithms

Table 1 Performance comparison

S. No.	Path selection algorithm	Length	Time (ms)	Operations
1	Dijkstra's algorithm	31.66	15	1872
2	Breath First Search algorithm	31.66	13	2325
3	A* algorithm	31.66	2	210
4	IDA* algorithm	31.66	Infinite	Infinite
5	Jump point algorithm	31.66	9	805
6	Orthogonal jump point algorithm	34	4	930

3.1 CNN Algorithm

Convolution neural network (CNN) is a feed-forward artificial network which needs a minimal preprocessing compared with classical image classification algorithms. Convolution neural network (CNN) takes an input layer and processes it as to convolutional layer, pooling layers and fully connected layers to produce the output layers.

Ran et al. [16] implemented CNN-based framework to detect the objects in the spherical images. The proposed framework consists of one input layer, one output layer and seven hidden layers (convolutional layers and pooling layers) to detect the objects on images. Himstedt et al. [17] proposed a system to classify the obstacles in the images based on the CNN. The proposed system was implemented in trucks for identifying objects in front of it. Cornacchia et al. [18] developed a method for autonomous obstacle detection and classification using different sensors like patterned light projector and camera. The deep learning algorithm such as CNN is employed to classify the objects.

3.2 RCNN Algorithm

RCNN is short form of region-based convolution neural network which is consists of two steps. The first one is to detect the region from the image for classification and detect the objects. The second step is to extract the CNN features from each region to detect the objects. Girshick et al. [19] proposed the RCNN method based on the simple bounding box concept which is used to select the region from the images, and from that region, CNN features are extracted to detect the object. The method combines the strength of computer vision tools and deep learning algorithms. Cao et al. [20] applied region proposal network (RPN) and CNN with spatial transformer network (STN) concept on airfield pavement for detecting the foreign object debris (FOD).

3.3 Fast RCNN Algorithm

Fast RCNN is an extension approach of RCNN to detect the object. The concept of Fast RCNN is similar to RCNN. But instead of extracting the CNN features for each region, these methods compound it as one CNN and forward it to detect the object and classification [21]. Li et al. [22] developed a framework using scale-aware Fast RCNN method for pedestrian detection in natural scenes. In this approach, several integral subnetworks identify the pedestrians with scales from separate series.

3.4 Faster RCNN Algorithm

Advanced version of Fast RCNN is Faster RCNN. To speedup, the approach constructed a region proposal network from convolutional layers and then applied the Fast RCNN to detect and classify the objects. Agarwal et al. [23] implemented the multiple object tracking (MOT) system to detect the real-time object based on the Faster RCNN and classified the object. Ren et al. [24] investigated the Faster RCNN

algorithm for object detection. The author also analyzed the detection performance of various convolutional architectures in the Fast/Faster RCNN.

3.5 YOLO

YOLO is the short form of You Only Look Once. Redmon et al. [25] proposed YOLO algorithm for pedestrian detection method. Instead of taking the convolution networks or region proposal network to perform regression and training, this model takes the original images to find out the objects. The performance of YOLO is faster than the RCNN family and other traditional image classification models. Zhou et al. [26] designed a system to detect the objects and track the Nao robot using YOLO architecture. The experimental results show that YOLO algorithm efficiently detects the objects and localizes landmarks. The summary of various object/obstacle detection algorithms and its limitations is presented in Table 2.

Table 2 Summarization of various object/obstacle detection algorithms

Author and year	Techniques	Objective	Feasibility level	Demerits
Ran et al. 2017 [16]	CNN	To detect the objects in the spherical images using CNN	Difficult (collecting dataset for training and testing)	High computation complexity
Himstedt et al. 2016 [17]	CNN	To classify the obstacles in the images based on the CNN to automated reach trucks	Moderate (used publically available dataset)	High processing time because each movement has to apply feature extraction
Cornacchia et al. 2018 [18]	CNN, patterned light projector, camera	To detect the objects using the sensors like patterned light field and camera. The CNN model is used to classify the objects	Moderate (portable to carry, easy to collect the data)	It is a binary classification i.e., the object is present or not. It does not predict multi objects
Girshick, R et al. 2014 [19]	RCNN	Objects are segmented and label the objects using RCNN	Difficult (after segmentation, the objects are classified)	Increase the processing time for segment the objects

(continued)

Table 2 (continued)

Author and year	Techniques	Objective	Feasibility level	Demerits
Cao et al. 2018 [20]	CNN RPN STN	Foreign object debris (FOD) was obtained using RCNN	Difficult (high design complexity)	Low response for retrieve the database
Li et al. 2018 [22]	RCNN	Scale-aware Fast RCNN method utilized to detect the pedestrian	Moderate (using pre-trained model and dataset)	This method applied only for pedestrian detection
Agarwal et al. 2017 [23]	Faster RCNN	To develop a MOT system to detect the real-time object based on the Faster RCNN	Moderate (easy to collect the data)	This model gives less efficient results
Ren et al. 2018 [24]	Faster RCNN	Analyzed the convolutional architectures in the Fast/Faster RCNN	Easy (apply the existing model to object detection)	Needs more time for training
Redmon et al. 2016 [25]	Yolo	Yolo algorithm is applied to detect pedestrian	Moderate (Easy to collect the data)	This method applied only for pedestrian detection
Zhou et al. 2018 [26]	Yolo	To detect the objects and track the Nao robot using YOLO algorithm	Moderate (using pre-trained model and modifying the last fully connected layer)	This model failure to detect the small and adjacent objects in an image

4 Conclusion

The development of indoor navigation system for visually impaired people helps to moves in and around the place without any assistant. This paper reviews the state-of-the-art techniques of path selection and object/obstacle detection in the closed environment. The work has also analyzed the performance of few methods based on various metrics. Each of the algorithms that has been presented in this article has performed well at the time proposal and available resources. However, on a current scenario, we find that A* algorithm is an effective in finding the shortest path when compared to other path selection algorithms. Likewise, YOLO algorithm functions efficiently in detecting the objects on a real-time basis.

Acknowledgements The authors would like to thank the Department of Science & Technology, Technology Bhavan, New Mehrauli Road, New Delhi, for funding this work under the Science for Equity Empowerment and Development Division (SEED) scheme (Vide Letter Number SEED//TIDE/202/2016/G, dated: 12/01/2018.).

Declaration We have taken permission from competent authorities to use the images/data as given in the paper. In case of any dispute in the future, we shall be wholly responsible.

References

1. www.who.int visited on 9 Aug 2019
2. Xu, Y., Wen, Z., Zhang, X.: Indoor optimal path planning based on Dijkstra algorithm. In: International Conference on Materials Engineering and Information Technology Applications (MEITA 2015). Atlantis Press (2015)
3. Eshu, G.: Global positioning system using Dijkstra's algorithm traffic planning system. *Int. J. Comput. Trends Technol. (IJCTT)* **35**(5), 231–235 (2016)
4. Akram, V., Dagdeviren, O.: Breadth-first search-based single-phase algorithms for bridge detection in wireless sensor networks. *Sensors* **13**(7), 8786–8813 (2013)
5. Subramanian, M.B., Sudhagar, D.K., RajaRajeswari, G.: Intelligent path planning of mobile robot agent by using breadth first search algorithm. *Int. J. Innov. Res. Sci. Eng. Technol.* **3**, 1951–1955 (2014)
6. Martinez-Sala, A., Losilla, F., Sánchez-Aarnoutse, J., García-Haro, J.: Design, implementation and evaluation of an indoor navigation system for visually impaired people. *Sensors* **15**(12), 32168–32187 (2015)
7. Zhang, H.M., Li, M.L., Yang, L.: Safe path planning of mobile robot based on improved A* algorithm in complex terrains. *Algorithms* **11**(4), 44 1–18 (2018)
8. Kurkovsky, S.: Experimenting with IDA* search algorithm in heterogeneous pervasive environments. *Artif. Intell. Rev.* **29**(3–4), 277–286 (2008)
9. Potts, C.M., Krebsbach, K.D.: Iterative-expansion A. In: Twenty-Fifth International FLAIRS Conference, pp. 1–12 (2012)
10. Aversa, D., Sardina, S., Vassos, S.: Path planning with inventory-driven jump-point-search. In: Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference (2015)
11. Harabor, D.D., Grastien, A.: Improving jump point search. In: Twenty-Fourth International Conference on Automated Planning and Scheduling (2014)
12. Pathfinding.js, <https://qiao.github.io/PathFinding.js/visual>
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection (2005)
14. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. *Comput. Vis. Image Underst. (CVIU)* **110**(3), 346–359 (2008)
15. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, vol. 99, issue 2, pp. 1150–1157 (1999)
16. Ran, L., Zhang, Y., Zhang, Q., Yang, T.: Convolutional neural network-based robot navigation using uncalibrated spherical images. *Sensors* **17**(6), 1341, 1–18 (2017)
17. Himstedt, M., Maehle, E.: Camera-based obstacle classification for automated reach trucks using deep learning. In: Proceedings of ISR 2016: 47th International Symposium on Robotics, pp. 1–6 (2016)
18. Cornacchia, M., Kakillioglu, B., Zheng, Y., Velipasalar, S.: Deep learning-based obstacle detection and classification with portable uncalibrated patterned light. *IEEE Sens. J.* **18**(20), 8416–8425 (2018)
19. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
20. Cao, X., Wang, P., Meng, C., Bai, X., Gong, G., Liu, M., Qi, J.: Region based CNN for foreign object debris detection on airfield pavement. *Sensors* **18**(3), 737 1–18 (2018)
21. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

22. Li, J., Liang, X., Shen, S., Xu, T., Feng, J., Yan, S.: Scale-aware fast R-CNN for pedestrian detection. *IEEE Trans. Multimedia* **20**(4), 985–996 (2018)
23. Agarwal, A., Suryavanshi, S.: Real-Time* Multiple Object Tracking (MOT) for Autonomous Navigation. Technical report (2017)
24. Ren, Y., Zhu, C., Xiao, S.: Object detection based on fast/faster RCNN employing fully convolutional architectures. *Math. Probl. Eng.* (2018)
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
26. Zhou, J., Feng, L., Chellali, R., Zhu, H.: Detecting and tracking objects in HRI: YOLO networks for the NAO “I See You” function. In: 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 479–482 (2018)

Conceptualization and Design of Remotely-Accessible Hardware Interface (RAHI) Laboratory



Shivam Mahesh Potdar , Vanshika Gupta, Pruthviraj Umesh,
and K. V. Gangadharan 

Abstract With the rising popularity of e-learning through means like Massive Open Online Courses (MOOCs), remote-triggered and virtual laboratories, new and innovative technologies for enhancing the learning experience are in demand. E-learning resources for electronics hardware are generally simulation-based, as getting access to high-end hardware is difficult for students due to cost and availability. In this paper, a novel method to create a remotely-accessible, low-cost, modular, and scalable hardware learning platform is proposed and demonstrated through a prototype. Users can interact with the system through a web interface anytime-anywhere and verify results on actual hardware through real-time visual and textual feedback and learn at their pace. The prototype demonstrates a web application hosted on a Linux-PC server interacting with a Raspberry Pi. Student activities are logged in a database for future reference and correction by instructors. The software stack used for the system is free and open-source. The prototype system was launched on a pilot run, gaining positive feedback from students and teachers. Hence, such a system can undergo comprehensive implementation in educational institutions and for the delivery of MOOCs with minimal investment for both laboratory setups and learners.

Keywords Intelligent e-learning systems · E-learning · Smart learning systems · Remote code execution · Electronics education · Remote laboratory · MOOCs · Embedded systems

S. M. Potdar () · V. Gupta · P. Umesh · K. V. Gangadharan
National Institute of Technology Karnataka (NITK), Surathkal, Mangalore, Karnataka, India
e-mail: shivampotdar99@gmail.com

V. Gupta
e-mail: vanshika421@gmail.com

P. Umesh
e-mail: pruthviu@nitk.edu.in

K. V. Gangadharan
e-mail: kvganga@nitk.edu.in

1 Introduction

1.1 *Background and Problem Identification*

With the continuously evolving industries, both theoretical knowledge and practical learning are essential for a modern education system, more so for students undergoing technical courses like Engineering. Electronics, especially Embedded Systems, is one such area where it is difficult to understand the concepts taught in theory, without trying them hands-on.

Using development boards such as Arduino and NodeMCU for introductory electronics courses is a widespread practice [1–3], as they are low-priced and popular. Although they are available in most university laboratories, the on-board features on these boards are minimal. Even if extended by external modules, the boards are limited in terms of memory and computing power.

Platforms such as Raspberry Pi (RPi) are useful in this scenario, as they allow students to build systems with much higher complexity. They are convenient as they have numerous on-board features, programming in Python, and Linux-based operating system. Python is also regarded as the best introductory programming language for students [4]. With the basic knowledge acquired through courses on boards like RPi, students can solve significant problems for the industry and academia in the future.

However, RPi is significantly more expensive than Arduino due to the high-end hardware, which limits its accessibility to students and availability in laboratories. Generally, in online courses based on such boards, students have to buy one themselves, which further adds to the course expenses. Simulation, being an alternative medium does explain concepts but is still limited by the ideal case results and difficulty in reconfiguration. The fact that the experiments are not conducted on real hardware reduces the interest of the students as well.

Although it is advantageous to use such platforms for teaching, cost, and accessibility is a significant issue for both setting up laboratories in institutes and conducting online courses.

1.2 *Literature Review and Proposed Solution*

Attempts to build a remote electronics laboratory on similar lines have been made in the past by various authors [5–8]. In some cases, simulation has been used as a learning medium [6, 7], while some have used hardware as well [5, 8]. In [8], the authors have tried to use simulation as a sandboxing factor to decide whether the HDL code should be executed on hardware, while in [5] using DAQ systems with proprietary GUI is proposed.

The proposed solution in our research adds a real-time visual feedback component along with maintaining the sandboxing, through different means, and hence enhancing the overall learning experience.

To address the above problem, we worked on developing a “Remote Code Testing for Hardware” platform, where multiple devices like a RPi can be paired to a server, students can remotely login to the system, and try running code experiments on hardware (various sensors and actuators) connected to the boards along with real-time visual and textual feedback. The conceptualization and prototype development of this system is explained in this paper.

2 Technical Description

The overall system has three components; user(s), server, and RPi(s), and its goal is to integrate these three components for seamless interaction with each other. The workflow can be summarized, as shown in Fig. 1.

To demonstrate the concept, we built a system with a Linux desktop as the server and one RPi connected to it. A diagrammatic overview of the system is shown in Fig. 2.

2.1 Server

Being a Linux-based computer itself, RPi is capable of hosting a web server directly. However, it is proposed to use an independent web server because it makes future expansion of the system very easy. With only one RPi available since the entered code has to be interfaced with hardware, only one user session can be allowed at a time. The waiting time for subsequent users can be reduced by having multiple units connected to the server and then allocating resources to the users dynamically.

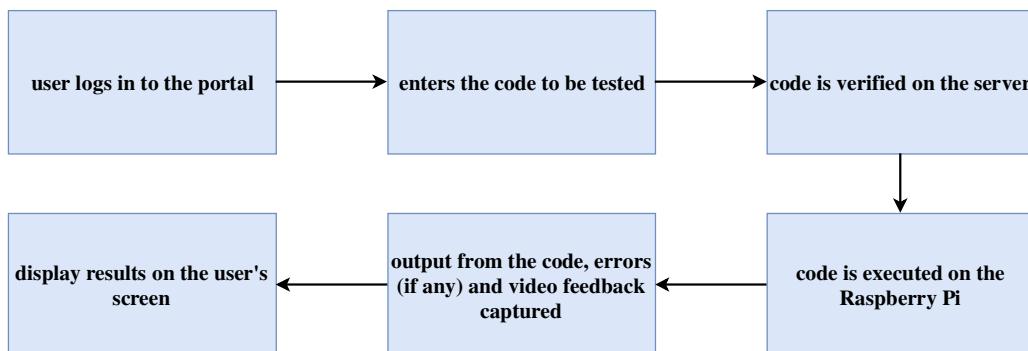


Fig. 1 Workflow chart for the remote code testing system

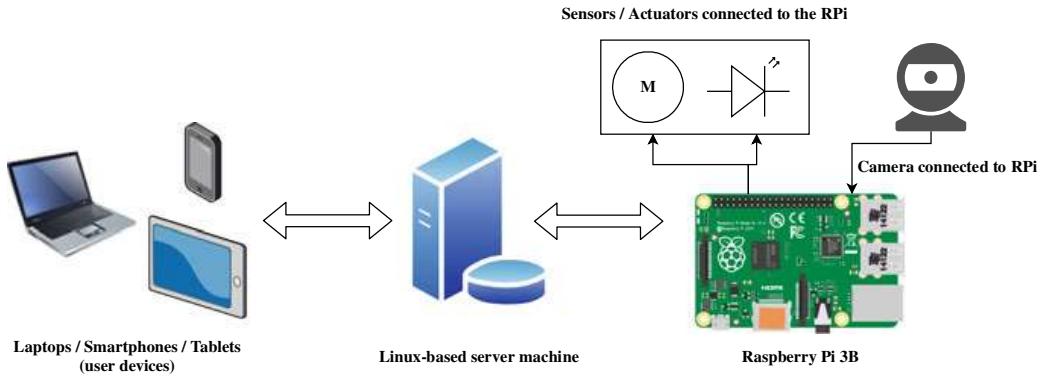


Fig. 2 Diagrammatic overview of the prototype system

Also, the storage capabilities of the RPi are limited, which makes it challenging to store logs of data on the system. In the current setup, an SQLite database is located on the server, which logs the code submitted by users, results obtained, and corresponding video clips. Hence, the right amount of free storage space on the server is required.

2.2 *Raspberry Pi and Hardware*

The RPi in this framework acts as a slave device. The server handles all the user management logic, code sandboxing, and then directs the RPi to execute the code. All the components viz sensors, actuators, etc., can be connected to it.

A USB web-camera or RPi Camera can be connected to the RPi directly for real-time visual feedback. Open-source software Motion [9] is installed on the RPi, which is triggered by the server for recording the video. While active, the stream is directly shown to the user and simultaneously stored on the local storage of the RPi. After the end of the users current session, this video file is transferred to the server, logged into the database, and removed from the RPi to free up storage space. A cronjob continuously runs on the RPi to ensure that if Motion remains triggered for an unexpectedly long time, the process is killed, and storage space is freed up.

It is suggested that the maximum number of components are connected to the RPi to make the system modular. Users can write code for whatever they want to try interfacing. For example, for course assignments, a setup like a smart home with various modules already connected to the RPi can be made. Incrementally students can interface individual modules and have an end goal of making a complete system using all modules.

In the presented prototype, both RPi and the server are connected to the institute intranet through WiFi and LAN, respectively. Hence, no direct connection between the two is required.

2.3 Web Application

A Python-based web application using Django MVC architecture is built to be hosted on the server. It provides features like user management, communication with the hardware, handling logging of user data to the database, etc.

2.3.1 User Management

Django's in-built user management system is made use of, which provides features like permissions, groups, password hashing, pluggable backend, object-level permissions, etc. out of the box.

2.3.2 Code Editor

Online code editors are quite common mainly due to competitive programming platforms and technical forums, which makes code-snippet sharing easy. Open-source Ace Editor [10] is integrated with the web application for code-entry and features like syntax-highlighting and formatting (Fig. 3).

2.3.3 Sandboxing

In the proposed solution, sandboxing is implemented by checking if the user is sending malicious commands such as `rm...`, `sudo...`, `shutdown`, `reboot`, `ifconfig`, etc. to the RPi. The idea here is to not restrict the users to solve a given problem in a predefined way and allowing them to explore multiple approaches with the constraint of not damaging the system.

```

1 import RPi.GPIO as IO          # calling header file which helps us use GPIO's of PI
2 import time                   # calling time to provide delays in program
3 IO.setwarnings(False)         # do not show any warnings
4 x=0                          #integer for storing the duty cycle value
5 IO.setmode (IO.BCM)           #we are programming the GPIO by BCM pin numbers. (PIN35 as'GPIO19')
6 IO.setup(13,IO.OUT)            # initialize GPIO13 as an output.
7 IO.setup(19,IO.IN)             # initialize GPIO19 as an input.
8 IO.setup(26,IO.IN)             # initialize GPIO26 as an input.
9 p = IO.PWM(13,100)            #GPIO13 as PWM output, with 100Hz frequency
10 p.start(0)                   #generate PWM signal with 0% duty cycle
11 while 1:                     #execute loop forever
12     p.ChangeDutyCycle(x)      #change duty cycle for changing the brightness of LED.
13     if(IO.input(26) == False):  #if button1 is pressed
14         if(x<50):
15             x=x+1                #increment x by one if x<50
16             time.sleep(0.2)        #sleep for 200ms
17         if(IO.input(19) == False): #if button2 is pressed
18             if(x>0):
19                 x=x-1              #decrement x by one if x>0
20                 time.sleep(0.2)        #sleep for 200ms

```

Fig. 3 Automatic syntax-highlighting and indentation by Ace editor

2.3.4 Code Execution

If the code successfully passes above checks, a .py executable file is created and transferred to the RPi using Python fabric module and then executed via SSH pipes using Python subprocess module. Output and error from this pipe are collected and displayed back to the user. The video stream is also shown on the page, along with the results (Fig. 4).

2.3.5 Database and Logs

All user-submitted codes and the corresponding videos are logged into the database. This feature is useful where a user would want to know their use history. If the platform is being used for course instruction, the teachers can check user logs to verify and evaluate the experiments carried out by their students (Fig. 5).

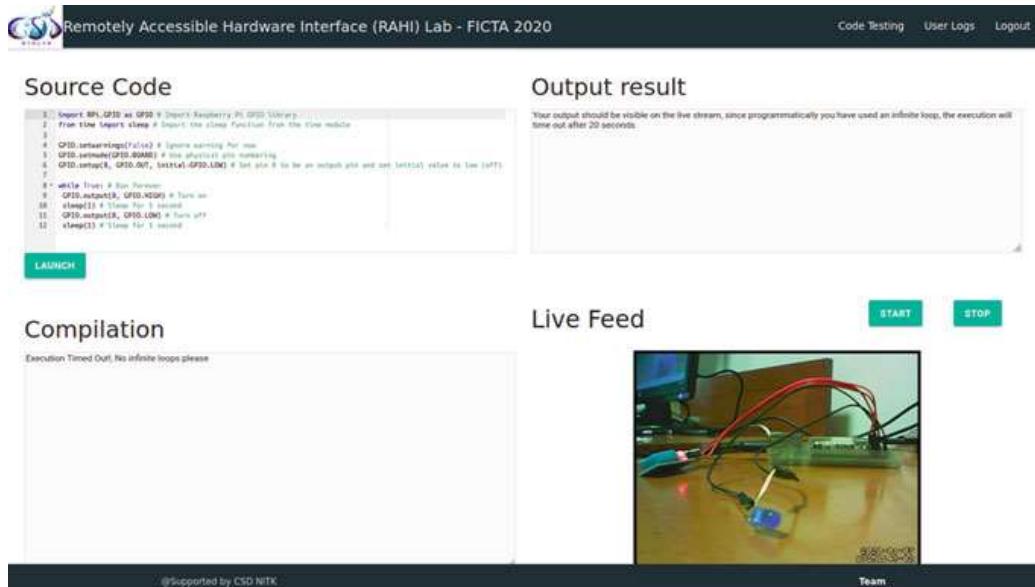


Fig. 4 Output on the user screen showing execution results

Author	Time Saved	Pycode	Author	Post Date	Video
shivam	April 16, 2019, 6:55 p.m.	user_T_shivam/16-04-19_105551.py	shivam	April 16, 2019, 6:55 p.m.	videos/user_T_shivam/shivam-16-04-19_10-54-23_358Pyv6.mp4
			shivam	April 16, 2019, 6:56 p.m.	videos/user_T_shivam/shivam-16-04-19_10-55-08_wocGtpt.mp4

Fig. 5 Logs stored in the database (as viewed by an instructor)

2.3.6 Ensuring Single-User Activity

In the prototype system, only one RPi is connected to the server, hence to ensure that only a single user is using the system at a time, there are few conditions implemented through Django middleware.

- Maximum time allotted per session: If the user exceeds the maximum time as set in the web-app settings, then he is automatically logged out of the system.
- If one user is active (without exceeding timeout), another user should be barred from logging in, either until the first user logs out or their session is timed out. It is also ensured that if the same user tries to log in from a different device (or IP address), their earlier sessions are auto-expired.

3 Discussion

3.1 Testing

The prototype was kept under a month-long pilot run with access open to NITK students through the institute's intranet. In the initial period, few bugs were identified, and code was patched accordingly. Users tried basic experiments like LED control, DC motor control, sensors interfacing, etc.

Feedback was taken from the users, which gave deep insights for the analysis and future scope for improvement. Most of the users indicated that they found the system useful and would like to have it as a part of their electronics courses.

3.2 Analysis

Section 2 describes the technical build-up of the prototype of the proposed solution. Several expected objectives are achieved, and there are some shortcomings as well. Some of the noteworthy points from the analysis of the prototype system are:

Positives

- *Anytime Anywhere Access.* It gives a new kind of motivation for students to try out the experiments according to their convenience and hence, a better learning impact. It follows the 3P principle that is learning at your place, your pace, and your period.
- *Real-Time Video Feedback.* This is one of the most prominent features of the proposed system. It provides a convenient learning experience to the user, which is closer to being physically present in the laboratory as compared to simulations or demonstrations.

- *Investment for Setup.* The only initial investment is in a PC and a few RPis. Once set up, virtually there is no limit on the number of users with proper scheduling. Whereas in a traditional laboratory, generally, at least a group of 3–4 students gets only one board.
- *Safety of High-End Hardware.* Laboratories generally tend to provide expensive hardware only to senior students as there is a risk of the components getting damaged. Here, the system is safe from such problems; even school kids can learn through the platform.
- *Use of Independent Server.* It makes the future expansion to multiple devices and complex resource allocation easy.
- *Hardware connections.* Making connections of prototype boards consumes much time in a laboratory, here it is served ready-made, focus being on algorithmic and interfacing part.

Negatives

- *Latency.* Live-streaming of video consumes a large amount of data. Even at a resolution of 640,480, the stream starts lagging with slow speed connections. This part is partially improved by directly sending the stream from RPi to the user without the server in the path, and can be improved further.
- *Testing is Not Automated.* The instructor has to check the submission manually, which is useful in the sense that students can be given personal feedback but increases the efforts for instructor compared to automated testing. It is challenging to automate directly since the actual outcome on the hardware is more important than code debugging outputs.
- *No Objective-Based Tasks.* This feature is an advantage in the sense that it does not suppress creativity and allows students to explore beyond the assigned task. Nonetheless, since there are no fixed checking parameters, instructors would have to take extra effort.
- *Plagiarism.* In the current setting, an automated plagiarism check is missing, which can affect students performance.

4 Future Scope

The system described in detail in this paper is a prototype of the proposed concept. It can surely be extended, and additional features can be added, such as:

- *Better Resource Allocation.* Ability to handle multiple hardware devices and scheduling, with slot booking (analogous to that in [5]). Also, in the current setup, users are logged out after a fixed time. Dynamic inactivity checks, the maximum number of test runs allowed, etc. can be implemented.
- *Reducing Latency.* The video feed is served to the local network directly by Motion using multipart JPEG. Faster algorithms for streaming can be implemented.

- *Automated Checking.* Assignments with specific objectives may be automatically checked, similar to what is generally done for programming contests.
- *Extension.* To other kinds of hardware devices like FPGAs, as even they are quite expensive for broad availability.
- *More Robust Sandboxing.* The current system filters code based on specific key-words; it might be bypassed.
- *Analysis of the Video Feed.* Since a large amount of video feed from the experiments is being stored in the server database, it can potentially work as a database for artificial-intelligence-based analysis and automated checking of assignments.
- *Collaborative Work.* With advanced web-development protocols like WebSockets, collaborative work in real-time can be achieved.

5 Conclusion

From the analysis and discussion, it can be concluded that the project has excellent potential for use in two main target areas: Institute laboratories for undergraduate as well as school students and MOOC content.

The presented prototype can be made more robust by implementing the points in Sect. 4 and implemented anywhere quickly with a one-time investment, which can serve thousands of students.

It would also be particularly useful in low-income regions where research infrastructure is difficult to access.

Acknowledgements The authors acknowledge the contribution of Centre for System Design (CSD): A Centre of Excellence at NITK Surathkal pertaining to support in the form of technical equipment and experimental facility. The support received from the members of SOLVE: The Virtual Lab @ NITK Surathkal (csd.nitk.ac.in) is deeply appreciated.

References

1. Galadima, A.A.: Arduino as a learning tool. In: Proceedings of the 11th International Conference on Electronics, Computer and Computation, ICECCO 2014 (2014)
2. Ciencias, A.D.E.L.A.S., En, T.E.I., Alberto, M.C.: Arduino as a tool for the improvement of the teaching. Quid (2015)
3. Cuartielles, D.: Opensource Hardware and Education (2015)
4. Guo, P.: Why Python is a Great Language for Teaching Beginners in Introductory Programming Classes. <http://pgbovine.net/python-teaching.htm>
5. Macas, M.E., Mndez, I.: eLab—Remote electronics lab in real time. In: Proceedings—Frontiers in Education Conference, FIE (2007)
6. Daz, G., Loro, F.G., Castro, M., Tawfik, M., Sancristobal, E., Monteso, S.: Remote electronics lab within a MOOC: design and preliminary results. In: Proceedings—2013 2nd Experiment@ International Conference, exp.at 2013 (2013)

7. Kay, J.S., McKlin, T.: The challenges of using a MOOC to introduce absolute beginners to programming on specialized hardware. In: L@S 2014—Proceedings of the 1st ACM Conference on Learning at Scale (2014)
8. Chen, Y., Quan, C. Bin, Gao, Y.: Programming online judge system. In: Proceedings—2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016 (2017)
9. GitHub—Motion-Project/motion: Motion, a software motion detector. <https://github.com/Motion-Project/motion>
10. GitHub—ajaxorg/ace: Ace (Ajax.org Cloud9 Editor). <https://github.com/ajaxorg/ace>

A Non-invasive approach for Driver Drowsiness Detection using Convolutional Neural Networks



K. K. Sreelakshmi and J. Jennifer Ranjani

Abstract Driver drowsiness has been observed as one of the most common causes for road accidents, producing nearly 40% of death and casualties. When a driver falls asleep, he starts losing control and is unable to take reflex action to avoid the accident or to reduce its impact. This necessitates the need for developing a mechanism that provides timely alerts to the driver when he is drowsy. In this paper, an efficient and non-intrusive algorithm that uses a deep convolutional neural network to analyze yawn behavior is proposed. The proposed technique is built by modifying the VGG16 architecture to include batch normalization, ReLu activation for the intermediate layers and sigmoid activation after the final dense layer. The performance of the proposed approach is verified on the YawDD dataset and is compared against VGG16, VGG19, MobileNet, and AlexNet. Experimental results show that the proposed approach outperforms the other networks in terms of accuracy.

Keywords Deep learning · Convolutional neural networks · Drowsiness detection · Transfer learning · VGG

1 Introduction

Traffic reports highlight that majority of the road accidents are caused by drowsy drivers and the numbers are increasing year by year. According to a survey conducted by the Central Road Research Institute on the 330km Agra–Lucknow expressway, it is observed that almost 40% of the accidents were due to driver dozing off while driving. National Highway Traffic Safety Administration (NHTSA) revealed that every year, approximately 100,000 reported crashes were caused due to drowsy

K. K. Sreelakshmi · J. J. Ranjani (✉)
Department of Computer Science and Information Systems, Birla Institute
of Technology and Science, Pilani, Rajasthan333031, India
e-mail: j.jenniferranjanii@yahoo.co.in

K. K. Sreelakshmi
e-mail: sreelakshmikk2005@gmail.com

drivers. These accidents have caused fatality figures to peak up to an astounding 1550 deaths per year with reported injuries border-lining 71,000. A study conducted by the AAA Foundation for Traffic Safety divulged that 328,000 crashes per year were caused as a result of drowsy driving. The same study also highlighted that 109,000 of the total drowsy driving mishaps resulted in the drivers suffering grievous injury, while 6400 of them culminated in fatalities. The researchers opined that in reality, the instances of fatalities caused due to drowsy driving was well above 350% of what was actually being reported. Beyond the human mortality factor, drowsy driving incidents entail an economic setback as well. Excluding property damage, NHTSA study also reports an astounding \$109 billion loss to society caused by drowsy driving accidents. A driver mostly fails to realize that he or she is fatigued since the onset of drowsiness is hard to identify. This necessitates the need to research the state-of-the-art solutions in this domain for developing robust solutions that can continuously track the drivers to assess their focus levels while driving and to provide a timely alert to avoid such casualties.

2 Literature Survey

A lot of intrusive and non-intrusive methods have been previously suggested to design the alert system. Physiological feature-based approaches like brain wave, heart rate, and body temperature are intrusive since external devices are to be installed on the driver's body. Therefore, non-invasive visible indicators such as frequent blinking of eyes, rubbing eyes, eyebrow shape, eyelid movement, jaw dropped, repeated yawning, tilt of the head, and frequent movement of the head are mostly adopted to distract from drowsiness. In general, algorithms for driver drowsiness detection analyze one or more combinations of these features. The drivers' drowsiness condition is divided into two categories: normal (alert or awake) and drowsy. Based on these parameters, an automatic drowsiness detection system is expected to yield accurate results even when driver has accessories such as glasses, wigs, hats, or caps.

Some of the previous invasive approaches [1, 2] use electroencephalography (EEG)-based fatigue detection. These systems provide classification accuracies between 84% and 99%. In [1], machine learning model built using Bayesian-Copula discriminant classifier is proposed. It extracts features such as ratio, amplitude, symmetry, and extension from EEG signals and uses a Bayesian-Copula function to classify the samples into drowsy and normal state. The discriminant classifier utilizes the kernel density estimation and Copula theory to build the class conditional density function. In [2], a convolutional neural network that utilizes multichannel EEG signals to extract both spatial and temporal features to monitor the chauffeur's drowsiness state is proposed. The temporal dependencies in EEG are extracted using the core block, and spatial features are fused using dense layers. This model outperformed the conventional machine learning algorithms with an accuracy of 97.37%. However, fatigue detection using EEG is difficult due to low signal-to-noise ratio,

even though it could reflect the mental and pathological states of a human body with rich information and temporal resolution.

Sometimes a combination of intrusive and non-intrusive approach is used to achieve the desired accuracy levels. In [3], a Bayesian-based fusion approach is developed to assess the drowsiness level by classifying the heart rate and facial expressions using local binary patterns. An accuracy level of 76% was achieved which is greater than the accuracy levels achieved when the features were taken separately. In [4], conclusive knowledge provided by the brain-machine interfaces (BMIs) is used for improving driving experience in intelligent vehicles. BMI monitors the chauffeur's cognitive states for potential to correct errors, readiness for movement, and apprehending events. The BMI interface classifies the driver's attention level by extracting features from the EEG signals and by utilizing linear discriminant analysis.

In [5], a prediction approach on lane-keeping and lane-departure behavior is proposed. The Gaussian mixture model and hidden Markov model are combined together to predict the path of the vehicle coming ahead and to decide whether the driver will comply with the lane-departure or correction behavior. The prediction algorithm could warn the drivers according to the predicted trajectory of the forthcoming vehicles. The average false alarm rate of this system is reduced up to 3.13%. However, lane-departure behavior is largely influenced by factors like road curvature and driving behavior which varies from driver to driver.

The yawn rate and its duration are the commonly used metrics to analyze the yawning behavior of the driver and to determine his drowsiness level. One of the most common approaches is to locate the face and detect the mouth regions [6–9]. Several algorithms are proposed to localize the face and/or mouth regions [10–14]. Some of the commonly used approaches for face detection include genetic algorithm with AdaBoost [10], log-polar signatures using static supervised classification [11], color-based feature [12], etc. In [13], AdaBoost and multinomial ridge regression are used to analyze different facial movements such as yawning and blinking. In [15], a two-stage yawn detection algorithm is proposed. In the initial phase, Viola Jones' face detection algorithm [14] is modified to detect the face and the mouth regions. In the second phase, yawning is detected by determining the rate and amount of changes in the mouth region using back propagation.

In the paper, the drowsiness symptoms from the drivers' face are observed by analyzing the chauffeurs' face behavior particularly the yawn pattern. Yawning features are learnt using deep learning techniques such as convolutional neural networks (CNN). Because deep learning techniques have the ability to extract features, analyze and understand the data automatically, whereas, performance of the machine learning algorithms are dependent on the efficiency of manually crafted feature extraction techniques. The rest of the content in the paper is cataloged as follows: Sect. 3 introduces the proposed system and its benefits over the previous approaches. Section 4 describes the implementation setup and analyzes the experimental results, and Sect. 5 presents a summary and future directions.

3 Proposed System

The proposed architecture shown in Fig. 1 aims to train CNN to recognize the yawning pattern for detecting the drowsiness state of the driver. Convolutional neural networks (CNNs) have been used for a long time, mainly in image processing applications such as face detection and classification. CNNs are proved to achieve more accuracy than the conventional machine learning approaches such as support vector machines and HAAR classifiers. This provides the rationale behind using CNN to predict the level of drowsiness. CNNs have the ability to hierarchically extract high-level features using a number of interconnected multilayer neural networks. Convolutional and pooling layers are the two basic layers in a conventional CNN framework. Filtering function in the convolutional layer is used to perform spatial feature extraction from the images. Initial convolutional layers extract local patterns like corners and edges, and high-level image structures are extracted by the final convolutional layers. These characteristics make the CNNs ideal for learning the spatial hierarchical patterns. Convolutional layers are basically defined using patch size, and number of filters used for obtaining the feature map. When compared to fully connected layer, CNNs have reduced number of parameters as all the neurons assigned for a specific feature map have the same weight, bias, etc.

The proposed CNN architecture has been derived by modifying the VGG [16] architecture. VGG networks have the ability to learn complex features using its increased network depth achieved using small convolutional layers followed by multiple nonlinear layers. VGG16 architecture consists of 13 convolutional, 3 fully connected layers, and a softmax optimizer as a final layer. Deep networks often yield poor performance when the model is unable to find the minimum value of a loss function. Batch normalization can be utilized to standardize the data as it provides a mechanism to feedforward the input and to compute the gradient via a backward pass using the parameters and its input. In the proposed architecture, the convolutional layers are followed by batch normalization. Instead of normalizing per activation, all the activations contained in a mini-batch are normalized.

Rectified linear unit (ReLU), which is a non-linear activation function, is applied on the intermediate layers [17]. The training phase can be accelerated using the ReLU activation function compared to the gradient descent-based conventional functions. Pooling layers are utilized to avoid similar feature maps obtained from common pixels in the filter window. Max pooling layer decreases feature variance using maximization and it generalizes the output of the convolutional layers to present abstract features to the higher layers. Dropout layers are incorporated to avoid problems due to over-fitting. Sigmoid optimization function is utilized in the final dense layer as it is ideal when the probability is predicted as an output. The sigmoid function is monotonic and differentiable.

Figure 1 includes image processing operations like RGB to grayscale conversion and data augmentation [18]. Color has little impact on the image descriptors, and conversion to grayscale reduces the computational complexity of the deep network. Data augmentation is incorporated to generate more training samples by applying

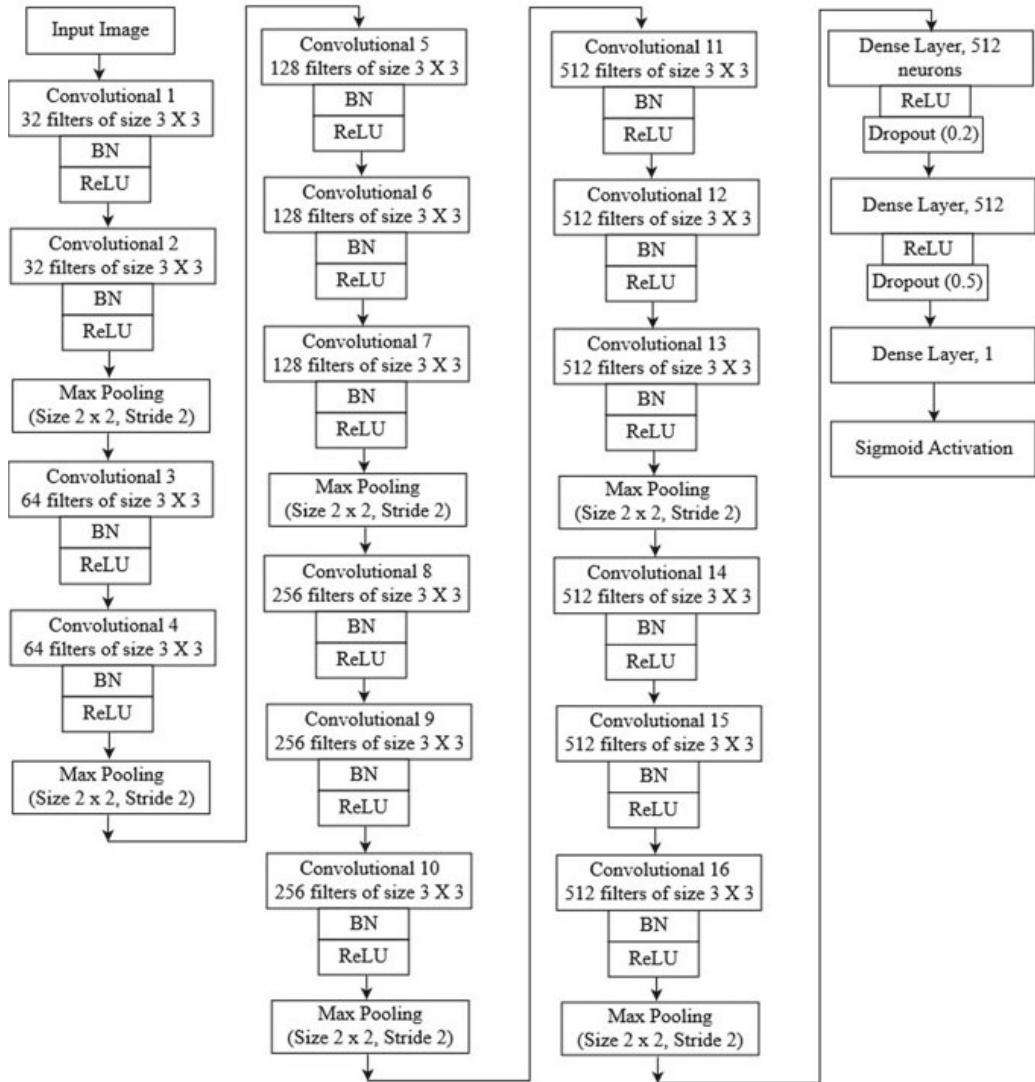


Fig. 1 Yawn detection model using deepCNN

transformations like reflection, shearing, and scaling on the existing training samples. Data augmentation provides better generalization as the model never sees the same image twice.

Algorithm 1: Implementation of the Proposed Work

- 1 Preprocess the training and validation dataset.
 - 2 Define the CNN model by adding layers using the architecture shown in Fig. 1.
 - 3 Use the SGD optimizer and set the hyper-parameters.
 - 4 Compile the model.
 - 5 **foreach** epoch **do**
 - 6 | Fit the model on the training set.
 - 7 | Validate using binary cross-entropy loss and accuracy metric.
-

4 Results and Discussions

Training a CNN requires a large number of positive and negative examples given as input. The dataset for training the CNN has been carefully designed to include almost all varieties of yawn patterns as positive examples. It has been ensured that the data samples are randomized and shuffled. The proposed deep CNN then learns to distinguish features between normal and drowsy state from the examples given. The trained model can then be used to evaluate the drowsy state of the driver and alert him/her.

The YawDD dataset [19] used for training the proposed model comprises of: (a) 29 videos captured from dashboard side and (b) 322 videos captured from the mirror side. Each category consists of videos of women and men from different ethnicities with accessories such as glasses, mustache, beard, and wigs in normal, talking or yawning conditions. The video is captured at 30 fps, where each frame is 24-bit true color with 640×480 pixels. In [20], the comparison between normal and abnormal driving conditions in YawDD dataset is described.

Keras deep learning library in Python is used to implement the proposed CNN model. The image frames are first extracted from the videos and are converted to grayscale. It is scaled down to get the required images of dimension 224×224 . The images are well shuffled to randomize the samples in each batch. Subsequently, the dataset is divided such that the training and test set contain 60% and 40% of the samples, respectively. The model is then compiled, and the hyper-parameters are set. Stochastic gradient descent is the optimizer incorporated in the design. The model is tested by shuffling the training samples for every epoch. The accuracy and loss obtained for various epochs are depicted in Fig. 2a,b, respectively. Validation accuracy and loss are obtained by testing the model using unseen images from the validation set. The performance of the proposed model is analyzed by varying the number of epochs and learning rates. From Fig. 2b, it can be verified that the validation loss settles down at 0.4. The optimal hyper-parameters thus obtained are 0.001 and 20 for the learning rate and number of epochs, respectively. The proposed model achieves an accuracy of 86.85% on the training set with 11323 samples and 86.84% on the validation set with 7551 samples using the optimal hyper-parameters.

The proposed model outperforms the method suggested in [15] by 22% in detecting yawn patterns. It is to be noted that their solution was centered around the image processing perspective, which involves using the Viola–Jones algorithm to detect the face and the mouth region. Then, the yawn rate is counted and the driver is alerted if the yawn rate goes beyond a certain threshold. The technique proposed in [21] uses a 3D-DCNN to extract spatial and temporal features to analyze yawn patterns, head movement, and eye condition. They have compared their model against different pre-trained convolutional models such as VGG FaceNet [16], AlexNet [18], GoogLeNet [22], and FlowImageNet [23] on the NTHU dataset [24]. The accuracy achieved by their model is 76%.

The performance of our proposed model is compared with several pretrained models such as VGG [16], MobileNet [25], and AlexNet [18]. We have used transfer

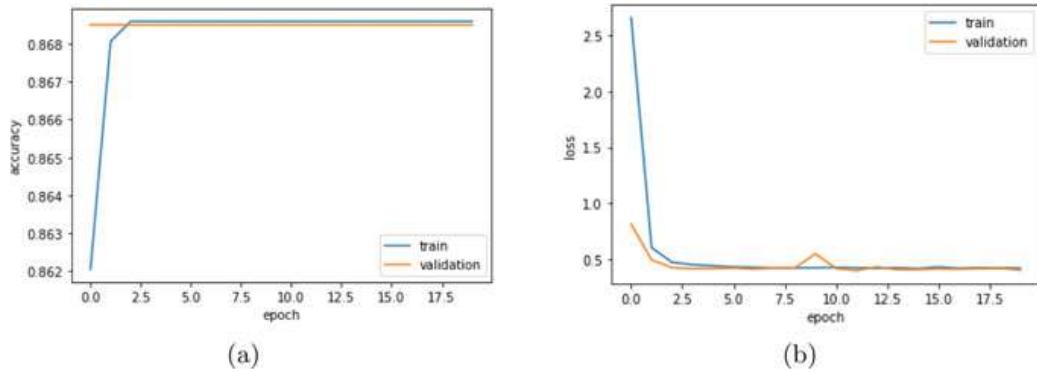


Fig. 2 **a** Training accuracy versus validation accuracy and **b** training loss versus validation loss

Table 1 Comparison with pretrained models

Model	VGG16	MobileNet	AlexNet	VGG19	Proposed model
Accuracy	73.33	69.34	56.54	79.11	86.85

learning to facilitate the pretrained models to train on the YawDD dataset. The results obtained are tabulated in Table 1. We found that the proposed model can perform yawn detection with better accuracy than other standard pretrained models. The improvement in performance has been attributed to the batch normalization and finely tuned hyper-parameters. The time complexity of the proposed system is observed to be 12% and 26% faster when compared to the VGG16 and VGG19 models, respectively.

5 Conclusion and Future Work

In this paper, the potential use of yawn patterns for detecting the drowsiness level of the driver is analyzed. The proposed system uses CNNs as the deep learning approach for training and classification of the sample images into the drowsy or normal state. The proposed CNN model is derived from the existing VGG16 model. Additionally, it includes batch normalization after the convolutional layers to standardize the data. The sigmoid activation function is applied after the final dense layer as drowsiness detection is treated as a binary classification problem. Since yawn patterns are used as the metric to detect drowsiness, the system outperforms even when the driver wears accessories such as spectacles, caps, hats, and wigs. The proposed model is fast enough to determine the onset of drowsiness well in advance that the driver could be alerted immediately. Results prove that our model is able to achieve better accuracy than prior networks such as VGG, MobileNet, and AlexNet.

In this work, only yawning patterns are utilized to detect the onset of drowsiness. In the future, a combination of parameters can be collected and analyzed to arrive at a more certain conclusion. To be specific, temporal and spatial features could be incorporated to analyze the test subject over a period of time and to predict the state of the driver. This could be achieved by using long short-term memory or 3D-CNN. However, careful collection and preparation of dataset samples would be a major challenge.

References

1. Qian, D., Wang, B., Qing, X., Zhang, T., Zhang, Y., Wang, X., Nakamura, M.: Drowsiness detection by Bayesian-Copula discriminant classifier based on EEG signals during daytime short nap. *IEEE Trans. Biomed. Eng.* **64**(4), 743–754 (2017)
2. Gao, Z., Wang, X., Yang, Y., Mu, C., Cai, Q., Dang, W., Zuo, S.: EEG-based SpatioTemporal convolutional neural network for driver fatigue evaluation. *IEEE Trans. Neural Netw. Learn. Syst.* (early access) (2019)
3. Monkaresi, H., Bosch, N., Calvo, R.A., DMello, S.K.: Automated detection of engagement using video-based estimation of facial expressions and heart rate. *IEEE Trans. Affect. Comput.* **8**(1), 15–28 (2017)
4. Chavarriaga, R., Uscumlic, M., Zhang, H., Khaliliardali, Z., Aydarkhanov, R., Saeedi, S., Gheorghe, L., Millan, J.D.R.: Decoding neural correlates of cognitive states to enhance driving experience. *IEEE Trans. Emerg. Top. Comput. Intell.* **2**(4), 288–297 (2018)
5. Wang, W., Zhao, D., Han, W., Xi, J.: A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Trans. Veh. Technol.* **67**(10), 9145–9157 (2018)
6. Soldera, J., Schu, G., Schardosim, L.R., Beltró, E.T.: Facial biometrics and applications. *IEEE Instrum. Meas. Mag.* **20**(2), 4–10 (2017)
7. Yuen, K., Trivedi, M.M.: An occluded stacked hourglass approach to facial landmark localization and occlusion estimation. *IEEE Trans. Intell. Veh.* **2**(4), 321–331 (2017)
8. Kar, A., Corcoran, P.: A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access* **5**, 16495–16519 (2017)
9. Jeong, M., Ko, B.C., Kwak, S., Nam, J.-Y.: Driver facial landmark detection in real driving situations. *IEEE Trans. Circ. Syst. Video Technol.* **28**(10), 2753–2767 (2018)
10. Yang, M., Crenshaw, J., Augustine, B., Mareachen, R., Wu, Y.: AdaBoost-based face detection for embedded systems. *Comput. Vis. Image Underst.* **114**(11), 1116–1125 (2010)
11. Bouvier, C., Benoit, A., Caplier, A., Coulon, P.-Y.: Open or closed mouth state detection: static supervised classification based on log-polar signature. In: Advanced Concepts for Intelligent Vision Systems, vol. 5259, pp. 1093–1102. Springer, Heidelberg, Germany (2008)
12. Minotto, V.P., Lopes, C.B.O., Scharcanski, J., Jung, C.R., Lee, B.: Audio-visual voice activity detection based on microphone arrays and color information. *IEEE J. Sel. Top. Sig. Process.* **7**(1), 147–156 (2013)
13. Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., Movellan, J.: Drowsy driver detection through facial movement analysis. In: Proceedings of the ICCV Workshop on Human Computer Interaction, pp. 6–18 (2007)
14. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)
15. Omidyeganeh, M., Shirmohammadi, S., Abtahi, S., Khurshid, A., Farhan, M., Scharcanski, J., Hariri, B., Laroche, D., Martel, L.: Yawning detection using embedded smart cameras. *IEEE Trans. Instrum. Meas.* **65**(3), 570–582 (2016)
16. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv (2014). [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)

17. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 807–814 (2010)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
19. Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., Hariri, B.: YawDD: a yawning detection dataset. In: Proceedings of the ACM Multimedia Systems, pp. 24–28 (2014)
20. Chiou, C.-Y., Wang, W.-C., Lu, S.-C., Huang, C.-R., Chung, P.-C., Lai, Y.-Y.: Driver monitoring using sparse representation with part-based temporal face descriptors. *IEEE Trans. Intell. Transp. Syst.* (2019) (early access)
21. Yu, J., Park, S., Lee, S., Jeon, M.: Driver drowsiness detection using condition-adaptive representation learning framework. *IEEE Trans. Intell. Transp. Syst.* (2019) (early access)
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
23. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634 (2015)
24. Weng, C.H., Lai, Y.H., Lai, S.H.: Driver drowsiness detection via a hierarchical temporal deep belief network. In: *Proceedings of the Asian Conference on Computer Vision*, pp. 117– 33. Springer (2016)
25. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv (2017). [online] Available: <https://arxiv.org/abs/1704.04861>

Disaster Severity Analysis from Micro-Blog Texts Using Deep-NN



Ramesh Wadawadagi and Veerappa Pagi

Abstract The current decade has witnessed a significant amount of research in the field of sentiment analysis (SA). Several applications have emerged to evidence the necessity of research in this area. On the contrary, the size of micro-blogs content is overgrowing and likely to increase even faster shortly. Social media applications have become part and parcel of our daily lives, as they urge the public to express their opinions and share information around the world. Especially during disasters, people are likely to utilize social media to communicate their hindrances. In this article, we investigate the severity of disaster events from micro-blog messages posted by people during natural calamities and emergencies using deep learning techniques. In particular, the work employs a joint model that combines the features of convolutional neural networks (CNN) with recurrent neural networks (RNN), taking account of the coarse-grained local features generated via CNN and long-range dependencies learned through RNN for analysis of small text messages. Furthermore, the proposed model is evaluated for both binary and fine-grained analyses tested over two different datasets. The accuracy of 87% is observed for binary classification and up to 65% for a three-class problem. The intended work finds usefulness in many instants of disaster relief and crisis management.

Keywords Deep learning · Sentiment analysis · Convolutional neural networks · Recurrent neural networks

1 Introduction

Recently, micro-blog sites have emerged as leading mass media platform, as users are authorized to work collectively and publish their content [1]. Consequently, a large volumetric and semantically rich information is being generated and accumu-

R. Wadawadagi (✉) · V. Pagi
Basaveshwar Engineering College, Bagalkot 587102, India
e-mail: rswlib@yahoo.co.in

V. Pagi
e-mail: veereshpagi@gmail.com

lated every day in terms of tweets, posts, blogs, news, commentaries, reviews, etc. Investigating hidden but potentially useful patterns from a huge collection of micro-blogs is beneficial in several sentiment analysis (SA) tasks [2]. The problem of SA deals with the collection of data from micro-blog sites and automatically detecting whether a text holds sentiment or opinionative content and further determines the opinion polarity [3]. However, identifying sentiments in natural language poses many intellectual challenges, as they are composed of incomplete, chaotic, and unstructured sentences, erratic phrases, ungrammatical expressions, and non-lexical terms. Moreover, it is hard to predict correlations among opinion sentences due to a wide range of linguistic issues and forces the task of SA still more challenging [4]. Hence, it is necessary to develop a real-time SA system to address many challenges and to process a vast amount of sentiment data in moderately sufficient time. As pointed out earlier, micro-blog sites have become an alternative news media to conventional media channels. Social media sites have ranked with the forth prominent source of information necessary to access during emergencies [5]. More specifically, the people utilize social media for reporting activities linked to crises such as warning others to stay in safe areas or requesting for the disaster relief donations [6]. In India, Chennai rains during 2015 and Kerala rains during 2018 are few examples that reveal the influence of social media in flood management. There are plenty of other examples that can be quoted to explain how social media helped in disaster management and emergencies, such as Indonesia's earthquakes, tsunamis in Japan, Hurricane in the USA, and recent Nipah virus in Kerala. People who have been directly or indirectly affected by the crises have utilized social media to keep the status informed, trace the family members, or seek help from concerned authorities. Nevertheless, the analysis of social media posts related to disaster could assist the officials to identify the messages that carry people's feelings, anxiety, and conditions.

The contemporary techniques for SA have not gone far beyond the bag-of-words (BOW) model that ignores semantic knowledge and often operating at a document level [7]. Influenced by the recent success stories of deep neural networks (DNN), this work is an attempt toward building a fine-grained sentence-level classification model for micro-blog contents. The DNN is primarily composed of several hidden layers of nonlinear information processing devised to obtain higher-level hierarchies by the combination of lower-level features, to obtain each subsequent layer a more general meaning [8]. Many DNNs have been proposed in the literature; however, convolutional neural networks (CNN) and recurrent neural networks (RNN) have attained notable success in many natural language processing (NLP), speech recognition, and pattern recognition problems. In contrast, semantic vector-spaces or word-embeddings are successfully utilized for capturing fine-grained semantic regularities in natural language texts [9, 10]. This paper presents a joint model that blends the features of CNN and RNN. Here, the local features extracted from short sentences using CNN are utilized in RNN to study the long-term dependencies. Hence, a bottom-up and end-to-end model is designed to represent the sentences. The model generates feature maps through initializing the CNN with different weight matrices (word-embeddings) and variable-length windows. After iterative convolution and pooling operations, the feature maps are encoded and passed to RNN. The RNN capable of

learning long-term dependencies is utilized in the model as a sentence-level representation method. These representations are further processed with a fully connected network with a softmax layer for a classification task. The intended model is assessed against two distinct disaster-related datasets collected from Twitter. The remainder of the paper is structured according to the following sections. Section 2 presents literature on SA of micro-blog messages related to disaster events. Section 3 gives background knowledge of the proposed joint model C-RNN and its mathematical description. In Sect. 4, the performance of the model is estimated for both binary and fine-grained classification tasks on two different datasets. Finally, Sect. 5 gives a summary of our research contributions, concluding notes, and later improvements.

2 Related Works

The study on SA has continuously evolved from traditional BOW models to semantics-preserving compositions based on machine learning (ML), NLP, and deep learning (DL). The following section presents a literature study on contemporary techniques for disaster situation analysis in social media content. For instance, Koustav et al. in [11] introduced a framework to extract the situational information from Twitter data and to summarize the tweet information using support vector machine (SVM). The model employs both low-level lexical and grammatical features to discriminate the situational and non-situational information present in the tweet data. The data collected for analysis is first pre-processed for dealing with classification effectively. The features set include frequency of different unigrams and bigrams, POS-tags, the rate of strongly subjective words, and the occurrence of appropriate pronouns. Finally, the SVM classifier fused with RBF kernel is used for the classification of tweets into situational and non-situational. The tweets classified as situational tweets are further employed to produce a summary of tweets using integer linear programming (ILP). The model achieves excellent performance when contrasted with modern tweet summarization techniques. Similarly, the classification of disaster-related tweets using annotation schema is discussed [12]. The procedure follows handcrafted annotations with fine-grained labels created in an iterative process scrutinized by domain experts and social scientists. During the process, different categories are identified and annotated with the following labels: sentiment, action, preparation, reporting, information, and movement. For classification of tweets, several techniques such as SVM, maximum entropy (MaxEnt), and Naive Bayes (NB) classifiers are employed for both binary and fine-grained analysis. Among the three classifiers, the SVM results in better performance compared to other models. Furthermore, in [13], Debnath et al. proposed a lexicon-based technique to analyze WhatsApp chat-log data for assessing the situations and requirements during the crisis. The study was intended to automate the process of clustering relevant and irrelevant information based on querying a specific WhatsApp group conversation. The model builds a dictionary of keywords and phrases related to topics on flood or landslide for a specific query. The linked structure of WordNet is utilized to obtain

the collection of closely related words stored in the dictionary. More specifically, the hypernyms or hyponyms of some core or root words that are intuitively deemed as the root words of a specific topic are extracted. Then, a string-wise keyword matching algorithm is designed to identify the lines that contain one or more of these keywords. However, Nguyen et al. in [14] introduced a classification model based on CNN to categorize the disaster-related data into the situational and non-situational present on a social media platform. The most prominent n-gram information in disaster data is captured with convolution and max-pooling operations. Furthermore, the model is extended with a multi-layer perceptron to implement a multi-class task that yields better results compared to traditional CNN. Likewise, Imran et al. [15] formed a human-annotated Twitter corpus collected from several disaster events that took place at various points in time. To illustrate the effectiveness of the corpora, the authors trained three different classifiers, including SVM, NB, and random forests (RF). The authors published the first largest word-vectors termed *word2vec* trained on 52 million disaster-related tweets. The empirical results show that all three classifiers yield good results for multi-label classification.

In addition to this, a comparative study of document-level event classification models for recognizing different disaster-related event-types based on social media content is discussed in [16]. The work conducts an empirical survey on different ML techniques and compares them with other DL approaches such as CNN and hierarchical attention networks (HAN). The authors developed a new semantic word-vectors trained on a large corpus of highly standard news articles, which lead the model to achieve significant improvements in the performance. The study also reports that SVM and CNN outperform other classifiers. Alfarrarjeh et al. [17] proposed a new framework for geospatial opinion classification of disaster-related micro-blog data. The framework is intended to address three major challenges: (i) The performance evaluation of multi-modal sentiment classifiers, (ii) the geo-sentiment inconsistency among data elements in a local geographical area, and (iii) the identification of different sentiments from multimedia data. The sentiment scores are aggregated based on the geographical locations to extract more accurate local regional insights. To determine the sentiment score with high certainty, the model computes the variance between correlated sentiment labels either by entropy or variance metric.

However, To et al. in [18] proposed a five-step framework for investigating tweets accumulated during natural calamities that enhance situational awareness. They discussed two significant challenges of classifying relevant tweets: (i) A large number of unique hashtags used for a particular disaster, and (ii) lack of large labeled datasets for training classification models. Furthermore, they employed some techniques based on similarity matching to extract the relevant tweets, which significantly increases the performance. The extracted tweets are represented using the BOW model, where each dimension indicates a distinct word and its frequency in a particular tweet. The samples are trained using scikit-learn logistic regression and also tested on the same dataset to evaluate the accuracy of the model. Similarly, Neppalli et al. [19] worked on SA of tweets posted on Twitter during the disastrous hurricane, *Sandy* and visualized users' emotions on a topographical map gathered around the hurricane. The work further explained how users' emotions vary not only with their geographical

locations but also based on the distance from the accident spots. Both unigram and sentiment-related features (polarity clues, emoticons, acronyms, punctuation, url, and senti-scores) are analyzed for building feature vectors. Two classifiers, namely NB and SVM, are used to train the feature vectors, and SVM ($C = 0.1$) with the joint features trained on hurricane Sandy dataset performed better. Additionally, the researchers studied how the divergence of opinions in a tweet posted during the hurricane thumps the tweet retweetability. In [20], Hernandez-Suarez et al. suggested a methodology that considers Twitter as a social sensor. To accomplish this, the authors employed a sequential information extraction (ISE) method called as named entity recognition (NER), which defines particular objects, such as place, time, personage, and corporations. This framework uses the semantic, morphological, and contextual data concerning each term, composing a tweet and its surrounding context, thus allowing it to identify a named place (toponym) correctly. Firstly, the tweets get tokenized and transformed into word-vectors with a strong syntactic and semantic relationships established by neighboring words. To ensure high-performance of processing sequential data, a variant of RNN called a bidirectional long short-term memory (biLSTM) network is applied as an alternative to handcrafted feature extraction techniques. The biLSTM network operates with long-distance dependencies that feed-forward algorithms cannot handle. It is accomplished through contextual information processing, progressed in both directions of a word in a given tweet. It is evident from the preceding literature that there has been less work reported on disaster analysis from micro-blog text contents. Hence, this work is an attempt toward building a DL model for assessing the severity of disaster situations based on the content of the micro-blogs.

3 Proposed Methodology

There has been an increased interest in developing joint models to transfer knowledge through a process line, thereby overcoming the limitations of applying a single network model. Hence, in this paper, a model that combines the features of CNN and RNN for fine-grained SA of disaster-related micro-blog messages is discussed. Taking into account the capability that CNN can capture local features from input sequences and RNN can handle long-term dependencies, it would be beneficial to combine these features for any sequence modeling task. The proposed model named convolutional-recurrent neural network (C-RNN) consists of several layers that include a convolution and pooling layer, concatenation layer, and RNN layer followed by a fully connected layer with softmax output. The joint model C-RNN is depicted in Fig. 1, and the following subsections describe its essential components.

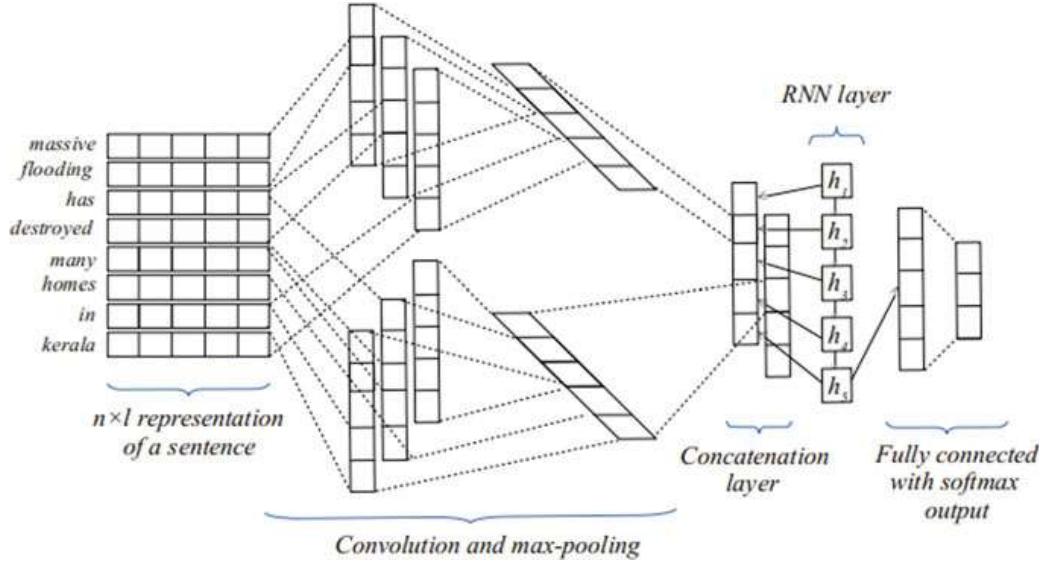


Fig. 1 The C-RNN architecture

3.1 Input Layer

Word-embeddings have shown great success in capturing fine-grained semantic regularities for text analysis. Hence, many DL models utilize word-embeddings to represent input documents. These vectors consist of high-dimensional real values that model syntactic and semantic information of words in a text corpus. Eventually, these vectors are used as pre-trained features for many classification tasks. Mathematically, given an input sequence x with l entries, each record is represented using a d -dimensional dense-vector. The whole input sequence x is represented as a feature map of $d \times l$ dimensions and input to the neural model.

3.2 Convolution Layer

Convolution is widely used for learning representations with sliding window of length w . Hence, for an input sequence $x_1, x_2, x_3, \dots, x_l$ of length l , let $c_i \in R^{w \times d}$ be the concatenated embeddings of w entries: x_{i-w+1}, \dots, x_i where w is the filter size and i ranges between 0 and $s + w$. Further, the embeddings x_i , with $i < 1$ and $i > l$ are zero padded. Finally, the representation $p_i \in R^d$ for w -gram x_{i-w+1}, \dots, x_i is computed from the convolution weights $W \in R^{d \times wd}$ as in Eq. 1.

$$p_i = \rho(W \cdot c_i + b) \quad (1)$$

where $b \in R^d$ is a bias vector and ρ is a nonlinear function either tanh or ReLu.

3.3 Max-Pooling Layer

To select the most prominent features from the features map a max-pooling operation is performed. The representations of all w -gram $p_i (i = 1 \dots s + w - 1)$ obtained from the convolution layer are used to create the representation for an input sequence x by max-pooling using Eq. 2.

$$x_j = \max(p_{1j}, p_{2j}, \dots) \quad \forall j = 1, \dots, d \quad (2)$$

3.4 RNN Layer

The RNN layer receives representations from the convolution layer in sequence and extracts global features to learn the long-term dependencies. Hence, this model is well known for sequential data processing, which utilizes internal memory for a series of inputs [21]. The RNN performs the exactly same task on each element of an input sequence recurrently; while considering the previous computations, it generates the output. A simple RNN is demonstrated in Fig. 2a.

In Fig. 2a, a hidden state h_t is determined based on an input x_t at current time step t , and previously hidden state h_{t-1} . Then, h_t can be written as in Eq. 3.

$$h_t = f(W^{hh} \times h_{t-1} + W^{hx} \times x_t) \quad (3)$$

where W^{hh} and W^{hx} are the weights on input x_t , and the previously hidden state h_{t-1} , respectively. The output y_t is the distribution of probability over a given vocabulary at time step t and is given by Eq. 4.

$$y_t = f(W^{hy} \times h_t) \quad (4)$$

The activation function f can be chosen appropriately as either sigmoid or softmax. The hidden state h_t is logically interpreted as the internal memory of the network; hence, the result of y_t is dependent only on the memory h_t at time t , and weight matrix

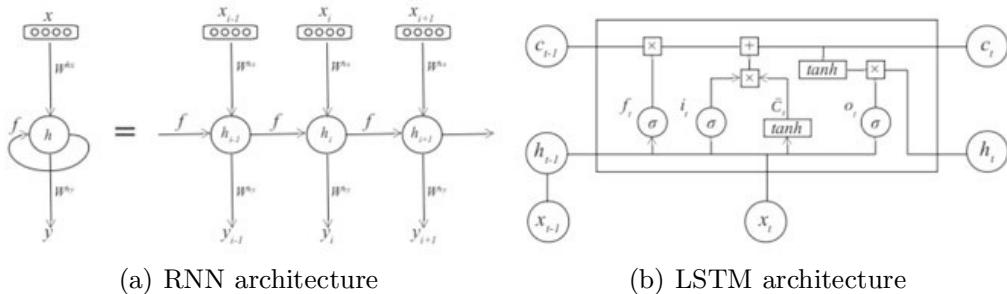


Fig. 2 RNN and its variant LSTM

W^{hy} . Unlike feed-forward networks that use different weights on different layers, RNNs share the same weights across all the iterations [22]. The above formulation is suitable only for small and fixed-length input sequences, but for arbitrary long sequences, it suffers from vanishing gradient or exploding gradient problems. Hence, a variant of RNN is used as an alternative pooling technique that learns long-term dependencies. In contrast to RNN, LSTM network units are constituted from four essential parts, namely a cell, an input gate, an output gate, and a forget gate [23]. During a specified time interval, the cell records information, and the gates control the flow of information from and to the cell. Two states are observed in this process: a hidden state and a cell state. A cell state emulates the memory of an LSTM cell, and a hidden state stores the result of this cell. Both the hidden state and a cell input together decide the information stored in the cell, i.e., to discard or to record new information. The LSTM is shown in Fig. 2b. Initially, at time step t , the forget gate employing a sigmoid function (ρ) decides on information to be stored in the cell. This function receives the output of previously hidden state h_{t-1} and accepts input x_t to compute the probability of retaining the information using Eq. 5.

$$f_t = \rho(W^i \times x_t + U^i \times h_{t-1}) \quad (5)$$

Consequently, LSTM decides over new information to be stored in the cell. The input gate (sigmoid) of LSTM computes the values to be updated based on Eq. 6. Then, \tanh function generates a candidate vector \bar{C}_t , which is now joined with cell state to bring an update to the cell. Equation 7 is used to generate the vector \bar{C}_t . The previous cell state C_{t-1} gets updated into new cell state C_t using Eq. 8.

$$i_t = \rho(W^n \times x_t + U^n \times h_{t-1}) \quad (6)$$

$$\bar{C}_t = \tanh(W^n \times x_t + U^n \times h_{t-1}) \quad (7)$$

$$C_t = f_t \times C_{t-1} + i_t \times \bar{C}_t \quad (8)$$

In Eq. 8, the forget gate f_t controls the gradients passes through it and allows memory operations to help in alleviating vanishing or exploding gradient problems of RNN. Lastly, the sigmoid function at output gate decides which portion of the cell to be returned using Eq. 9. The resultant cell state is given to tanh function and multiplies with another sigmoid function, as shown in Eq. 10.

$$o_t = \rho(W^o \times x_t + U^o \times h_{t-1}) \quad (9)$$

$$h_t = o_t \times \tanh(C_t) \quad (10)$$

3.5 Fully Connected Layer

3.5.1 Softmax Layer

The features generated from RNN form the penultimate layer are passed to a fully connected softmax layer. The result of the softmax function is equal to a categorical probability distribution, i.e., the probability that any of the classes are true and is given by $\text{softmax}(x) = \frac{e^x}{\sum_{k=1}^K e^{x_k}}$.

3.5.2 Loss Function

Errors on validation set are measured during training and stopped early if the validation error does not get improved. It is calculated using a loss function that compares the true value Y_i and predicted value \bar{Y}_i by a network model [24]. Here, a sparse categorical cross-entropy (SCCE) loss function is employed to estimate the loss error and is given by $H(Y_i, \bar{Y}_i) = \sum_{c=1}^C Y_i \log(\bar{Y}_i)$.

4 Experimental Setup

To study the empirical performance of the proposed C-RNN model, a set of experiments has been conducted. The following subsections present the necessary steps involved in the process of training and evaluating the C-RNN.

4.1 Preparation of Datasets

The effectiveness of the proposed model is evaluated on two different disaster-related datasets. The datasets are collected from Twitter during the flood and earthquakes using the Twitter Streaming API. The first dataset contains 12,000 tweets crawled related to flood and retrieved with search keywords flood, flooding, rain, hurricane, and hashtags containing the term flood, etc., with an average length of 13.5 tokens per sentence. Similarly, the second dataset contains 12,500 tweets related to earthquake extracted from Twitter with search keywords earthquake, landslide, tremor, and hashtags containing the term earthquake, etc., with an average length of 12.8 tokens per sentence. To effectively deal with tweets, it is necessary to carryout pre-processing over the dataset. Firstly, standard pre-processing techniques such as removing irrelevant text (retweets, hashtag, url, end-marks, etc.), case-folding, and lemmatization are applied. Further, an abbreviation lexicon is used to replace shrunk words (abt, ppl, b4, 2moro, etc.) by their complete forms and numbers to their text forms. Since the model evaluates both binary and fine-grained classification, two categories of

datasets are considered. For binary classification, the tweets are annotated by human experts with two-class labels as minor or major. However, for fine-grained (three-class) classification, the same tweets are relabeled with three classes as low, medium, or high. Finally, the model is trained using tenfold training (70%), validation (15%), and test (15%) set partitioning on both the datasets.

4.2 Word-Embeddings

The embedding matrix is initialized with a pre-trained word-vector trained on crisis data, as discussed in [25]. Alam et al. [9] trained a continuous bag-of-words (CBOW) *wrod2vec* model on a large crisis dataset with vector dimensions of 300, a context window size of 5, and $k = 5$ negative samples. The trained model consists of vocabulary with a size of 2.2 M words. Moreover, overfitting is a common problem in DNN while trained with a more number of parameters and types. A simple dropout regularization technique is used to overcome the problem of overfitting. The specific units from networks are selected randomly and ignored during training. In the embedding layer, the dropout rate is fixed to 0.30.

4.3 Network Training

The network is trained with an optimization that needs a loss function to estimate the model error. Gradient descent is the most popularly used algorithm to perform the optimization of DNN. It provides a way to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in R^d$ through updating the parameters in the opposite direction of the gradient of the objective function $\nabla_{\theta} \cdot J(\theta)$ with respect to the parameters. However, the variant of gradient descent Adadelta [26] is employed in this work. In Adadelta, the sum of gradients is recursively used as a decaying average of all past squared gradients. Thus, its update rule is given by, $\theta_{t+1} = \theta_t - \left[\frac{\text{rms}[\Delta\theta]_{t-1}}{\text{rms}[g]_t} \right] \times g_t$. The learning rate on classification loss is varied between 0.1 and 0.95. The batch size is set to 60, the maximum number of epochs is set to 100, and the dropout [27] rate on RNN layer is set to 0.10.

5 Results and Discussion

In this section, the empirical results of both binary and fine-grained classification tasks on two different Twitter datasets are presented. The proposed model is compared with two baselines, including linear SVM (lSVM) and multinomial Naive Bayes (MNB), that use BOW features. For SVM, only two labels are used viz, 0 =

Table 1 Results of proposed C-RNN

Datasets	Dataset description	Classifier	Fine-grained (three-class)	Binary
Dataset-1	Labeled tweets carrying information related to flood and calamity due to heavy rains	SVM	–	85.38%
		MNB	57.23%	82.81%
		C-RNN	65.59%	87.26%
Dataset-2	Labeled tweets carrying information related to earthquake and landsliding	SVM	–	84.75%
		MNB	54.62%	83.48%
		C-RNN	64.08%	85.57%

minor, and 1 = major. Therefore, each term of a tweet is represented as either 0 or 1 in a feature vector. Now, this feature vector and class labels are given to a linear kernel SVM classifier to classify tweets as minor or major. However, for the MNB classifier, the dataset with three-class labels, as discussed in Sect. 3, is employed. Table 1 summarizes the comparative analysis of C-RNN in terms of its accuracy against the baselines. It is observed that the proposed C-RNN outperforms the traditional models for both binary and fine-grained classification.

6 Conclusion

The sentiment analysis of user-generated text streaming over micro-blogs during disaster situations plays a critical role in crisis management and relief. In this research article, a joint model based on Deep-NN to investigate the intensity of disaster situations from the micro-blog contents is presented. More specifically, the capability of CNN to capture the local features and exploiting the learning capabilities of RNN that remembers long-range dependencies are combined to form a deep model named C-RNN. The model is trained on two distinct datasets containing tweets related to disaster. The performance of C-RNN is evaluated for both binary and fine-grained (three-class) classification tasks and compared with other baselines. The proposed C-RNN model outperforms the traditional machine learning models on both tasks.

References

1. Wadawadagi, R.S., Pagi, V.B.: An enterprise perspective of web content analysis research: a strategic road-map. *Int. J. Knowl. Web Intell.* **6**(2), 51–88 (2019)
2. Yang, M.C., Rim, H.C.: Identifying interesting Twitter contents using topical analysis. *Expert Syst. Appl.* **41**, 4330–4336 (2014)
3. Jianqiang, Z., Xiaolin, G.: Comparison research on text pre-processing methods on Twitter sentiment analysis. *IEEE Access* **5**, 2870–2879 (2017)
4. Choi, H.J., Park, C.H.: Emerging topic detection in Twitter stream based on high utility pattern mining. *Expert Syst. Appl.* **115**, 27–36 (2019)
5. Cobo, A., Parra, D., Navn. J.: Identifying relevant messages in a Twitter-based citizen channel for natural disaster situations. In: Proceedings of the 24th International Conference on World Wide Web Companion, pp. 1189–1194 (2015)
6. Beigi G., Hu X., Maciejewski R., Liu H.: An overview of sentiment analysis in social media and its applications in disaster relief. In: Pedrycz, W., Chen, S.M. (eds.) *Sentiment Analysis and Ontology Engineering. Studies in Computational Intelligence*, vol. 639. Springer, Cham (2016)
7. Wadawadagi, R.S., Pagi, V.B.: A deep recursive neural network model for fine-grained opinion classification. In: Santosh K., Hegadi R. (eds.) *Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2018. Communications in Computer and Information Science*, vol. 103. Springer, Singapore (2019)
8. Wang, Z., Jiang, W., Luo, Z.: Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2428–2437. Osaka, Japan (2016)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, pp. 1–9 (2013). [arXiv:1310.4546](https://arxiv.org/abs/1310.4546)
10. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
11. Rudra, K., Subham Ghosh, S., Ganguly, N., Goyal, P., Ghosh., S.: Extracting situational information from microblogs during disaster events: a classification summarization approach. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, pp. 583–592 (2015)
12. Stowe, K., Paul, M., Palmer, M., Palen, L., Anderson, K.: Identifying and categorizing disaster-related tweets. In: Proceedings of Fourth International Workshop on Natural Language Processing for Social Media, Austin, TX, Association for Computational Linguistics, pp. 1–6 (2016)
13. Debnath, P., Haque, S., Bandyopadhyay, S., Roy, S.: Post-disaster situational analysis from whatsapp group chats of emergency response providers. In: Tapia, A.H., Antunes, P., Bauls, V.A., Moore, K., Porto, J. (eds.) *Proceedings of the ISCRAM 2016 Conference Rio de Janeiro, Brazil* (2016)
14. Nguyen, D.T., Mannai, K.A.A., Joty, S., Sajjad, H., Imran, M., Mitra, P.: Rapid Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks (2016). arXiv preprint [arXiv:1608.03902](https://arxiv.org/abs/1608.03902)
15. Imran, M., Mitra, P., Castillo, C.: Twitter as a Lifeline: Human-Annotated Twitter Corpora for NLP of Crisis-Related Messages (2016)
16. Nugent, T., Petroni, F., Raman, N., Carstens, L., Leidner, J.L.: A comparison of classification models for natural disaster and critical event detection from news. In: Proceedings of the 5th IEEE International Conference on Big Data, (Big Data). IEEE, pp. 3750–3759 (2017)
17. Alfarrarjeh, A., Agrawal, S., Kim, S. H. Shahabi, C.: Geo-spatial multimedia sentiment analysis in disasters. In: Proceedings of the International Conference on Data Science and Advanced Analytics. IEEE, pp. 193–202 (2017)

18. To, H., Agrawal, S., Kim, S. H., Shahabi, C.: On identifying disaster-related tweets: matching-based or learning-based? In: Proceedings of the International Conference on Multimedia Big Data (2017)
19. Neppalli, V.K., Caragea, C., Squicciarini, A., Tapia, A., Stehle, S.: Sentiment analysis during Hurricane Sandy in emergency response. *Int. J. Disaster Risk Reduction* **21**, 213–222 (2017)
20. Hernandez-Suarez, A., et al.: Using twitter data to monitor natural disaster social dynamics: a recurrent neural network approach with word embeddings and kernel density estimation. *Sensors* **9**, 7 (2019). <https://doi.org/10.3390/s19071746>
21. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaei, E.: Recent Advances in Recurrent Neural Networks (2017). Arxiv Preprint [Arxiv:1801.01078](https://arxiv.org/abs/1801.01078)
22. Ma, S., Ji, C.A.: Unified approach on fast training of feedforward and recurrent networks using EM algorithm. *IEEE Trans. Sign. Process.* **46**(8), 2270–2274 (1998)
23. Nguyen, N., Le, A., Pham, H.T.: Deep bi-directional long short-term memory neural networks for sentiment analysis of social data. *IUKM* **2016**, 255–268 (2012)
24. Janocha, K., Czarnecki, W.M.: On Loss Functions for Deep Neural Networks in Classification (2017). Arxiv Preprint [Arxiv:1702.05659](https://arxiv.org/abs/1702.05659)
25. Alam, F., Joty, S., Imran, M.: Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets. In: Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018), pp. 556–559 (2018)
26. Zeiler, M.D. Adadelta: An Adaptive Learning Rate Method (2012). Arxiv Preprint [Arxiv:1212.5701](https://arxiv.org/abs/1212.5701)
27. Melamud, O., Goldberger, J., Dagan, I.: Context2vec: learning generic context embedding with bidirectional LSTM. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pp. 51–61 (2016)

WEKA Result Reader—A Smart Tool for Reading and Summarizing WEKA Simulator Files



Ranjit Panigrahi^{ID}, Samarjeet Borah^{ID}, and Udit Kumar Chakraborty

Abstract The Waikato Environment for Knowledge Analysis (WEKA) is popular tool for knowledge discovery and analysis. Researchers prefer WEKA over other similar tools due to the vast set of preprocessing and visualization mechanisms that it has to offer. Aimed at measuring performance of various supervised and unsupervised classification and clustering techniques, WEKA offers a wide range of data exploration facilities. However, a major shortcoming of this powerful tool is the output that it generates. Stored in ASCII, these files need manual conversion to spreadsheets for analysis and interpretation. Certain parameters even need recomputation, as these are returned as weighted averages. The current paper presents WEKA Result Reader a handy yet powerful tool that transforms WEKA output to spreadsheet. Thoroughly tested for system- and application-level performances, WRR proves to be a worthy and much-needed augmentation to WEKA.

Keywords WEKA · WEKA Result Reader · WRR · WEKA to spreadsheet · WEKA reader

1 Introduction

Waikato Environment for Knowledge Analysis (WEKA) is a leading machine learning simulator developed at the University of Waikato for knowledge mining and analysis. WEKA provides a graphical user interface equipped with various

R. Panigrahi (✉) · S. Borah · U. K. Chakraborty
Sikkim Manipal Institute of Technology, Sikkim Manipal University, Gangtok 737136, India
e-mail: ranjit.panigrahi@gmail.com

S. Borah
e-mail: samarjeet.b@smit.smu.edu.in

U. K. Chakraborty
e-mail: udit.c@smit.smu.edu.in

preprocessing, supervised and unsupervised algorithms, all of which can be visually simulated for analysis and predictive modeling [1]. Equipped with the training and testing samples, WEKA allows the users to choose from a substantial number of performance measures to precisely characterize supervised and unsupervised models. Owing to their ability to intelligently assign unknown objects to appropriate classes, WEKA supervised classifiers are particularly popular among researchers [2]. Using numerical similarity between objects, derived from a set of known objects, WEKA provides researchers with some very handy tools and methods to explore, analyze and build appropriate models.

Its popularity notwithstanding, WEKA has a serious shortcoming in terms of ease of use. Developed to aid in analysis of data, the tool presents information to the user as an ASCII output file. The cumbersome and painstaking task of arranging and copying this data into a spreadsheet for analysis has to be done manually by the user. Furthermore, WEKA returns weighted average values as the final output for measures of true positive rate (TPR), false positive rate (FPR), precision and other performance measures. These values need to be interpreted correctly and recomputed for measurements.

This paper presents the WEKA Result Reader (WRR) a lightweight multifunction tool designed to assist researchers reap maximum benefits from WEKA. Functionally, WRR is equipped to:

- a. Scan earmarked folders along with subfolders for WEKA output files. This smart search facility does not get into reading all files and intelligently bypasses non-WEKA files.
- b. Extract content from WEKA result files and writes specified performance outputs in spreadsheets.
- c. Intelligently computes the actual performance metrics from weighted averages, thereby empowering the users to deduce actual meaning from data retrieved.

The WRR tool thus augments the efficiency of WEKA and saves time and effort in the research.

2 The WEKA Result Reader (WRR)

WEKA enables the analysis of performance measures of various supervised classifiers using different metrics. The choice of the metric, however, depends entirely on the domain of study and type of dataset on which the classifiers are modeled and tested. Major performance measures of interest are training and testing time, correct and incorrectly classified instances, model accuracy, misclassification rate, kappa statistics [3–5], mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), root relative squared error (RRSE), true and false positive rate, precision, F-measure, receiver operation curve (ROC) value [6] and Matthews

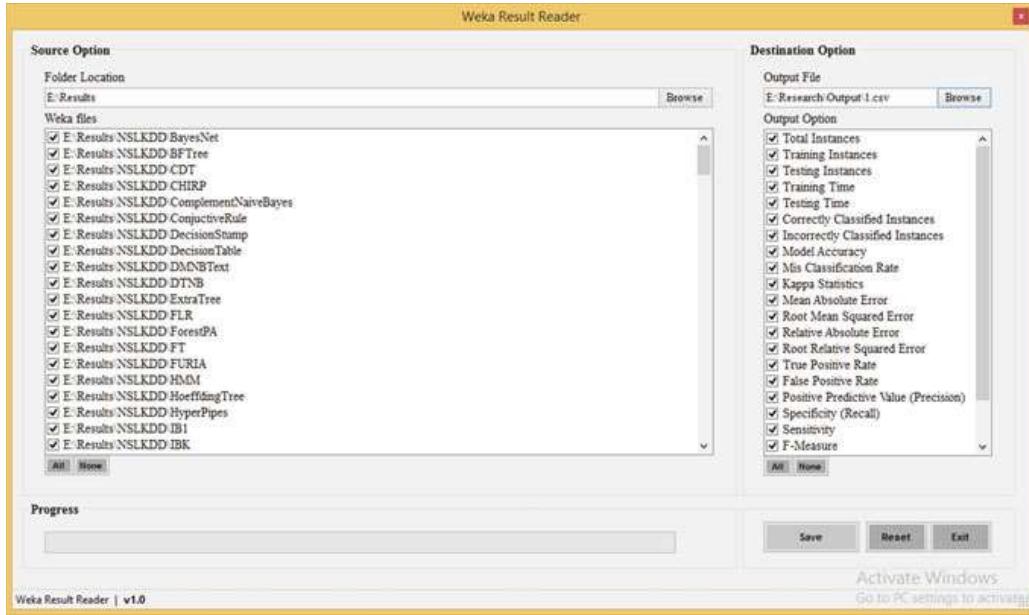


Fig. 1 WEKA Result Reader (WRR) tool

correlation coefficient (MCC) value [7]. The proposed WRR tool extracts these information from the WEKA result file and summarizes them in a CSV file for analysis. The result reader tool for WEKA is depicted in Fig. 1.

Featuring a simple GUI, the WRR presents the user with a screen split into two halves representing the source and destination section, respectively. The source section on the left-hand side shows the location of the folder where the result files are available and the destination section shows the output file location where the results will be extracted. The destination section features selecting specific performance measures apart from the destination path. The source section enables the user to select a specific classifier result file for which the performance values need to be extracted.

On selecting the location of the result file to be extracted in the source section, the system automatically scans the folder and its sub-folders recursively for the source files returning only valid WEKA files, thus improving the searching time significantly. Upon initiating the extraction using “Save” button, the extracted information is stored in the specified file in “.csv” format. A progress bar dynamically depicts the progress of the extraction process. Figure 2 shows a typical WRR screenshot.

The WRR tool benefits the users with its unique ability to quickly read and summarize performance values of supervised classifiers from the WEKA result files. The output CSV file can be easily analyzed in any spreadsheet application, thereby saving time. This system will therefore significantly ease research in the field of machine learning. Designed as modules, the tool allows easy integration of new features as upgrades and thus allows easy mapping with future WEKA versions.

The screenshot shows a Microsoft Excel spreadsheet titled "1 (version 3) [Recovered] - Excel". The data is organized into columns representing different classifier categories and their specific implementations. The rows represent individual experiments or runs. Key columns include "Classifier Category", "Classifier", "Total Instances", "Training Instances", "Testing Instances", "Training Time", "Testing Time", "Correctly Classified Instances", "Incorrectly Classified Instances", "Model Accuracy", "Misclassification Rate", "Kappa Statistics", "Mean Absolute Error", "Root Mean Squared Error", "Relative Absolute Error", "Root Relative Squared Error", "TP Rate", and "FP Rate". The data spans multiple rows and columns, providing a detailed performance comparison for each classifier across different datasets.

Fig. 2 WEKA Result Reader (WRR) output tool

3 Performance Evaluation

The WRR tool was tested for performance against both application- and system-level performances measures. At application-level, the speed, stability and consistency were considered for performance evaluation. On the other hand, memory usage, CPU utilization, number of page faults and thread cycle were monitored at system level.

3.1 Application-Level Testing of WRR Tools

Under this evaluation scheme, three recent datasets, NSL-KDD [8], ISCXIDS2012 [9] and CICIDS2017 [10], were used to train and test 54 classifiers of six different classifier groups found in WEKA. The output result for each dataset for each classifier was stored in separate ASCII files. As a result, there existed three dataset folders, each folder containing 54 files pertaining to each classifier. Now, in order to test the WRR tool, 20 different experiments are conducted to extract performance information from the result files for each dataset. The speed of WRR tool was recorded as the time required for extracting performance information from the result file. This consisted of the time taken to read the result file and write that consolidated information to a CSV file. The speed of the WRR tool for each dataset and experiment recorded is presented as Table 1.

Further, the time spent for a single result file of an experiment of a dataset by averaging the total time taken by the WRR tool for reading 54 such files of the concern dataset was calculated. Finally, the average time taken per result file for all

Table 1 Execution speed of WRR tool

Experiment No.	CICIDS2017		ISCXIDS2012		NSL-KDD		Average speed	
	Total time	Time per file	Total time	Time per file	Total time	Time per file	Total time	Time per file
Ex#01	1.958	0.036	1.292	0.024	2.364	0.044	1.871	0.035
Ex#02	1.821	0.034	1.163	0.022	2.040	0.038	1.675	0.031
Ex#03	1.820	0.034	1.139	0.021	2.037	0.038	1.665	0.031
Ex#04	1.817	0.034	1.154	0.021	2.064	0.038	1.678	0.031
Ex#05	1.811	0.034	1.439	0.027	2.035	0.038	1.762	0.033
Ex#06	1.874	0.035	1.192	0.022	2.033	0.038	1.700	0.031
Ex#07	1.892	0.035	1.144	0.021	2.044	0.038	1.693	0.031
Ex#08	1.853	0.034	1.180	0.022	2.032	0.038	1.688	0.031
Ex#09	1.880	0.035	1.150	0.021	2.027	0.038	1.686	0.031
Ex#10	1.850	0.034	1.193	0.022	2.597	0.048	1.880	0.035
Ex#11	1.858	0.034	1.166	0.022	2.033	0.038	1.686	0.031
Ex#12	1.851	0.034	1.171	0.022	2.619	0.049	1.880	0.035
Ex#13	1.860	0.034	1.140	0.021	2.115	0.039	1.705	0.032
Ex#14	1.881	0.035	1.149	0.021	2.029	0.038	1.686	0.031
Ex#15	2.060	0.038	1.153	0.021	2.051	0.038	1.755	0.032
Ex#16	1.849	0.034	1.148	0.021	2.100	0.039	1.699	0.031
Ex#17	1.856	0.034	1.140	0.021	2.037	0.038	1.678	0.031
Ex#18	1.853	0.034	1.149	0.021	2.037	0.038	1.680	0.031
Ex#19	1.844	0.034	1.223	0.023	2.183	0.040	1.750	0.032
Ex#20	1.853	0.034	1.156	0.021	2.051	0.038	1.687	0.031
Average speed	1.867	0.035	1.182	0.022	2.126	0.039	1.725	0.032

the datasets and for all the experiments was computed, which ultimately yields the approximate time taken by the WRR tool to generate a CSV file from the WEKA result file. The overall execution speed of WRR tool is represented as Fig. 3.

It can be seen from Fig. 3 that the execution speed of WRR tool to process a single file is consistent throughout all the 20 experiments. The processing time for the result files of ISCXIDS2012 dataset is less as compared to other datasets. It is because ISCXIDS2012 is a binary dataset, which contains less class information. Further, the overall execution speed of WRR tool is found to be 0.032 s per result file.

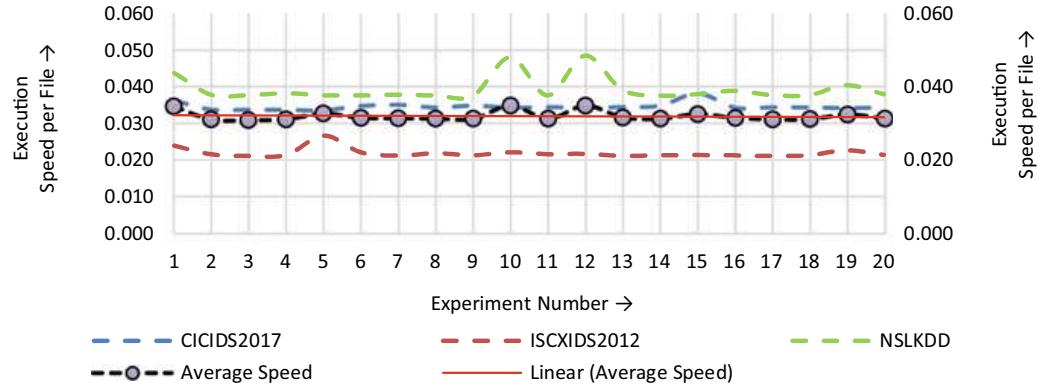


Fig. 3 Overall execution speed of WRR tool per result file

3.2 System-Level Testing of WRR Tools

During the application-level testing, WRR tool claims an impressive duration of 0.032 s to process a single result file. Further, a system-level test of the proposed tool on those test aspects was conducted. In order to investigate and measure the system-level performance of WRR tool, a third-party tool called SysGauge process monitor [11] was used. SysGauge is a free to use performance monitoring software for monitoring the CPU usage, memory usage, operating system performance, the status and resource usage of running processes, disk space usage, disk read and write activity, disk read and write transfer rate [12] of individual processes.

In order to test the proposed WRR tool, an Intel-based Core i5-4210U CPU system with memory size of 4 GB having processing speed of 1.7 GHz was deployed. The operating system for testing purpose was 64-bit Windows 8.1 Enterprise Edition. The WRR tool was run continuously for a period of 300 s, while the SysGauge monitored the activity of the WRR in the background. The results returned are as shown in Table 2.

Table 2 is self-explanatory. In the course of 300 s run, the average CPU usage achieved was 23.5%, which is impressive in multi-threaded environment. The good thing is that the CPU usage is consistent throughout the execution period (Fig. 4).

Further, the memory consumption by WRR tool is also quite remarkable. It was found that the proposed model consumes 36.1 MB of memory which is even below 1% of total physical memory of the testing environment. This shows that the system will not affect any other background processes during its course of the run. Similarly, the number of threads and handles used by WRR model was monitored, and it was found that as a multi-threaded application, WRR tool deploys an average of 11 threads and 255 handles while fetching information from WEKA result files. The 11 threads returned a reasonable number of handle count.

A page fault [13] is a situation, when a program tries to access code or data that is available in its address space but is not currently located in the system memory. A system is said to be effective if it is highly fault tolerant in nature. In case of the proposed WRR tool, the system shows an average of 681 frames per second, which

Table 2 SysGauge observation of WRR tool

Performance monitor	From	To	Average	Minimum	Maximum
CPU usage	2019/02/09 09:49:19	2019/02/09 09:54:29	23.5%	0.0%	25.0%
Memory used	2019/02/09 09:49:19	2019/02/09 09:54:29	36.1 MB	0.0 MB	40.4 MB
Thread count	2019/02/09 09:49:19	2019/02/09 09:54:29	11	0	13
Handle count	2019/02/09 09:49:19	2019/02/09 09:54:29	255	0	271
Page fault rate	2019/02/09 09:49:19	2019/02/09 09:54:29	681 F/s	0 F/s	4114 F/s
Throughput	2019/02/09 09:49:19	2019/02/09 09:54:29	0.3 MB/s	0.0 MB/s	0.6 MB/s
Reading speed	2019/02/09 09:49:19	2019/02/09 09:54:29	0.29 MB/s	0.20 MB/s	0.55 MB/s
Writing speed	2019/02/09 09:49:19	2019/02/09 09:54:29	0.01 MB/s	0.00 MB/s	0.04 MB/s

is not encouraging. The reason behind such a significant amount of page fault is that the entire process of reading result files and writing to CSV files was tested through a timer, which automatically calls the entire reading and writing process in a specific time interval. It is obvious that the timer initiated the reading process when already a reading process is going on. Similarly, a maximum of throughput, reading and writing speed of WRR tool has been achieved by 0.6 MB/s, 0.55 MB/s and 0.04 MB/s, respectively.

4 Conclusion

WEKA's supervised classifiers produce results on screen or through ASCII files. The ASCII files contain a huge amount of information along with the relevant information of the research domain, and locating and retrieving information on the interest area is a tedious task. In order to retrieve and tabulate such information, a WEKA Result Reader (WRR) tool has been developed which is presented in this paper. The WRR tool is very fast and reads the WEKA result files accurately and tabulates the required information in CSV format, which helps the researchers to analyze the results quickly in any spreadsheet application. The performance of the proposed tool has been tested both at application and system level. At the application level, the system needed approximately 0.032 s to read each WEKA file. The tool is low on memory usage and has a reasonable handle count. However, the system exhibits slightly higher amount of page faults. The tool is effective, efficient, reliable and user friendly. Most

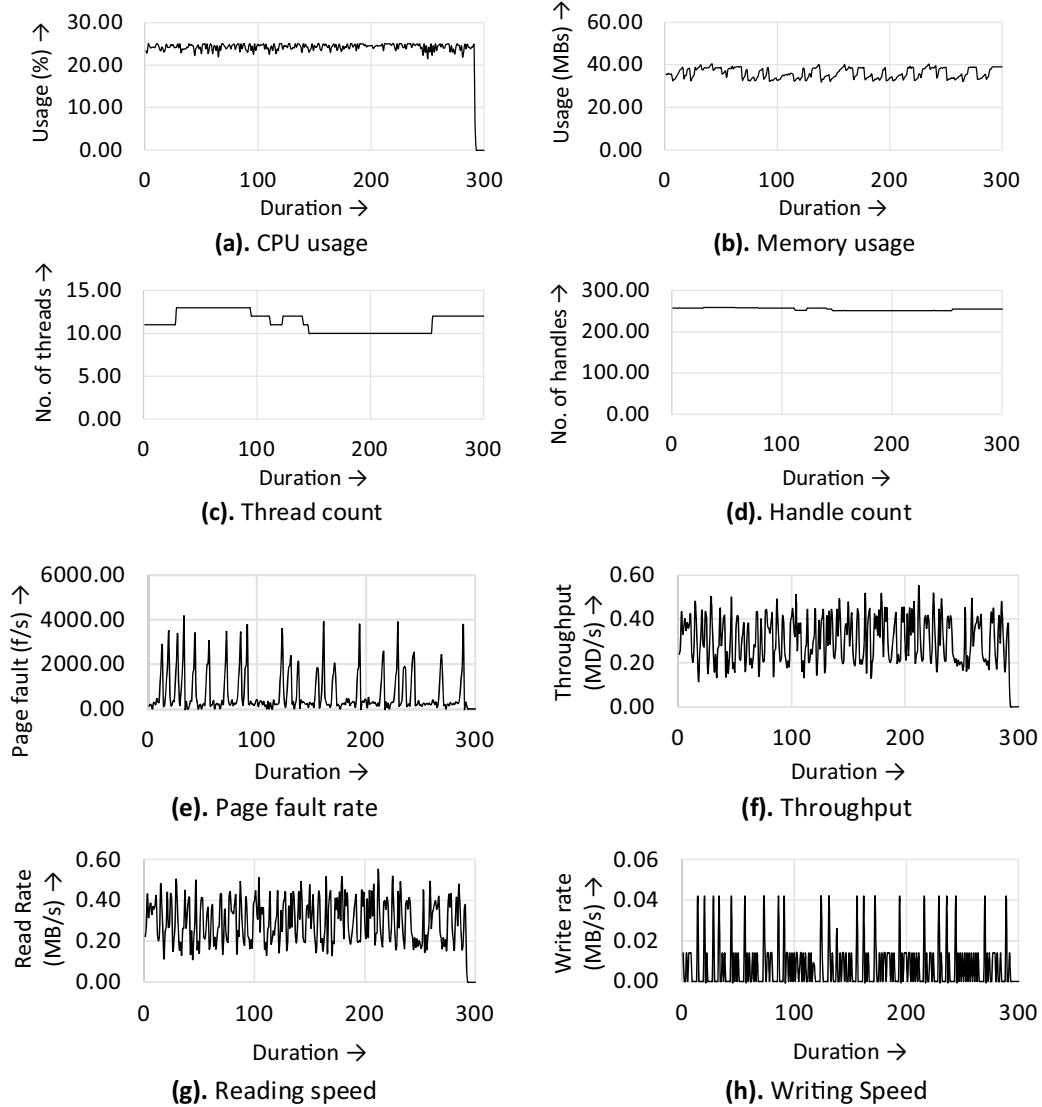


Fig. 4 WEKA Result Reader process statistics

importantly, it is one of the kinds and would certainly augment WEKA with its features, thereby also helping users in interpretation of results.

References

1. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, San Francisco (2011). Retrieved 19 Jan 2011
2. Wing, S., Johnson, J.R.: Machine Learning Techniques for Space Weather, pp. 113–145. Elsevier. ISBN 9780128117880 (2018)
3. Mishra, S.: Understanding the calculation of the kappa statistic: a measure of inter-observer reliability. Int. J. Acad. Med. 2(2), 217 (2016)

4. Du, H.: Cohen's kappa in plain English, <https://stats.stackexchange.com/questions/82162/cohens-kappa-in-plain-english>. Last accessed 8 Feb 2019
5. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. *Biometrics* **33**(1), 159–174. JSTOR 2529310. PMID 843571 (1977)
6. Radford, B.J., Richardson, B.D., Davis, S.E.: Sequence aggregation rules for anomaly detection in computer network traffic. In: American Statistical Association, Symposium on Data Science and Statistics, pp. 1–13, May 2018
7. Matthews, B.W.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta (BBA) Protein Struct.* **405**(2), 442–451 (1975)
8. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. IEEE (2009)
9. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **31**(3), 357–374 (2012)
10. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP, pp. 108–116 (2018)
11. SysGauge System Monitor, <http://www.sysgauge.com/>, Flexense Ltd. Last accessed 7 Feb 2019
12. SysGauge 5.8.16, <https://www.techspot.com/downloads/6957-sysgauge.html>, TechSpot Inc. Last accessed 7 Feb 2019
13. Weik, M.H.: Page fault. In: Computer Science and Communications Dictionary. Springer, Boston, MA (2000)

Predicting Reliability of Web Services Using Hidden Markov Model



Shridhar Allagi and Pradeep Surasura

Abstract The dynamic environment applications are the approaches for the construction of distributed business framework. The quality of service (QoS) banks of these systems of these systems depends on the Web services (WS) and Internet associations. Outlining productive and viable reliability prediction of WS have turned into an imperative issue. This paper focuses reliability of systems by using hidden Markov model (HMM) for the modeling of failure and prediction of Web service reliability. The forward--backward estimation-maximization is used to estimate the modeling parameters of HMM and by using Bayesian Information Criterion (BIC), model selection is done. The favorable circumstances and disadvantages of this approach concerning regular modeling are examined. Examination of these models is done on real Web service data. Regarding reliability prediction, the hidden Markov model performs better with respect to other regular models.

Keywords Web service · Reliability · Hidden Markov chains · EM · BIC

1 Introduction

Cloud computing and service-oriented computing (SOC) are emerging as a significant computing paradigm, which makes the rapid change in the design, delivery and consumption of software applications. Web services (WS) are the fundamental elements of SOC and cloud computing which supports in the rapid development of applications in heterogeneous environments.

Service-oriented and cloud computing systems are largely used in various domains like multi-media services, business to business collaboration, automotive systems, etc. Since these systems are depend on the WS, the reliability of these systems also depends on the reliability of WS. To make these systems as reliable, one has to

S. Allagi (✉) · P. Surasura

Department of Computer Science and Engineering, K L. E. Institute of Technology, Hubballi, 580027, Karnataka, India

e-mail: shridharallagi1@gmail.com

select reliable WS, and for that, the reliability of WS must be predicted. Predicting reliability at the early phase like in system architecture design phase helps to reduce the cost of re-engineering. Hence, WS reliability techniques are basics for building SOC and cloud computing applications.

Presently, many stochastic models exist to identify the software failure rates. The objective of these models is to gauge the software reliability in present and future, considering past failures and the corrections made. For stochastic software reliability models, Gaudoin [1] and Chen and Singpurwalla [2] proposed a general framework which is self-exciting random point processes, which is given as follows. Let T_l , $l > 0$ be the successive software failure in times, with $T_0 = 0$. The software may be corrected or not, and restarted after each failure in software. Then, let N_t be the number of failures arrived between time 0 and t ; and let $X_l = T_l - T_{l-1}$, $l > 0$, be the time between consecutive software failures. The failure process for these software is equivalent to one of random processes among $\{T_l\}_{l>0}$, $\{N_t\}_{t \geq 0}$, or $\{X_l\}_{l>0}$. Then the failure intensity is defined as follows:

$$\lambda_t = \lim_{dt \rightarrow 0} \frac{1}{dt} P(N_{t+dt} - N_t = 1 | H_t) \quad (1)$$

where $H_t = \sigma(\{N_s\}_{s \leq t})$ is the interior filtration of failure process.

At time t , reliability R_t indicates the probability that there is no software failure for any time length Γ by considering the past failure process. This is given as below

$$\begin{aligned} R_t(\tau) &= P(N_{t+\tau} - N_t = 0 | H_t) = P(T_{N_t+1} - t > \tau | H_t) \\ &= \exp\left(- \int_t^{t+\tau} \lambda_s ds\right) \end{aligned} \quad (2)$$

Majority of the software reliability models presume that number of failures $\{N_t\}_{t \geq 0}$ is a non-homogeneous Poisson process (NHPP). And these models consider the failure rates are deterministic and continuous functions of time: $\lambda_t = \lambda(t)$.

Following software reliability models are the most usual NHPP models among them:

- Duane or power law process (PLP) model [3]
- The Goel-Okumoto model [4] (GO)
- Moranda geometric mode [5] (MG)
- Gaudoin, Lavergne and Soler's [6] proportional model

However, above mentioned models expect a rectification for every failure, and the debugging for those failures is effectiveness in homogeneous time. Practically speaking, after software failures, maximum time systems are rebooted without performing any rectification in system. And debugging is done when an adequately vast measure of failure has happened.

In the circumstances of service-oriented and cloud computing applications, the reliability of these systems depends on the framework and primarily relies on the remote WS and client attributes (e.g., operational profiles, geographic areas, network/system conditions, and so on). WS are affected by the Internet connections and numerous other environmental and operational elements. Various clients might observe very distinctive reliability for the same WS. So, it is a difficult task to construct reliability models for such systems which takes these facts into account. This is one of the hidden Markov model's case [7].

Rest of the paper is organized as follows: Most commonly used NHPP models are explained in Sect. 2. Section 3 explains the way hidden Markov model predicts the reliability, by considering the hidden states restoration. It also explains the use of BIC values for model selection. HMM is applied in Sect. 4 for real-time dataset to check failures and results obtained are compared with other models. And Sect. 5 concludes the paper.

2 Models of NHPP

This section describes other models which uses number of failures $\{N_t\}_{t \geq 0}$ as non-homogeneous Poisson process (NHPP), to predict reliability deterministic and continuous function of time is used as failure intensity.

The primary purpose behind the wide utilization of NHPP in software reliability is the straight forwardness of their utilization. However, principal disadvantage of NHPP models is the hypothesis of those systems. The hypothesis states as failure intensity is a continuous function of time: it is more practical to believe that debugging affects discontinuity in failure intensity.

A. Duane model or power law process (PLP)

This model assumes following two things to remove fault and increase the software reliability growth

- The system starts with N_0 faults at $t = 0$
- At the point when a fault reveals itself by causing system into failure, the fault is instantly uprooted and the system comes back to its working state.

Considering N as Poisson random variable, with mean M , and X_i be identically distributed and s-independent, with probability distribution function $f(x)$. The unconditional stochastic process would be a NHPP with rate:

$$\lambda(t) = Mf(x) \quad (3)$$

In particular for stochastic process, the NHPP rate (fault detection rate) is given as follows

$$\lambda(t) = \alpha\beta t^{\beta-1}, \alpha \in \mathbb{R}^+, \beta \in \mathbb{R}^+ \quad (4)$$

B. Moranda geometric (MG) model

Moranda geometric model for the generated data in depict to explain fault-detection phenomenon is given as follows:

$$\Pr\{N_i = n\} = \frac{\{\lambda k^{i-1}\}^n}{n!} \cdot \exp[-\lambda k^{i-1}] \quad (5)$$

where

N_i = number of detected faults in the i th interval of time $(T_{i-1}, T_i]$ with $T_0 = 0$; $\lambda > 0$; $0 < k < 1$; and $n \geq 0$.

λ = average number of faults in interval $(0, T_1]$

k = ratio of faults decreased in each iteration.

The overall fault-detection rate by MG model is given as follows:

$$\Lambda_i = \sum_{i=1}^n \lambda k^{i-1}, \lambda \in \mathbb{R}^+, k \in (0, 1] \quad (6)$$

C. Goel-Okumoto (GO) model

The GO model is similar to other exponential-class of NHPP model, but parameters of model have quantitative measures for software reliability. This model assumes that issue bringing on failure is reviewed immediately; otherwise, reoccurrence of that failure is not considered.

The fault-detection rate is given as follows:

$$\lambda(t) = \lambda e^{-\phi t}, \lambda \in \mathbb{R}^+, \phi \in \mathbb{R} \quad (7)$$

where ‘ t ’ represents time of interval between failures; λ represents the expected number of defects and ϕ is the rate of decrease of failure.

D. Gaudoin, Lavergne and Soler (GLS) model

Gaudoin et al. considering debugging effects defined a class model called as proportional model

$$\forall i \geq 1, \Lambda_{i+1} = \Lambda_i e^{-\Theta_i} \quad (8)$$

where Θ_i represents consecutive debugging effects

$\Theta_i = 0 \Rightarrow \Lambda_{(i+1)} = \Lambda_i$, means debugging has no effect;

$\Theta_i > 0 \Rightarrow \Lambda_{(i+1)} < \Lambda_i$, means debugging has positive impact and reduces failure rate, which makes the debugging as a good quality; $\Theta_i < 0 \Rightarrow \Lambda_{(i+1)} > \Lambda_i$, means debugging has negative impact and increases failure rate; therefore, debugging is of bad quality. The improved version of this model considers the good and bad quality of debugging, which is given as follows

$$\forall i \geq 1, \Lambda_i = (1 - \alpha_i - \beta_i) \Lambda_{i-1} + \mu \beta_i \quad (9)$$

where $\llbracket \alpha \rrbracket_i$ represent the good quality of debugging and β_i represents the bad quality of debugging. For simple model $\alpha_i = \alpha$ and $\beta_i = \beta$ is considered for all i .

3 Hidden Markov Model for Reliability Model

HMM is a Markov process with unobservable states [8]. The HMM modeling methodology is described in this section.

3.1 Modeling the Failure Process with HMM

To apply HMM model, it is essential to presume that rate of failure process $\Lambda = \{\Lambda_l\}_{l \geq 1}$ is of finite set with M elements in it. Consider, $\Lambda = \{\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(M)}\}$ is a possible set which results to following:

- Λ is Markov chain;
- WS failures are observed in different time for different users is same as times between failures $\{X_l\}_{l \geq 1}$ are independent;
- WS failure are induced from server rises the failure rate exponentially for all users is same as X_l has an exponential distribution with parameter $\lambda^{(j)}$.

These premises meet the process of hidden Markov chain. The HMM looks at the discontinuities in failure intensity brought by the debugging which makes it superior model compared with other NHPP models.

The HMM is defined by using below given parameters:

- The initial state's distribution Λ_1 , is given by $\pi_j = P(\Lambda_l = \lambda^j) \quad 1 \leq j \leq K$.
- $P(\Lambda_{i+1} = \lambda^{(l)} | \Lambda_i = \lambda^{(j)}) = p_{jl}^{(i)}, i \geq 1, 1 \leq j \leq K, 1 \leq l \leq K$ is the transition probabilities; do not depend on i . Therefore, p_{jl} is used to define transition parametric matrix P .
- The failure rates are given as $(\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(K)})$.

3.2 Parameter Estimation

HMM parameter valuation includes batch learning, construct either in light of the Baum--Welch (BW) algorithm or specific expectation–maximization (EM) techniques [9], like, on numerical optimization techniques (like gradient descent algorithm). In either case, valuation of HMM parameters is done enough number of

times compared to training iterations, so that some independent validation data is maximized for some objective function (like maximum likelihood).

In HMM complete data $y_1^n = (x_1^n, \lambda_1^n)$ divided into observed data x_1^n and missing data λ_1^n . This missing data can be restored by EM algorithm. The EM algorithm works in two steps as follows:

- Estimation (E): The function Q will be determined

$$Q(\eta, \eta^{(k)}) = E_{\eta^{(k)}} [\log P_\eta(\Lambda_1^n, X_1^n = x_1^n) | X_1^n = x_1^n] \quad (10)$$

- Maximization (M): The function Q will be maximized with respect to η

$$\eta^{(k+1)} = \arg \max_{\eta} Q(\eta, \eta^{(k)}) \quad (11)$$

The maximization sequence leans toward consistent solution once if it is iterated for some number of items. To get consistent solution, this paper uses 50 iterations.

The BM algorithm [10] is used to calculate distribution values, from start time toward forward and from end time toward backward direction; hence, it is also known as *forward-backward* algorithm.

3.3 Restoration of Hidden States

The unknown states are restored, by assigning values to that state sequence λ_1^n . One of the methods to restore hidden states is consider known sequence values x_1^n to process hidden state's most likely value. This leads to maximum a posteriori, which is implemented using Viterbi algorithm, and it is computed as below:

$$\arg \max_{\lambda_i^n} P_\eta(\Lambda_1^n = \lambda_1^n | X_1^n = x_1^n) \quad (12)$$

The correlation among original data and hidden state restored is simple since the normal bury failure time when $\Lambda_i = \lambda^j$ is $1/\lambda^j$.

Figure 1 demonstrates number of access failed for a WS dataset group (WSDG2, of 25 users) which is established on the ideal state sequence. The restoration of states is done using Viterbi algorithm.

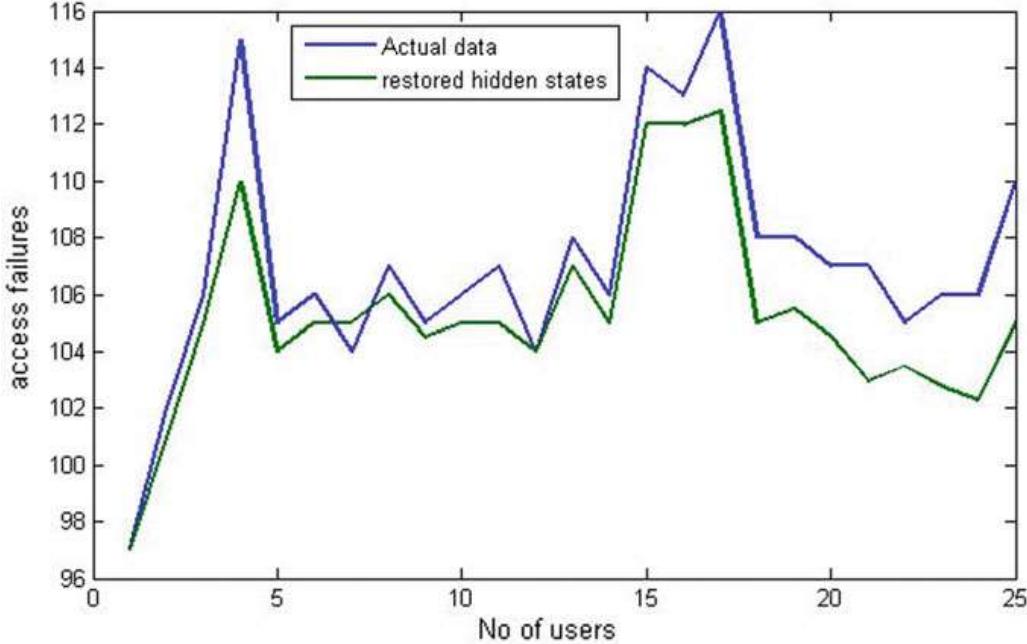


Fig. 1 Most likely sequence for WSDG2

3.4 Bayesian Information Criterion (BIC)

It is difficult to judge the good model among the models having moderate complexity. The criterion based on a penalization of the log-likelihood is used to overcome this problem. Schwarz criterion or Bayesian Information Criterion [11] is a criterion for selecting a good model from the set of various models. The model having lowest BIC value is considered as good model. The BIC evaluated as

$$\text{BIC}(M) = \log(P_{\hat{\eta}M}(x_1^l)) - \frac{v_M}{2} \log(l) \quad (13)$$

where l is length of observed failure sequence; M represents number of hidden states; maximum log-likelihood is represented by $\log(P_{\hat{\eta}M}(x_1^l))$; and v_M represents independent parameters in η_M .

The BIC rule is made out of a term evaluating the fit between the information and the model (like, the log-likelihood) and of a term punishing composite models. Consequently, augmenting this criterion ideally prompts choosing models offering a mixture of the fit and composite. In this paper, log-likelihood for models is identified and compared the BIC values to get fit best model.

4 Experiments

To evaluate the WS reliability prediction from various models, WS dataset is taken from quality of service (QoS) dataset [12]. This dataset includes more than five thousand WS invocation from 50 different users on 100 real-world WS. For each WS, a numbers of invocations made are observed and number of successful and failure invocations are recorded. From this observation, failure rate is calculated.

These 100 WS are used to create 20 different groups each having 5 WS. A group contains 5 WS of same functionality, and there is a probability that the user can switch over any of remaining 4 WS on successful/failure of running WS. The probability of transition of WS on successful invocation is captured in a transition probability matrix of 5×5 .

The hidden Markov model is applied to these groups, the description for first group (WSDG1) is given in this section, and comparisons of all the models for all 20 groups are done. The first thing is to choose the hidden states M , here group WSDG1 is run with $M_{\min} = 1$ to $M_{\max} = 7$. There it has been found that for $M = 5$, BIC value is less which fits the complexity of data. Same thing is repeated for other group of the dataset, and it has been found that $M = 5$ gives the better result so hidden states are considered is 5. Table 1 lists the transition probability matrix for 1st group (WSDG1).

The emission matrix is derived as the number of times failure occurred in the invocation of the WS. Here 5×50 matrix is used to represent emission matrix. The WS W1 has made 5129 number of invocation by fifty different users; among them 98 times failures are occurred for user1. From this, one can make WSDG1 group's emission matrix entry as 0.0191, i.e., EMIS_MATRIX(1,1) = 0.0191.

Sequence of emission symbols and sequence of states are generated using emission and transition matrixes. Most likelihood states are identified by EM algorithm having executed in an iteration of 50. The most likelihood states are restored by Viterbi algorithm. This restoration of states for dataset group-1 (WSDG1) is given in Fig. 2.

The calculation of logarithmic likelihood probability of failure is done for WSDG1 by known and restored states. Then the BIC value for the WSDG1 is calculated for HMM model. The BIC values of WSDG1 for other models are calculated. Result shows that BIC value for HMM model is least which indicates the best fit model.

Table 1 Transition matrix for WSDG1

	W1	W2	W3	W4	W5
W1	0.840	0.050	0.030	0.030	0.050
W2	0.100	0.750	0.025	0.025	0.100
W3	0.035	0.050	0.800	0.075	0.040
W4	0.060	0.040	0.030	0.770	0.100
W5	0.025	0.020	0.050	0.025	0.880

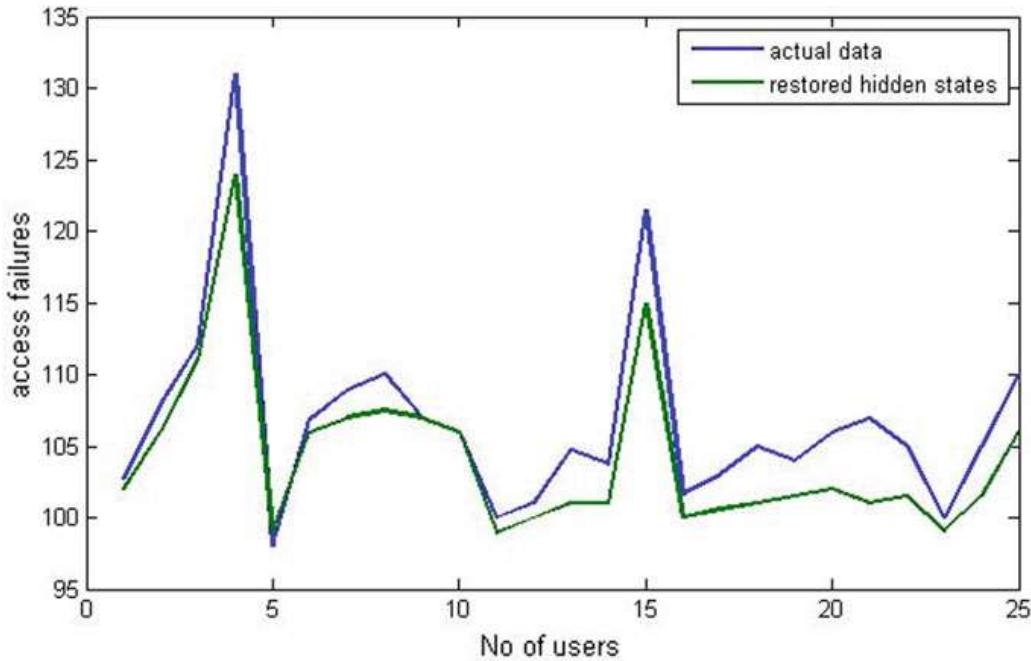


Fig. 2 Most likely sequence for WSDG1

This procedure has been followed for other 19 groups; the BIC values of the dataset from all models are shown in Table 2.

In Table 2, one can conclude that HMM model has low BIC value for all group of WS dataset, except for WSDG7 group, GLS model fits best for that. This indicates the HMM has the best predictive validity model than other NHPP models.

Figure 3 compares the BIC values of all models for all 20 groups. There it can be observed that GO and PLP model almost exactly same curve with a multiple of some factor. The HMM model is almost same for all groups, which is better than other models. For WSDG7, GLS model gives better result than the HMM.

The study concludes the following result:

- Always it is a crucial task to identify number of hidden states for any dataset. The study finds the number of hidden states for this dataset is 5, i.e., $M = 5$. Although it is quite large number, but it closely observes the failure rate
- The predictive validity of HMM good, which predicts the time of future failure
- In this dataset, the restoration detected for hidden states is homogeneous which guides as there should be some rectification in the WS.

5 Conclusion

Web services are the fundamental elements of SOC and cloud computing which helps in the rapid development of applications in heterogeneous environments. It cuts down the coupling among the software modules and enforce on the reusability

Table 2 Model comparison for WS dataset through BIC criterion

Data set/models	HMM	GO	GLS	PLP	MG
WSDG1	7748.35	10,906.12	14,465.42	10,214.39	12,041.11
WSDG2	7595.10	11,093.45	14,966.27	10,372.51	11,613.57
WSDG3	7559.30	11,449.46	16,065.65	10,672.57	10,843.38
WSDG4	7628.83	10,123.93	12,563.33	9551.47	14,001.23
WSDG5	7545.28	9568.46	10,591.90	9076.75	15,640.61
WSDG6	7551.73	11,382.31	10,599.57	10,614.56	11,073.93
WSDG7	7558.33	9217.64	6956.82	8775.28	16,775.83
WSDG8	7577.42	11,867.89	19,139.83	11,025.87	9874.62
WSDG9	7762.00	12,044.31	17,833.34	11,172.75	9598.75
WSDG10	7779.57	9516.64	15,986.00	9034.34	15,653.74
WSDG11	7574.57	11,189.98	22,428.24	10,454.55	11,374.74
WSDG12	7628.46	11,800.15	24,940.68	10,967.33	10,112.47
WSDG13	7536.35	8956.16	14,474.26	8552.39	17,490.10
WSDG14	7600.85	10,364.96	14,932.06	9756.70	13,356.75
WSDG15	7595.62	8789.31	11,146.39	8408.41	18,072.95
WSDG16	7512.30	9532.25	14,668.71	9043.70	15,908.89
WSDG17	7551.59	10,690.91	16,932.87	10,033.75	12,483.19
WSDG18	7621.94	11,537.39	19,719.32	10,747.58	10,595.21
WSDG19	7786.97	9479.40	9058.65	9002.91	15,731.89
WSDG20	7578.12	11,967.05	14,429.07	11,108.93	9683.22

of WS. Reliability of SOC frameworks depends upon the remote WS and furthermore on the Internet affiliations. However, these WS are publicly available and regularly invoked by service users. It is a difficult task to find the suitable model which can be used to identify reliability of these kinds of systems.

In this paper, to predict the failure rate of WS HMM, GO, GLS, PLP, and MG models are used. Real-time dataset is used by forming 20 different groups of similar functionality of WS. The BIC value for each group is calculated. Finally, the hidden Markov model approach gives a new modeling method for WS. HMM is the best fit model to identify the WS reliability compared to other NHPP models. A conceivable expansion of this methodology is to considering the times between failures without a constant rate.

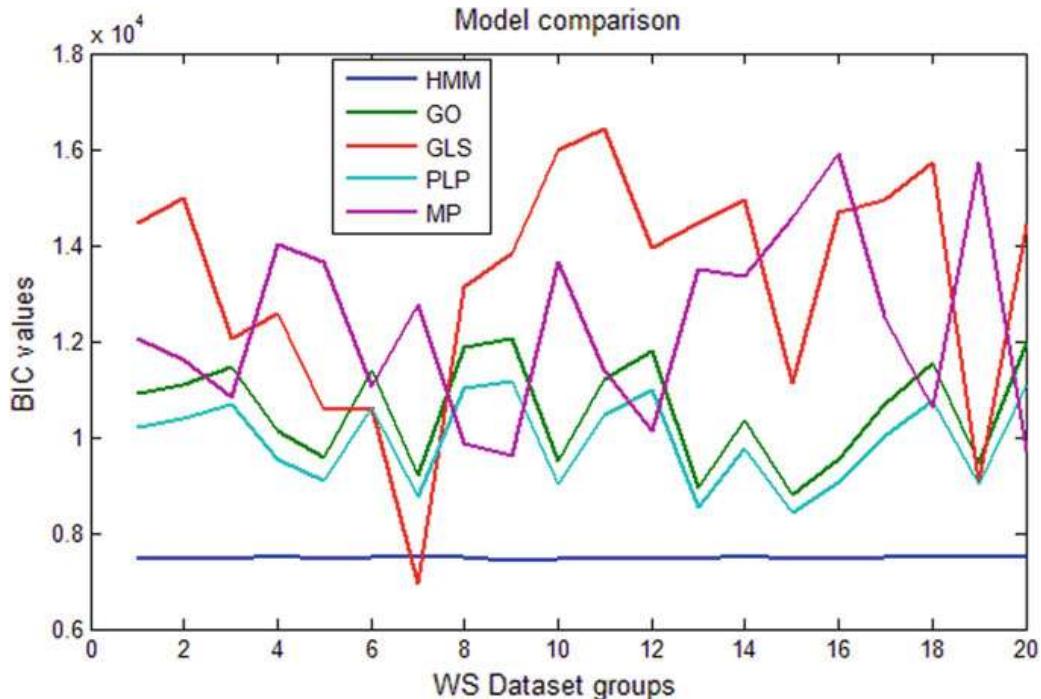


Fig. 3 BIC values comparison for all models

References

1. Durand, J.-B., Gaudoin, O.: Software reliability modelling and prediction with hidden Markov chains. *Stat. Model.* **5**(1), 75–93 (2005)
2. Chen, Y., Singpurwalla, N.D.: Unification of software reliability models by self-exciting point processes. *Adv. Appl. Probab.* **29**, 337–352 (1997)
3. Cai, Z., et al.: Method on integrated reliability assessment of test data based on Duane model. In: 2014 International Conference on Reliability, Maintainability and Safety (ICRMS). IEEE (2014)
4. Ohishi, K., Okamura, H., Dohi, T.: Gompertz software reliability model: estimation algorithm and empirical validation. *J. Syst. Softw.* **82**(3), 535–543 (2009)
5. Boland, P.J., Singh, H.: A birth-process approach to Moranda's geometric software-reliability model. *IEEE Trans. Reliab.* **52**(2), 168–174 (2003)
6. Gaudoin, O., Lavergne, C., Soler, J.-L.: A generalized geometric de-eutrophication software-reliability model. *IEEE Trans. Reliab.* **43**(4), 536–541 (1994)
7. Honamore, S., Rath, S.K.: A web service reliability prediction using HMM and fuzzy logic models. *Procedia Comput. Sci.* **93**, 886–892 (2016)
8. Khreich, W., et al.: A survey of techniques for incremental learning of HMM parameters. *Inf. Sci.* **197**, 105–130 (2012)
9. Moon, T.K.: The expectation-maximization algorithm. *IEEE Sig. Process. Mag.* **13**(6), 47–60 (1996)
10. Oudelha, M., Ainon, R.N.: HMM parameters estimation using hybrid Baum-Welch genetic algorithm. In: 2010 International Symposium on Information Technology, vol. 2. IEEE (2010)
11. Chang, I., Kim, S.W.: Modelling for identifying accident-prone spots: Bayesian approach with a Poisson mixture model. *KSCE J. Civ. Eng.* **16**(3), 441–449 (2012)
12. Zheng, Z., Lyu, M.R.: Personalized reliability prediction of web services. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **22**(2), 1–25 (2013)

Optimal Contrast and Size-Invariant Recursive VCS Using Perfect Reconstruction of White Pixels



T. E. Jisha and Thomas Monoth

Abstract Visual cryptography is an image secret sharing technique which encrypts the image into n shares and reconstructs the image using k shares from n shares by human visual system. Recursive visual cryptography scheme (RVCS) again encrypts the encrypted shares into sub-shares using recursion. In this paper, we proposed a new model for size-invariant RVCS based on Perfect Reconstruction of White Pixels (PRWP) which enhances the contrast. The size-invariant RVCS based on random basis column pixel expansion with PRWP has been demonstrated based on experimental investigations. We have also reviewed the shortcomings of the existing models and made an experimental analysis between the previous models and the proposed model. From the analysis, we proved that the proposed method enhances the contrast and maintains both security and reliability.

Keywords Recursive visual cryptography scheme · Pixels · Contrast · Security · Random basis

1 Introduction

Visual cryptography scheme (VCS) is a technique to encode a surreptitious image into n shares, and the overlapping of k shares can decode the furtive image which was introduced by Naor and Shamir in 1994. Stacking of less than k shares cannot reveal the image [1, 2]. The key feature of the VCS is that decryption can be done without any computation and can be performed by human eyes. This makes possible for anyone to use the scheme without any familiarity of cryptography. Due to these

T. E. Jisha (✉)

Department of Information Technology, Kannur University, Kannur, Kerala 670567, India
e-mail: jishatevinoy@gmail.com

T. Monoth

Department of Computer Science, Mary Matha Arts & Science College, Kannur University,
Mananthavady, Wayanad, Kerala 670645, India
e-mail: tmonoth@yahoo.com

features, VCS takes an unavoidable role in image security. The contrast of the image, pixel expansion and security are the three focal issues in the research community of visual cryptography. The aforementioned parameters in VCS need extra concern [3–6].

The commencement of many size-invariant VCS defeats the problem of pixel expansion [7, 8]. The size of the secret image, encoded transparencies and decoded image is identical since the pixel expansion is one in this scheme. The issues allied to security and contrast remain in the size-invariant VCS [9–17]. RVCS enhances the security and reliability of the system in which the image can be encrypted into n shares and each encrypted shares can again be encoded into p sub-shares in a recursive manner [18].

The vital aim of the suggested method is to construct a size-invariant RVCS which advances the contrast of the overlapped image and retains security and reliability of the system. We exploited random basis column pixel expansion method and introduced a new RVCS with perfect reconstruction of white pixels in place of black pixels [18, 19]. The forthcoming sectors of the paper come after in order: Part 2 describes RVCSSs. Part 3 demonstrates the proposed model, the experimental results and analysis of various schemes. The final section draws the conclusion.

2 Recursive Visual Cryptography Scheme

In conservative VCS, the image encryption can be performed in a single level. However, in RVCS, the encryption can be made in multiple levels. Here, the image can be encoded into shares, and then, these encrypted shares are again encrypted into sub-shares recursively [18].

2.1 The RVCS Model

The RVCS is illustrated in Fig. 1 using a tree structure based on (2, 2) VCS with three levels of encryption. The encryption can be performed recursively in multiple levels, since the security and reliability of the system are remarkably enhanced. In (2, 2) RVCS, there are two shares in the first level, four shares in the next level, eight shares in the third level and so on. The image can be reconstructed by overlapping of these shares by different combinations as mentioned below:

$$SI = Sh_1 + Sh_2$$

$$SI = Sh_1 + Sh_{21} + Sh_{22}$$

$$SI = Sh_2 + Sh_{11} + Sh_{12}$$

$$SI = Sh_{11} + Sh_{12} + Sh_{21} + Sh_{22}$$

$$SI = Sh_{11} + Sh_{12} + Sh_{211} + Sh_{212} + Sh_{221} + Sh_{222}$$

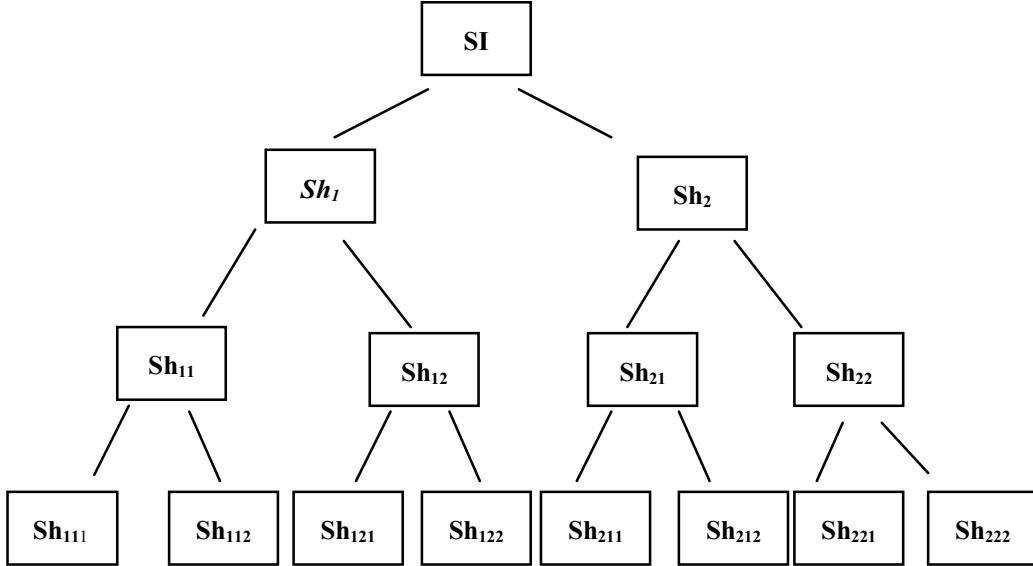


Fig. 1 (2, 2) RVCS with three levels of encryption

$$SI = Sh_{21} + Sh_{22} + Sh_{111} + Sh_{112} + Sh_{121} + Sh_{122}$$

$$SI = Sh_1 + Sh_{211} + Sh_{212} + Sh_{221} + Sh_{222}$$

$$SI = Sh_2 + Sh_{111} + Sh_{112} + Sh_{121} + Sh_{122}$$

$$SI = Sh_{111} + Sh_{112} + Sh_{121} + Sh_{122} + Sh_{211} + Sh_{212} + Sh_{221} + Sh_{222}$$

Here, SI is the secret image, Sh₁ and Sh₂ are shares in the first level, Sh₁₁, Sh₁₂, Sh₂₁ and Sh₂₂ are the shares in the second level and Sh₁₁₁, Sh₁₁₂, Sh₁₂₁, Sh₁₂₂, Sh₂₁₁, Sh₂₁₂, Sh₂₂₁ and Sh₂₂₂ are the shares in the third level of encryption. Even if any few of the shares are lost during transmission, the image can be reconstructed in any of the combinations mentioned above. This leads to greater reliability [18, 19].

2.2 Working Example of RVCS

The size-invariant RVCS based on random basis column pixel expansion with perfect reconstruction of black pixels (PRBP) can be explained with two $n \times m$ binary basis matrices, B^w and B^b .

The basis matrices are:

$$B^w = \begin{bmatrix} 1010 \\ 1010 \end{bmatrix}, \quad B^b = \begin{bmatrix} 1010 \\ 0101 \end{bmatrix}$$

In the encryption of white pixels, one of the columns is arbitrarily selected from B^w and for black pixels from B^b . For example, the vector chosen from B^b is:

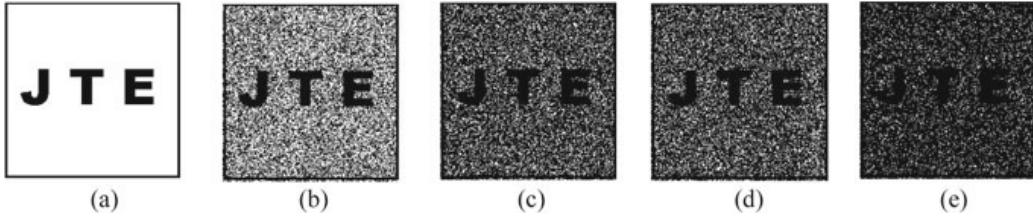


Fig. 2 **a** SI, **b** $\text{Sh}_1 + \text{Sh}_2$, **c** $\text{Sh}_1 + \text{Sh}_{21} + \text{Sh}_{22}$, **d** $\text{Sh}_2 + \text{Sh}_{11} + \text{Sh}_{12}$, **e** $\text{Sh}_{11} + \text{Sh}_{12} + \text{Sh}_{21} + \text{Sh}_{22}$

$$V = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

An example of (2, 2) size-invariant RVCS based on random basis column pixel expansion with two levels is shown in Fig. 2. The overlapping of different combinations of the shares and sub-shares can reveal the secret image. From Fig. 2, we noticed that the contrast of the decrypted image was degraded when the second-level encrypted shares were overlapped. Here, the RVCS is performed with PRBP. In this scheme, 50% of the white pixels are lost in decryption. To overcome this degradation, we proposed a new RVCS method with PRWP as a replacement for PRBP. In most of the natural images, the ratio of white pixel is superior to the black pixel. Hence, the perfect reconstruction of white pixels leads to higher contrast [18, 20, 21].

3 The Proposed Model: Size-Invariant RVCS with PRWP

The proposed method retains security and reliability and enhances contrast of the decrypted image. In this model, we used the enhanced security and reliability features of the size-invariant RVCS and enhanced contrast feature of the VCS with PRWP. The encryption of SI can be encoded into multiple levels.

In the first level, the SI is encoded into n shares.

$$\text{SI} = \text{Sh}_1, \text{Sh}_2, \dots, \text{Sh}_n$$

The image can be decrypted by any k shares using the Eq. (1).

$$\text{SI} = \sum_{i=1}^k \text{Sh}_i \quad (1)$$

Further, each share can again be encrypted into p sub-shares recursively.

$$\begin{aligned} \text{Sh}_1 &= \text{Sh}_{11}, \text{Sh}_{12}, \dots, \text{Sh}_{1p} \\ \text{Sh}_2 &= \text{Sh}_{21}, \text{Sh}_{22}, \dots, \text{Sh}_{2p} \end{aligned}$$

$$\begin{aligned} & \vdots \\ \text{Sh}_n &= \text{Sh}_{n1}, \text{Sh}_{n2}, \dots, \text{Sh}_{np} \end{aligned}$$

The shares $\text{Sh}_1, \dots, \text{Sh}_n$ can be decrypted by any k sub-shares:

$$\text{Sh}_1 = \sum_{(i=1)}^k \text{Sh}_{1i}$$

$$\text{Sh}_2 = \sum_{(i=1)}^k \text{Sh}_{2i}$$

\vdots

$$\text{Sh}_n = \sum_{(i=1)}^k \text{Sh}_{ni}$$

Generally, it can be stated as in Eq. (2).

$$\text{Sh}_i = \sum_{j=1}^k \text{Sh}_{ij}, \quad \text{where } 1 \leq i \leq n \quad (2)$$

Here, we constructed a $(2, 2)$ RVCS based on PRWP with two levels of encryption. In the proposed method, the image can be represented with two $n \times m$ binary basis matrices,

$$\text{BW}^b = \begin{bmatrix} 1010 \\ 1010 \end{bmatrix}, \quad \text{BW}^w = \begin{bmatrix} 1010 \\ 0101 \end{bmatrix}$$

Here, 1 denotes white and 0 denotes black. To encrypt the white pixels, one of the column vectors is arbitrarily selected from the BW^w and for black pixels from BW^b . For example, the vectors;

$$V = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ for white and } V = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ for black}$$

The pixel selection in RVCS with PRWP is just opposite to the pixel selection in RVCS with PRBP [18–21].

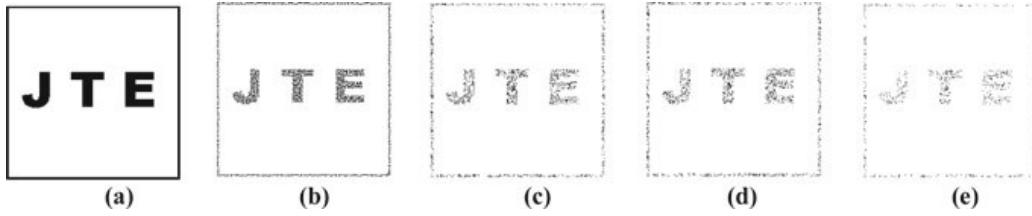


Fig. 3 **a** SI, **b** $Sh_1 + Sh_2$, **c** $Sh_1 + Sh_{21} + Sh_{22}$, **d** $Sh_2 + Sh_{11} + Sh_{12}$, **e** $Sh_{11} + Sh_{12} + Sh_{21} + Sh_{22}$

3.1 Experimental Results

The experiments were explained with a $(2, 2)$ RVCS based on PRWP with two levels of encryption. For instance, an image with $m \times n$ size (here, m and n are 200) can be encrypted into n shares. Each encoded share can again be encrypted into p sub-shares. Overlapping of different combinations of shares and sub-shares reveals the secret image. Different decrypted images from the investigational outcomes are shown in Fig. 3. By comparing Figs. 2 and 3, we have observed that the visual perception of the decrypted images in the proposed model is higher than the RVCS with PRBP.

The major advantages of the proposed model are:

- The scheme is size-invariant recursive VCS.
- It ensures both security and reliability.
- The scheme enhances the contrast of reconstructed image.
- The scheme is based on PRWP instead of perfect reconstruction of black pixels.

3.2 Analysis of Experimental Results

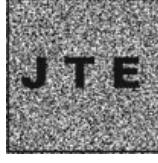
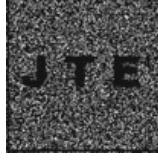
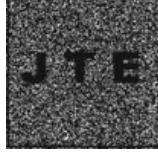
We examined the reliability, security and contrast of the conventional size-invariant RVCS and the proposed size-invariant RVCS with PRWP on a range of images. The experimental analysis is depicted on Table 1 in respect of PSNR (peak signal-to-noise ratio) values of the existing RVCS and the proposed model. The PSNR value refers to the visual quality dimension between the secret image and the decrypted image. Higher PSNR value indicates better contrast of the decrypted image. From Table 1, we proved that the PSNR value of the decrypted images in the proposed model is greater than the existing RVCS. Also, we ascertained that our scheme retained the security and reliability on par with the conventional schemes. The output of the decrypted images of these two schemes is portrayed on Table 2. From Table 2, we observed that the proposed RVCS model has greater contrast than the existing RVCSs.

Figure 4 illustrates the graphical representation of the PSNR values of the conservative scheme and the proposed scheme. From this figure, we corroborated that the value of PSNR in the existing model is tremendously low compared to the proposed RVCS with PRWP. Thus, we stated that our scheme provides optimal contrast.

Table 1 Analysis of conventional RVCS and the proposed model

Overlapping of shares and sub-shares	Conventional RVCS				Proposed RVCS with PRWP			
	White pixels	Black pixels	Total pixels	PSNR value	White pixels	Black pixels	Total pixels	PSNR value
Sh ₁ + Sh ₂	17,647	22,353	40,000	51.68	37,688	2312	40,000	60.68
Sh ₁ + Sh ₂₁ + Sh ₂₂	8751	31,249	40,000	49.92	38,826	1174	40,000	58.90
Sh ₂ + Sh ₁₁ + Sh ₁₂	8727	31,273	40,000	49.92	38,852	1148	40,000	58.86
Sh ₁₁ + Sh ₁₂ +Sh ₂₁ + Sh ₂₂	4304	35,696	40,000	49.25	39,405	595	40,000	58.20

Table 2 Decrypted images
of conventional and proposed
RVCS

Overlapping of shares and sub-shares	Conventional RVCS	RVCS with PRWP
Sh ₁ + Sh ₂		
Sh ₁ + Sh ₂₁ + Sh ₂₂		
Sh ₂ + Sh ₁₁ + Sh ₁₂		
Sh ₁₁ + Sh ₁₂ + Sh ₂₁ + Sh ₂₂		

4 Conclusion

This paper proposes a new scheme for size-invariant RVCS which provides optimal contrast of the reconstructed image and preserves security and high reliability of the scheme. Here, we constructed a recursive VCS with perfect reconstruction of white pixels as a substitute of black pixels. The existing methods and proposed method

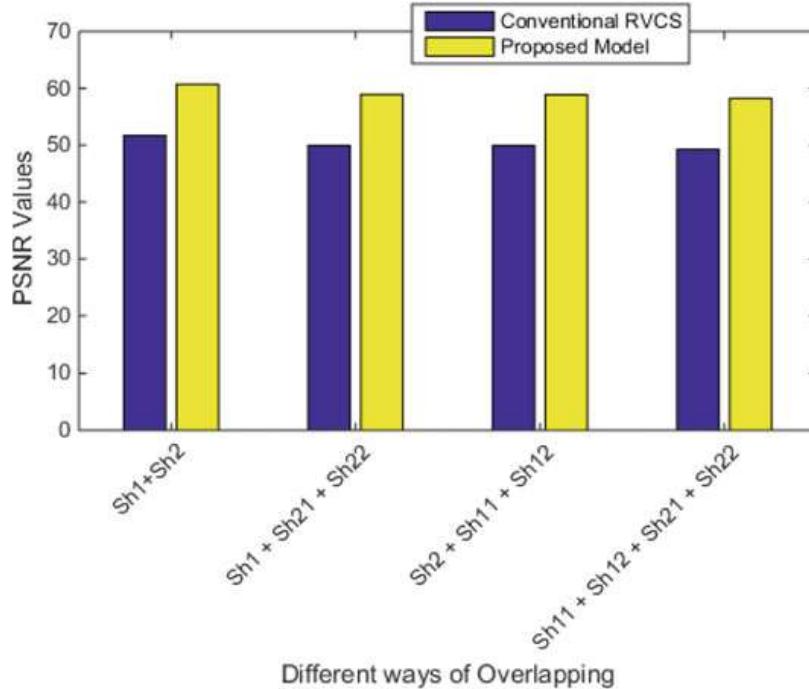


Fig. 4 PSNR values of conventional scheme and the proposed scheme

are demonstrated with investigational results and analyzed by tables and graphs. Our future research will concentrate on construction of RVCS with enhanced contrast, reliability and security for color images.

References

1. Naor, M., Shamir, A.: Visual Cryptography. Advances in Cryptology-Eurocrypt'94, LNCS 950, pp. 1–12 (1995)
2. Pandey, D., Kumar, A., Singh, Y.: Feature and Future of Visual Cryptography Based Schemes, Quality, Reliability, Security and Robustness in Heterogeneous Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 115, pp. 816–830. Springer (2013). https://doi.org/10.1007/978-3-642-37949-9_71
3. Monoth, T., Babu Anto, P.: Analysis and Design of Tamperproof and Contrast-Enhanced Secret Sharing Based on Visual Cryptography Schemes, Ph.D. thesis, Kannur University, Kerala, India (2012). <http://shodhganga.inflibnet.ac.in>
4. Ateniese, G., Blundo, C., De Santis, A., Stinson, D.R.: Constructions and bounds for visual cryptography. In: Meyer, F., Monien, B. (eds.) Automata, Languages and Programming. ICALP 1996. Lecture Notes in Computer Science, Vol. 1099. Springer, Berlin (1996)
5. Weir, J., Yan, W.A.: Comprehensive study of visual cryptography. In: Shi, Y.Q. (eds.) Transactions on Data Hiding and Multimedia Security V. Lecture Notes in Computer Science, vol. 6010. Springer, Berlin (2010)
6. Jisha, T.E., Monoth, T.: Research advances in black and white visual cryptography schemes. Int. J. Adv. Intell. Syst. Comput. (2019). <http://www.springer.com/series/11156>. (Accepted)

7. Lin, T.H., Shiao, N.S., Chen, H.H., Tsai, C.S.: A new non-expansion visual cryptography scheme with high quality of recovered image. In: IET International Conference on Frontier Computing, Theory, Technologies and Applications. IEEE Xplore (2010) <https://doi.org/10.1049/cp.2010.0571>
8. Huang, Y.-J., Chang, J.-D.: Non-expanded visual cryptography scheme with authentication. In: IEEE 2nd International Symposium on Next-Generation Electronics (ISNE). IEEE (2013). <https://doi.org/10.1109/isne.2013.6512319>
9. Chow, Y.W., Susilo, W., Wong, D.S.: Enhancing the perceived visual quality of a size invariant visual cryptography scheme. In: Information and Communications Security. Lecture Notes in Computer Science, vol. 7618. Springer, Berlin (2012)
10. Ito, R., Kuwakado, H., Thanka, H.: Image size invariant visual cryptography. IEICE Trans. Fundam. **E82-A**(10) (1999)
11. Liu, F., Guo, T., Wu, C.K., Qian, L.: Improving the visual quality of size invariant visual cryptography scheme. *J. Vis. Commun. Image Represent.* **23**(2), 331–342 (2012). <https://doi.org/10.1016/j.jvcir.2011.11.003>. (Elsevier)
12. Chen, Y.-F., Chan, Y.-K., Huang, C.-C., Tsai, M.-H., Chu, Y.-P.: A multiple-level visual secret-sharing scheme without image size expansion. *Inf. Sci.* **177**(21), 4696–4710 (2007)
13. Yan, B., Wang, Y.F., Song, L.Y., et al.: Size-invariant extended visual cryptography with embedded watermark based on error diffusion. *Multimedia Tools Appl.* **75**, 11157 (2016). <https://doi.org/10.1007/s11042-015-2838-4>
14. Yan, B., Xiang, Y., Hua, G.: Improving the visual quality of size-invariant visual cryptography for grayscale images. An analysis-by-synthesis (AbS) approach. *IEEE Trans. Image Process.* **28**(2) (2019). <https://doi.org/10.1109/tip.2018.2874378>
15. Yan, B., Wang, Y.-F., Song, L.-Y., Yang, H.-M.: Size-invariant extended visual cryptography with embedded watermark based on error diffusion. *Multimedia Tools Appl.* **75**(18), 11157–11180 (2016) <https://doi.org/10.1007/s11042-015-2838-4>
16. Ou, D., Sun, W., Xiaotian, W.: Non-expansive XOR-based visual cryptography scheme with meaningful shares. *Sig. Process.* **108**, 604–621 (2015). <https://doi.org/10.1016/j.sigpro.2014.10.011>. (Elsevier)
17. Sharma, R., Agrawal, N.K., Khare, A., Pal, A.K.: An improved size invariant (n, n) extended visual cryptography scheme. *Int. J. Bus. Data Commun. Netw.* **12**(2) (2016)
18. Monoth, T., Babu, A.P.: Recursive visual cryptography using random basis column pixel expansion. In: 10th International Conference on Information Technology (ICIT 2007), Orissa, IEEE Xplore (2007), pp. 41–43. <https://doi.org/10.1109/icit.2007.32>
19. Monoth, T., Babu Anto, P.: Contrast-enhanced visual cryptography schemes based on perfect reconstruction of white pixels and additional basis matrix. In: Computational Intelligence, Cyber Security and Computational Models, Advances in Intelligent Systems and Computing, vol. 412, pp. 361–368. Springer, Singapore (2016)
20. Monoth, T.: Contrast-enhanced recursive visual cryptography scheme based on additional basis matrices. In: Satapathy, S., Bhateja, V., Das, S. (eds.) Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies, vol. 105. Springer, Singapore (2019)
21. Mohan, A., Binu, V.P.: Quality improvement in Color Extended Visual Cryptography using ABM and PRWP. In: International Conference on Data Mining and Advanced Computing (SAPIENCE). IEEE Xplore (2016). <https://doi.org/10.1109/sapience.2016.7684159>

Performance Analysis of Brain Imaging Using Enriched CGLS and MRNSD in Microwave Tomography



N. Nithya , R. Sivani Priya , and M. S. K. Manikandan

Abstract A trade-off between computational time complexity and more number of sensing antennas is a hurdle in high-resolution microwave tomography image reconstruction process. This paper deliberates the efficacy of Krylov subspace-based gradient regularization methods such as Enriched Conjugate Gradient Least Square (Enriched CGLS) and Modified Residual Norm Steepest Descent (MRNSD) method imposed in the reconstruction algorithm which effectively handles the above impediment. The performance of the proposed methods has been tested with varying the number of antennas, operating frequency and the levels of Gaussian noise in brain phantom and mean square error (MSE) and number of iterations are the parameters used for the analysis. MRNSD method has proved its betterment in all the criteria. It achieves 77% accuracy within five iterations.

Keywords Microwave tomography · Regularization · Enriched CGLS · Brain imaging · MRNSD

1 Introduction

Microwave tomography (MWT) is an emerging imaging technology, and it has been widely used in applications like brain imaging [1] and breast imaging [2] because of its benefits like non-ionizing and portability. The image reconstruction algorithm is an important task to construct a map of the structure of internal tissues in tomography. It is known as an inverse scattering problem. The Born iterative method (BIM)

N. Nithya · R. Sivani Priya () · M. S. K. Manikandan
Department of Electronics and Communication Engineering, Thiagarajar College of Engineering,
Madurai, Tamil Nadu, India
e-mail: sivaniriyal@gmail.com

N. Nithya
e-mail: er.nithyacse@gmail.com

M. S. K. Manikandan
e-mail: manimsk@tce.edu

[3] algorithm, the modified BIM named distorted BIM (DBIM) [4] and gradient Gauss–Newton method (GNM) [5] are used in MWT. Generally, this problem is caused by ill-posedness which means the error differences in desired output value, and the resultant is unavoidable. So, every reconstruction algorithm is imposed by regularization method as one of the steps to reduce ill-posed problem. The error in unknown internal tissues (x) is raised due to the measurement devices. Regularization methods are used to solve this problem and to extract the x accurately. Good resolution and less time complexity are the challenges in MWT image reconstruction algorithms [6]. The number of antennas and its operating frequency in MWT has an impact on construction of the coefficient matrix and image with good resolution, but it leads to high time complexity. Trade-off between time complexity and resolution occurs by reducing the number of antennas. Selection of appropriate regularization method effectively handles these situations and produces a reconstructed image with better resolution in less convergence time and also improves the overall time complexity of the image reconstruction algorithm. In this paper, the performance of the proposed regularization methods is analyzed with the variations in the coefficient matrix.

1.1 Related Works

There are various regularization methods in Krylov subspace method [7] like conjugate gradient method (CG) [8], conjugate gradient least square method (CGLS) and sparsity methods [9] like two-step iterative shrinkage/thresholding (TwIST) [2] used in MWT. On analysis, these regularization methods are the most commonly used methods in the construction of structure of unknown tissues. The compressive sensing (CS) method is implemented in stroke detection at 0.7 GHz with the variation in the number of antennas from 32 to 64 in [10], and also, another paper in CS method is implemented by using 8, 12, 36 number of antennas and compared each result at 1 GHz. The inexact Newton method is used to reconstruct the unknown at 0.6 GHz with 30 numbers of antennas [11]. The performance of CGLS method and L^p Banach space method are compared using 20 antennas at 1 GHz in the detection of brain stroke [12]. The performance of different regularization methods like Banach subspace method, Krylov subspace methods and sparsity method is analyzed in different cases. Among these regularization methods, this paper proposed modified residual norm steepest descent (MRNSD) [13] and enriched conjugate gradient least square (Enriched CGLS) [14] which is a branch of Krylov subspace regularization. The efficiency of the proposed regularization methods is analyzed against variations in the coefficient matrix by different measurement configurations of MWT for the brain data model. The analysis is made on three criteria by varying the frequencies, number of antennas and also the noise levels. The mean square error (MSE) and number of iteration are the parameters used to examine the performance.

In this paper, the formulation of the proposed methods is discussed in Sect. 2. Section 3 provides the importance of regularization methods in microwave reconstruction of brain phantom. The numerical results aimed at analyzing the work of

regularization methods in brain phantom are reported in Sect. 4. Finally, Sect. 5 explains the concluded information on chosen three criteria.

2 Formulation of the Proposed Forward Process

The forward process uses a circular measurement system. The N-number of measurement antennas is arranged in the boundary of the circular region. The pictorial representation of the measurement system is shown in Fig. 1. The heterogeneous 2D slice of head phantom is characterized by the frequency-dependent complex permittivity as,

$$\varepsilon(r) = \varepsilon'(r) - j\varepsilon''(r) \quad (1)$$

where the term $\varepsilon'(r)$ denotes the dielectric constant of the material and $\varepsilon''(r)$ denotes the conductivity of the material. It is located at the center of the measurement domain.

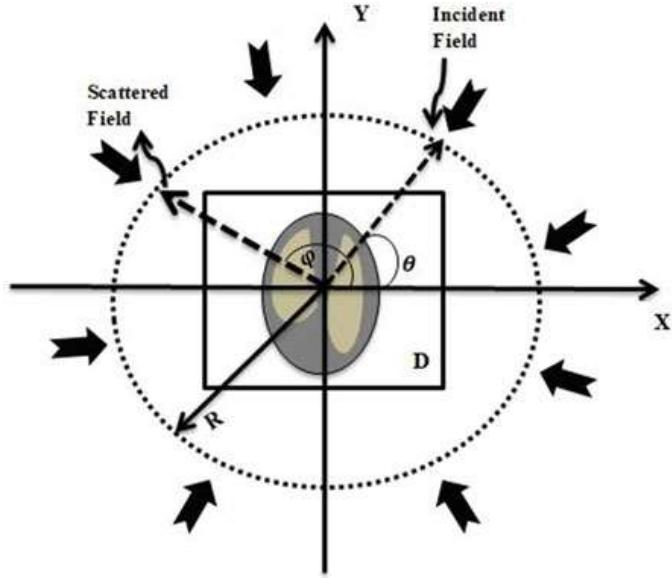
The objective function of the MWT is stated below,

$$E_{\text{tot}} = E_{\text{inc}} + E_{\text{sca}} \quad (2)$$

where E_{tot} is the total field, E_{inc} is the incident field and E_{sca} is the scattered field. The incident field ($E_{\text{inc}} = (x \cos \theta_i + y \sin \theta_i)$) is applied in the form of point source to the domain. The incident and scattered field is measured as

$$E_{\text{sca}} = (\omega^2 \mu / 4) \int J_z(d') H_0^2(k|l - l'|) dl' \quad (3)$$

Fig. 1 2D microwave tomography imaging system



where $k = \omega\sqrt{\mu\varepsilon}$ is the wave number and $\eta = \sqrt{\mu/\varepsilon}$ is the intrinsic impedance in the free space, J_z is the current density and $H_0^{(2)} = 1 - j\frac{2}{\pi} \ln \frac{\gamma}{2}$; $\gamma = 1.781$ is the Henkel function of the second kind (zero order). θ_i is the angle of incidence from different directions. The angles in which the antennas are positioned are calculated as (4) where θ_i is the angle of incidence from different directions.

$$\theta_i = (i - 1)\frac{2\pi}{N}; i = 1, 2 \dots, N \quad (4)$$

In the initial step in BIM iteration, the incident field is taken as the total field. It is called born approximation theory. The scattered field is updated in each iteration of BIM until the difference between the estimated and calculated scattered field meets the minimum. This integral system Eq. (3) can be solved by using method of moments which is generally used to reduce a functional equation to a matrix equation. To solve this functional equation, Eq. (3) is formulated mathematically as a linear system of equation ($A \cdot x = Y$). A represents the incident field coefficient matrix, and the unknown object denoted as x and b is the scattered field. This process of finding x is called the inverse problem.

3 Importance of Enriched CGLS and MRNSD Regularization Methods

The solution for a linear system is determined by finding $x = A^{-1}B$. If A is large, the inverse of A would be inefficient and increases complexity. Medical imaging application, such as image reconstruction, is modeled as an integral equation, which after discretization results a large-scale linear system. Here, the A matrix is defined as point spread functions of total electric field of each pixel that are dielectric properties of a tissue. The dimension of A is $M \times N$ and $M \ll N$. M depends on number of antennas used for transmitting and receiving the incident field and scattered field, respectively. The N is the total number of pixels of the object. The dimensionality changes in matrix A rased the ill-conditioned problem which means small changes in A may create high deviation in the solution of x . This large factorization of A is unable to be realized the appropriate solution of unknown x . The cost function of the proposed problem is,

$$f(x) = 1/2(x^T A^T A x - x^T A^T b) \quad (5)$$

$$x_k := x_{k-1} + \alpha_{k-1} \cdot p_{k-1} \quad (6)$$

x_k is the solution which is computed by the α the step length and the direction p . The value of x_k is determined in iterative manner, and α and p were updated in each iteration. The proposed methods Enriched CGLS and MRNSD were differed by the

computing the α and p . The proposed work states that the two Krylov subspace methods such as CGNR or Enriched CGLS and MRNSD methods worked well in solving ill-determined rank or ill-conditioned A matrix. It also estimates the good solution in noise distortion. The Enriched CGLS showed its efficiency in severely ill-conditioned matrix A . The minimum x_k is computed by a sequence of linear searches along the $A^T A$ conjugate search direction in every iteration. The direction p is associated with residual(r) of the previous direction.

$$p_k := r_k + \beta_{k-1} p_{k-1} \quad (7)$$

It is the stopping criteria for considerably reduced number of iterations. It is less sensitive in the error (e) in b ($b + e$). In MRNSD method, the preconditioner is accelerating the convergence of iteration and produces significantly accurate solution. It reduces the computation time for the entire reconstruction algorithm. It is efficient in ill-conditioned A matrix. The value of x is converged by finding gradient of $\varphi(x)$ as,

$$\text{grad } \varphi(x) = X A^T (Ax - b), \text{ where } X = \text{diag}(x) \quad (8)$$

$p_{k-1} = X_k A^T (Ax_{k-1} - b)$ denotes the step in the direction of negative gradient in MRNSD. Here, the preconditioner X is step dependent which helps fast convergence in noise distortion and ill-conditioned A . The non-negativity in the solution is the advancement of this method.

4 Numerical Results and Discussion

The numerical assessment is made on the proposed methods such as Enriched CGLS and MRNSD in the aspect of finding the applicability of the proposed methods in three constraints in the imaging system measurement configurations for brain data model. The measurement is made by changing the number of antennas, the operating frequencies and also by adding different Gaussian noise levels. The analysis is made on the 2D brain data model. The performance analysis of Enriched CGLS and MRNSD in the above mentioned scenarios is estimated by two factors such as MSE and the iteration count are used to analyze the applicability of the proposed methods. The applicability of the algorithms on above criteria is concluded based on the numerical results, and it is compared with the existing CGLS method.

4.1 Simulation Setup and Dataset

The system is configured as the antennas are placed in a circular fashion with a radius of 11.6 cm from the center. The brain phantom data is taken from [15]. The

Table 1 Dielectric properties of brain data model

Tissue name	Gray matter	White matter	Skull	CSF	Background medium
Dielectric value	$50 + j18$	$40 + j15$	$13 + j2$	$57 + j26$	$40 + 13j$

size of 2D brain phantom is 212×212 and each pixel with 1 mm size. The dielectric properties' brain phantom is shown in Table 1.

4.2 Performance Analysis on Varying Number of Antennas and Frequencies

The ill-determined and the ill-conditioned problems were raised in number of antennas and frequency range used in the measurement configuration. The size of matrix is varied according to the number of antennas. The existing works are realized by the minimum 16 antennas and maximum of 24 antennas. The proposed work was configured by 16, 20, 24 number of antennas, and the frequency is fixed as 1 GHz. The proposed regularization methods were run up to 100 iterations, and the corresponding MSE values are recorded. The resultant values for these considerations are given in Table 2.

In the CGLS algorithm, it is observed that 90% of MSE is acquired in all three cases of antenna considerations. In case of Enriched CGLS, 30% of MSE is achieved for 16, 20, 24 antenna considerations. But, for 24 numbers of antennas, convergence with 30% MSE is obtained within 69 iterations and is stable till 100 iterations. In case of MRNSD, the convergence with 30% of MSE is achieved in seven iterations and is stable in that values till 100 iterations. It is concluded that CGLS, Enriched CGLS and MRNSD provide similar MSE with the changes in the number of antennas. The variation in antenna numbers does not make a huge impact in reducing the ill-posedness problem. Secondly, the measurement is made on considering different frequencies which depicts ill-conditioned problem in A. The frequency range varied from 700 MHz to 1.5 GHz and measurements taken by 24 antennas. In case of CGLS method, the MSE is about 90% in all considered frequencies. In case of Enriched CGLS, the MSE is achieved about 30% for 0.7 GHz within 89 iterations and MSE is

Table 2 Comparison of MSE for 16, 20, 24 antennas with 1 GHz frequency and 0.7, 1 and 1.5 GHz with 24 antennas for 100 iterations

Scenario	CGLS	Enriched CGLS	MRNSD
16 antennas	0.9662	0.3027	0.3267
20 antennas	0.9684	0.3029	0.326
24 antennas	0.9687	0.3030	0.3266
0.7 GHz	0.9531	0.3135	0.3620
1 GHz	0.9681	0.3030	0.3266
1.5 GHz	0.9900	0.6466	0.3489

further reduced at 1 GHz within 69 iterations. This value is stable till 100 iterations. But, at 1.5 GHz, the MSE is obtained about 60% within the first iteration, and then, till 100, the values fluctuate. In case of MRNSD, 30% of MSE is achieved in all cases within 10 iterations. But, at 1.5 GHz, the MSE is increased about 2% than at 1 GHz which is obtained within 12 iterations till 100 iterations. As a result, these algorithms provide better results in the frequency range between 0.7 and 1 GHz. Also, the reduction of MSE in Enriched CGLS at 1 GHz is achieved within 69 iterations which further improve the betterment of the algorithm.

4.3 Performance Analysis on Noise Distortion

To assess the applicability of the proposed Enriched CGLS and MRNSD in the presence of noise Gaussian noise is added in the scattered value (b). The algorithms are analyzed by adding a Gaussian noise in scattered value (b) at the levels as 10, 30, 60% for 100 iterations. The frequency is fixed as 1 GHz and antennas numbers as 24. In case of CGLS, 90% of MSE is obtained in all cases of noise levels. In case of Enriched CGLS, the MSE is about 54%, but in higher noise levels, MSE is maximum. The presence of Gaussian noise provides unstable MSE. In case of MRNSD, 30% MSE is achieved in all noise levels. These results are obtained within 2–5 iterations.

It is concluded that the CGLS algorithm provides similar MSE in all levels of noise. In case of Enriched CGLS, the MSE fluctuates with the increase in the noise. But, in case of MRNSD algorithm, the MSE is minimum and there is only a considerable change in high levels of noise. The working of CGLS, Enriched CGLS and MRNSD in 10, 30 and 60% of noise level is shown in Fig. 2.

On comparing the suitability of the existing and proposed regularization methods, the Enriched CGLS fluctuates more in the noise minimum case but it is observed that MRNSD provides the best results in the noisy environment. The reconstructed image is shown in Fig. 3. The MRNSD works better in all the constraints so that it can produce high resolute image with less time consumption in microwave tomography imaging systems.

5 Conclusion

The performance of Krylov subspace methods such as Enriched CGLS and MRNSD in the microwave tomography imaging of 2D brain phantom has been analyzed. In this paper, the regularization methods such as CGLS, Enriched CGLS and MRNSD are used in scenarios like as ill-determined and ill-conditioned coefficient matrix. The numerical results obtained by varying the number of antennas, range of measurement frequencies and different noise level are analyzed. The applicability of the proposed regularization algorithms in all cases is concluded on the basis of their iteration count and MSE. It has been concluded that the variation in antenna numbers does not make

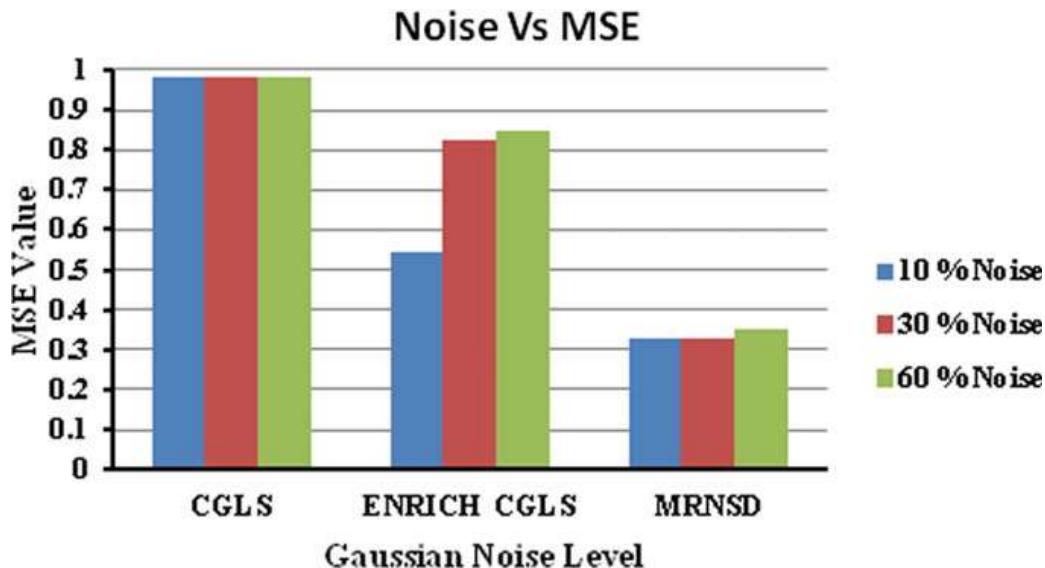


Fig. 2 Comparison of the existing CGLS method with the proposed Enriched CGLS and MRNSD methods at 60, 30 and 10% noise levels

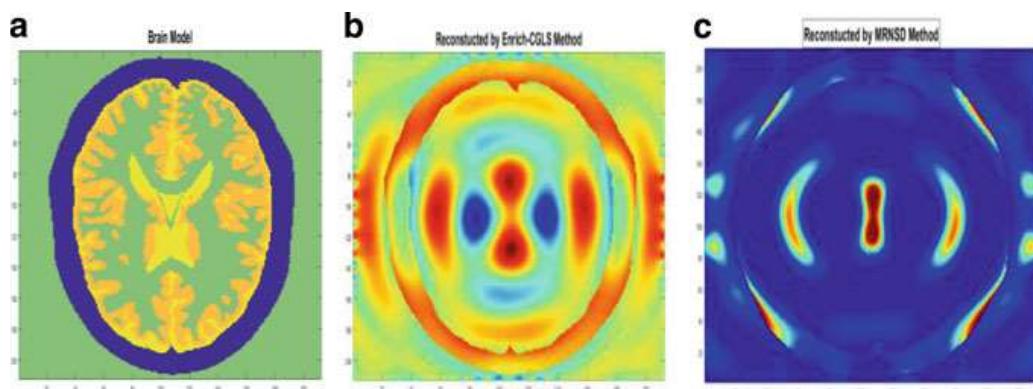


Fig. 3 Reconstructed images of brain **a** input brain model, **b** Enriched CGLS, **c** MRNSD regularization algorithms at 1 GHz

a huge impact in reduction of error in ill-posedness problem. Moreover, the change in frequency is the major factor of consideration in obtaining the image with high resolution.

References

1. Ireland, D., Bialkowski, K., Abbosh, A.: Microwave imaging for brain stroke detection using Born iterative method. *IET Microwaves Antennas Propag.* **7**(11), 909–915 (2013)
2. Mia, Z., Kosmas, P.: Multiple-frequency DBIM-TwIST algorithm for microwave breast imaging. *IEEE Trans. Antennas Propag.* **65**(5), 2507–2516 (2017)

3. Wang, Y.M., Chew, W.C.: An iterative solution of the two-dimensional electromagnetic inverse scattering problem. *Int. J. Imaging Syst. Technol.* **1**(1), 100–108 (1989)
4. Chew, W.C., Wang, Y.M.: Reconstruction of two-dimensional permittivity distribution using the distorted born iterative method. *IEEE Trans. Med. Imaging* **9**(2), 218–225 (1990)
5. Mojabi, P., LoVetri, J., Shafai, L.: A multiplicative regularized Gauss-Newton inversion for shape and location reconstruction. *IEEE Trans. Antennas Propag.* **59**(12), 4790–4802 (2011)
6. Chandra, R., Zhou, H., Balasingham, I., Narayanan, R.M.: On the opportunities and challenges in microwave medical sensing and imaging. *IEEE Trans. Biomed. Eng.* **62**(7), 1667–1682 (2015)
7. Kees, V., Sevink, A.G.J., Herman, G.C.: A preconditioned Krylov subspace method for the solution of least squares problems in inverse scattering. *J. Comput. Phys.* **123**, 330–340 (1995)
8. Estatico, C., Fedeli, A., Pastorino, M., Randazzo, A.: Comparison between conjugate gradient and Landweber based regularization approaches in L^p Banach spaces for microwave tomography. In: 2nd URSI AT-RASC, Gran Canaria (2018)
9. Winters, D.W., Van Veen, B.D., Hagness, S.C.: A sparsity regularization approach to the electromagnetic inverse scattering problem. *IEEE Trans. Antennas Propag.* **58**(1), 145–154 (2010)
10. Dilman, I., Bilgin, E., Cosgun, S., Cayoren, M., Akduman, I.: A compressive sensing application on microwave diffraction tomography for the microwave imaging of a stroke affected human brain. In: International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (2018)
11. Semenov, S.Y., Corfield, D.R.: Microwave tomography for brain imaging: feasibility assessment for stroke detection. *Int. J. Antennas Propag.* **2008**, 1–8 (2008)
12. Bisio, I., Estatico, C., Fedeli, A., Lavagetto, F., Pastorino, M., Randazzo, A., Sciarrone, A.: Brain stroke microwave imaging by means of a Newton-Conjugate-Gradient method in L^p Banach spaces. *IEEE Trans. Microw. Theory Tech.* **66**(8), 3668–36682 (2018)
13. Nagy, J.G., Strakos, Z.: Enforcing non-negativity in image reconstruction algorithms. SPIE 4121, Mathematical Modeling, Estimation and Imaging, San Diego (2000)
14. Calvetti, D., Reichel, L., Shuibi, A.: Enriched Krylov subspace methods for ill-posed problems. *Linear Algebra Appl.* **362**(2003), 257–273(2002)
15. Brain data Homepage. http://brainweb.bic.mni.mcgill.ca/brainweb/anatomic_ms.html

Analysis and Identification of EEG Features for Mental Stress



Mitul Kumar Ahirwal 

Abstract In this paper, an attempt for identification of electroencephalogram (EEG) signal features for detection of stressed mental state has been made. In present lifestyle, almost all peoples suffer from stress. Assessment and management of stress are necessary to avoid serious mental illness. By the mean of EEG signal analysis, stressed mental state is easily detected. Here, 24 EEG features including spectral, amplitude, connectivity, and range properties are extracted in different frequency bands. A well-known feature selection algorithm sequential floating backward selection (SFBS) is used to select subset of best features. Features of theta frequency band EEG signal are identified as best features for mental stress identification.

Keywords Stress · EEG signal · Feature extraction and selection

1 Introduction

Nowadays almost all the peoples feel stressed in their daily life routine. There is an important relationship of mental health with stress levels. Stress in general considered as psychological stress that is associated with emotions like anger, anxiety, and depression. Stress also has effects on social life as well as physical health. Some conditions like chronic stress, depression, and anxiety also causes abnormal autonomic nervous system (ANS) functioning. Therefore, stress is considered as major factors responsible for chronic disorders. Interest in work, work performance, and attitude toward life are also influenced by stress [1].

In various industries, higher level of stress during work and stress-related disease degrade/decrease the company output/overall performance and increase the medical expenses of employees/workers. Deaths are associated with stress because it is also responsible for heart or brain blood vessel diseases. Stress can increase social and

M. K. Ahirwal (✉)

Computer Science and Engineering Department, Maulana Azad National Institute of Technology,
Bhopal 462003, India

e-mail: ahirwalmitul@gmail.com

economic losses for any country. Indian Armed Forces are also affected by stress. This has been an important topic of discussion among civil society and political classes. Increase in the rates of suicide, stress-related physical disorders, and psychiatric illnesses are also reported [1, 2]. Therefore, defensive or precautionary steps, based on some measures to reduce stress and adequate management of stress, are essential for the welfare of society at national as well as international level.

The stress response will be measured, quantified, and evaluated on the basis of perceptual, behavioral, and physical responses. Self-report questionnaires evaluation by verbal or written enquiries are one of the most common methods used to measure stress levels. Different types of questionnaires have been developed and used in clinical practice and psychiatric research to measure stress levels. Some of them are stress response inventory (SRI), life events and coping inventory (LECI), and the perceived stress scale (PSS), which includes cognitive, emotional, and behavioral responses of stress. Presumptive stressful life events scale (PSLES) and armed forces medical college life events scale (AFMCLES) are used specifically for soldiers. Other than above methods, stress can also be measured by physiological responses of body [2].

Stress makes changes in autonomic functioning of various organs that lead to changes like, heart rate and blood pressure increase during stress. Analysis, like heart rate variability (HRV), is used to observe beat-to-beat variation in heart rate. HRV analysis is a biomarker of ANS activities effected by mental stress. Electrocardiogram (ECG) signal is used for HRV analysis. Imaging techniques like Computer Tomography (CT) and Magnetic Resonance Imaging (MRI) with movement of body parts have demonstrated measurable physical and mental changes due to fatigue [3]. But these imaging techniques are expensive, bulky and puts some constrain over subject like motionless state of body. They also include radiation exposure which may have bad effects over subject health when continually used. In case of electroencephalography (EEG), above-said problems are not there and it is a noninvasive technology that can be easily used to continuously record and observe subject's neurological states [3, 4]. Different EEG frequency bands, alpha (8–13 Hz frequency band), beta (15–25 Hz frequency band), and gamma (30–60 Hz frequency band) bands of the EEG, have been frequently used and referred in clinical research because of their functional importance, in different task [5]. Further, it is possible by recent development and advances in EEG technology that changes in brain state are observed in EEG recording. Numbers of EEG-based studies and researches are present that shows the importance of EEG signal usefulness in systems where user/subject can operate devices just by imagination in its brain. Such types of system are known as brain computer interface (BCI) systems [6]. Studies of brain activity patterns, during stressful conditions, utilize EEG signal. EEG signal analysis is the best way to monitor the brain states. Many researches are available that explores the importance of EEG signals in brain functions and states classification and identification problems [1, 7, 8].

Many professions due their nature of work impose significant mental and/or physical strain over employees/workers. Some professions such as police, fire-fighters, soldiers, pilots have an inherent responsibility of safety of others peoples. To make

sure that employees/workers in such type of professions are fit for duty is an important health/safety concern for the workers as well as for those they are serving [3].

These profession services include period of extreme/hard physical with mental fatigue/pressure in difficult work environments. There is need of warning or alarm before the physics or mental pressure/stress/fatigue of worker exceeds certain limit. Because under such difficult and challenging environment, mental and/or physical fatigue can affect the working capacity and performance of workers. In some cases, it leads to health damage of the workers [3].

Therefore, this study aims to identify important features and characteristics of EEG signals during stressed condition, which can be further used as biomarkers for warning system. Rest of paper is arranged as follows: Sect. 2 consists of feature extraction and selection procedure. Results are presented in Sect. 3. Discussion and conclusion are presented in Sects. 4 and 5. Section 6 shows future direction to extend this work.

2 Methodology

2.1 Brief Procedure

For this study DEAP dataset has been taken [9], this dataset contains EEG signals recorded at the time of audio-visual stimulation. Responses of subjects in terms of valence and arousal are also given in dataset. In first step, EEG recordings are identified in which stress and relax state are observed according to circumplex model of affect [10]. The conditions for stress and relax state are given in Eq. (1).

$$\left. \begin{array}{l} \text{Stressed} = \text{IF } (\text{valence} < 3 \text{ AND Arousal} > 7) \text{ then "stress"} \\ \text{Relaxed} = \text{IF } (\text{Valence} > 7 \text{ AND Arousal} < 3) \text{ then "stress"} \end{array} \right\} \quad (1)$$

Above conditions are based on values of valence and arousal. Stressed and relaxed recording has been identified according to above condition.

2.2 Feature Extraction

Total six types of features are extracted that are based on spectral, amplitude, connectivity, and range properties of EEG signal [11]. Before features extraction, signal of each channel is decomposed into three frequency bands theta— θ (4–8 Hz), alpha— α (8–16 Hz), and beta— β (16–32 Hz). Details of features are as following:

(A) Spectral Power (SP)

The spectral power or power spectral density (PSD) of EEG signal $x[n]$ of length N samples with sampling frequency f_s Hz is calculated by Eq. (2),

$$P[k] = \frac{1}{\text{LMU} f_s} \sum_{l=0}^{L-1} \left| \sum_{n=0}^{M-1} x[n] w[n - lK] e^{-j2\pi kn/M} \right|^2 \quad (2)$$

In the above equation, $w[n]$ is the analysis window of length M having energy $U = \frac{1}{M} \sum_{n=0}^{M-1} |w[n]|^2$. $K = \lceil M(1 - H/100) \rceil$ is the time-shift factor, H is percentage of overlap, and window length is M . $L = \lfloor (N + K - M)/K \rfloor$ is the number of segments.

(B) Spectral Entropy (SE)

Shannon entropy is used to measure spectral entropy, calculated by Eq. (3),

$$F_{\text{shannon}}^i = -\frac{1}{\log L_i} \sum_{k=a_i}^{b_i} \bar{P}_i[k] \log \bar{P}_i[k], \quad (3)$$

where L_i represents the length of sequence of $[a_i, b_i]$ representing the range of the frequency band. $\bar{P}_i[k] = P[k]/\sum_{k=a_i}^{b_i} P[k]$ is the normalized spectral density, $P[k]$ is the PSD estimate.

(C) Amplitude Standard Deviation (ASD)

It is the measures of variability of the EEG signal from the mean value, Eq. (4) is used, where N is length of signal and \bar{x} is mean value of signal.

$$\text{SD} = \sqrt{\frac{\sum_{n=0}^{N-1} (x[n] - \bar{x})^2}{N}}, \quad (4)$$

(D) Amplitude Envelope Mean (AEM)

Amplitude of signal envelope is calculated as the mean value of envelope $e_i[n]$, denoted by Eq. (5),

$$e_i[n] = |x_i[n] + jH\{x_i[n]\}|^2, \quad (5)$$

where H represents the discrete Hilbert transform.

(E) Connectivity Brain Symmetry Index (CBSI)

Brain symmetry index is the measure of symmetry among hemispheres. For this, PDS is estimated for all channels. $P_m[k]$ is the PDS of m th EEG channel. Left-hemisphere

channels are arranged as $m = 1, 2, \dots, M/2$ and right-hemisphere channels are arranged as $m = M/2 + 1, M/2 + 2, \dots, M$. After this, two mean PSDs are calculated over left and right hemispheres. In Eq. (6), case of left hemisphere is given,

$$P_{\text{left}}[k] = \frac{1}{M/2} \sum_{m=1}^{M/2} P_m[k], \quad (6)$$

where $P_m[k]$ is the PSD of m th channel. Similarly, P_{right} is calculated for right hemisphere. The measure of symmetry is the differences of two PSDs for the i th frequency band, as shown in Eq. (7),

$$C_{\text{BSI}}^i = \frac{1}{L_i} \sum_{k=a_i}^{b_i} \left| \frac{P_{\text{left}}[k] - P_{\text{right}}[k]}{P_{\text{left}}[k] + P_{\text{right}}[k]} \right|, \quad (7)$$

where $[a_i, b_i]$ is the frequency range for the i th band and $L_i = b_i - a_i$.

(F) Connectivity Correlation (CC)

Connectivity correlation also measures the hemisphere connectivity based on signal envelope for different frequency band between channels and across the hemispheres. Channels are arranged into pairs based on their locations. Frontal channels are paired as (F3 and F4) and central channels as (C3 and C4) are example of pairing. Pearson correlation coefficient is used. In Eq. (8), calculation is shown,

$$C_{\text{corr}}^i = \text{median}[c_i(m)], \quad (8)$$

In above equation, $c_i(m)$ is the m th pair for $m = 1, 2, \dots, M/2$. The median over all pairs is used as final value.

(G) Range EEG Mean (REEGM)

Range EEG is an alternative to amplitude-integrated EEG (aEEG). Normally, EEG machines implement different versions of the aEEG algorithm. In range EEG, peak-to-peak measure of voltage is estimated. Features of range EEG using either the full-band signal $x[n]$ or specific frequency bands $x_i[n]$ is calculated. In a short-time windowed segment, the difference between the maximum and minimum is generated by Eq. (9),

$$r_i[l] = \max(x_i[n]w[n - lK] - \min(x_i[n]w[n - lK])), \quad (9)$$

For window $w[n]$ and $K = \lceil M(1 - H/100) \rceil$ is the time-shift factor with percentage overlap H and window length M . Equation (10), shows the mean value of $r_i[l]$ for i th frequency band,

$$R_{\text{mean}}^i = \text{mean}(r_i[l]), \quad (10)$$

(H) Range EEG Asymmetry (REEGAS)

Range EEG asymmetry measures the skewness about median and defined by Eq. (11),

$$R_{\text{symm}}^i = \frac{(R_{\text{upper}}^i - R_{\text{median}}^i) - (R_{\text{median}}^i - R_{\text{lower}}^i)}{R_{\text{bw}}^i}, \quad (11)$$

R_{median}^i is the median (as measures of central tendency) of $r_i[l]$. R_{lower}^i and R_{upper}^i are the lower and upper margins. $R_{\text{bw}}^i = R_{\text{upper}}^i - R_{\text{lower}}^i$ is the difference between the upper and lower margins. R_{symm}^i varies from -1 to 1 , 0 indicating symmetry and values near to ± 1 indicating asymmetry of the Range EEG.

The above-mentioned features are calculated for three frequency bands, theta, alpha, and beta. For easy understanding features are listed with serial numbers in Table 1.

Table 1 Features names with serial numbers

S. No.	Name
1	Spectral power (theta)
2	Spectral power (alpha)
3	Spectral power (beta)
4	Spectral entropy (theta)
5	Spectral entropy (alpha)
6	Spectral entropy (beta)
7	Amplitude SD (theta)
8	Amplitude SD (alpha)
9	Amplitude SD (beta)
10	Amplitude envelope mean (theta)
11	Amplitude envelope mean (alpha)
12	Amplitude envelope mean (beta)
13	Connectivity BSI (theta)
14	Connectivity BSI (alpha)
15	Connectivity BSI (beta)
16	Connectivity corr (theta)
17	Connectivity corr (alpha)
18	Connectivity corr (beta)
19	Range EEG mean (theta)
20	Range EEG mean (alpha)
21	Range EEG mean (beta)
22	Range EEG asymmetry (theta)
23	Range EEG asymmetry (alpha)
24	Range EEG asymmetry (beta)

Table 2 Best features subset

Feature No.	Name
05	Spectral_entropy (alpha)
19	Range EEG_mean (theta)
07	Amplitude_SD (theta)
13	Connectivity_BSI (theta)
10	Amplitude_env_mean (theta)
01	Spectral_power (theta)

2.3 Feature Selection

Feature selection methods are used to select a subset features from the given set of features. The aim of feature selection is to identify the significant features and removal of redundant and irrelevant features. There are many verities and versions of feature selection algorithms are available in literature [12–14]. Here, sequential floating backward selection (SFBS) algorithm is used for features selection.

3 Results

After the identification of stressed and relaxed samples, they are treated as two classes for classification. All the features listed in Table 1 are extracted and SFBS feature section algorithm has been applied. The best feature subset that gives approx 75–80% classification accuracy is listed in Table 2 with their name and feature number.

4 Discussion

Due to modern and fast lifestyle, all peoples suffer by stress. Causes of stress may be different for different age group or professions. For better assessment and management of stress, identification of stressful state is necessary. With help of EEG signal analysis, stress is identified. Several features are extracted from EEG signal and best among them are selected. In above selected features, all most all the features are of theta band. EEG signals in theta (4–8 Hz) frequency band, regardless of location generally represent drowsiness. This study shows the significance of theta band in identification of stressed state. Alpha band EEG signal feature is also selected in best feature subset.

5 Conclusion

Identification of best features for stress state from EEG signal has been done. Total 24 features that represent different characteristics of EEG are extracted. SFBS features selection algorithm is used to select beat features. Features named as spectral entropy (alpha), range EEG (theta), amplitude_SD (theta), connectivity_BSI (theta), amplitude_env_mean (theta) and spectral_power (theta) are selected as best features out of all features. It is concluded that theta band signal contributes more as compared to other bands in identification of stressed mental state.

6 Future Work

This work can be extended in the direction of channel and locations identification for stress state. Analysis of individual channels can be performed with the best selected features.

Declaration

I have taken permission from competent authorities to use the images/data as given in the paper. In case of any dispute in the future, we shall be wholly responsible.

References

1. Seo, S.-H., Lee, J.-T.: Stress and EEG. In: Convergence and Hybrid Information Technologies. IntechOpen (2010)
2. Ryali, V.S.S.R., Bhat, P.S., Srivastava, K.: Stress in the Indian Armed Forces: how true and what to do? Med. J. Armed Forces India **67**(3), 209 (2011)
3. Kadambi, P., Lovelace, J.A., Beyette, F.R.: Changes in behavior of evoked potentials in the brain as a possible indicator of fatigue in people. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE (2013)
4. Ko, L.-W., et al.: Neural oscillations in temporoparietal lobes under inhibitory control in a naturalistic situation. In: 2016 IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE). IEEE (2016)
5. Ko, L.-W., et al.: Mobile EEG & ECG integration system for monitoring physiological states in performing simulated war game training. In: 2015 IEEE Conference on Computational Intelligence and Games (CIG). IEEE (2015)
6. Rezeika, A., et al.: Brain-computer interface spellers: a review. Brain Sci. **8**(4), 57 (2018)
7. Panicker, S.S., Gayathri, P.: A survey of machine learning techniques in physiology based mental stress detection systems. Biocybern. Biomed. Eng. (2019)
8. Charles, R.L., Nixon, J.: Measuring mental workload using physiological measures: a systematic review. Appl. Ergon. **74**, 221–232 (2019)
9. Koelstra, S., Muhl, C., Soleymani, M., Lee, J.S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., Patras, I.: DEAP: a database for emotion analysis; using physiological signals. IEEE Trans. Affect. Comput. **3**(1), 18–31 (2012)
10. Russell, J.A.: A circumplex model of affect. J. Pers. Soc. Psychol. **39**(6), 1161 (1980)

11. Toole, J.M., Boylan, G.B.: NEURAL: quantitative features for newborn EEG using Matlab. arXiv preprint [arXiv:1704.05694](https://arxiv.org/abs/1704.05694) (2017)
12. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recogn. Lett. **15**(11), 1119–1125 (1994)
13. Ververidis, D., Kotropoulos, C.: Fast and accurate feature subset selection applied into speech emotion recognition. Signal Process. **88**(12), 2956–2970 (2008)
14. Ververidis, D., Kotropoulos, C.: Feature Selection using Matlab. <https://in.mathworks.com/matlabcentral/fileexchange/22970-feature-selection-using-matlab?focused=5164696&tab=function> (2009)

Blockchain-Based Grievance Management System



Rakshitha Shettigar, Nishant Dalvi, Ketan Ingale, Farhan Ansari, and Ramkrushna C. Maheshwar

Abstract Since the abrupt outburst of Bitcoin, there has been a meteoric rise in the popularity of blockchain which has been scaled to various domains. Its features have changed the outlook of solutions for many problems and can be applied in various sectors, one being grievance redressal system. In this system, the grievant will submit a grievance which will pass through different levels of hierarchical authoritative screening. Each screening level will have the authority to debar, resolve, revert and forward the grievance to the higher level. Since data integrity is a built-in feature of this system, it eradicates any chances of misuse/abuse of power by the authorities. The dynamic time threshold is an additional feature that automatically transfers the grievance to the higher authority in the hierarchy, thus eliminating any chances of ignorance. Thus, the anomalies and shortcomings of the current grievance system can prevail over.

Keywords Blockchain · Dynamic time threshold · Grievance redressal system · Hierarchical authoritative screening · Immutable · Distributed · Decentralized · Data integrity

R. Shettigar · N. Dalvi (✉) · K. Ingale · F. Ansari · R. C. Maheshwar
International Institute of Information Technology, Pune, India
e-mail: dalvi.nishant@gmail.com

R. Shettigar
e-mail: rakshithashettigar@gmail.com

K. Ingale
e-mail: iketu312@gmail.com

F. Ansari
e-mail: farhanahraf03@gmail.com

R. C. Maheshwar
e-mail: remomaheshwar1987@gmail.com

1 Introduction

Blockchain is a decentralized ledger containing transactions across a P2P network. A blockchain contains a set of rules and protocols that facilitate transactions. It is an incorruptible digital ledger of transactions that can be programmed to record not just financial transactions but virtually everything of value.

Blockchain can be viewed as a custom data structure consisting of various fields defined as per application requirements. The first block of a blockchain network is called a genesis block. Every block in the network consists of data entries, the current block's hash value, timestamp of creation and the hash value of the previous block [1]. A hash is a function generated encrypted value that depends on the input data of the block. If the data within the block is modified, then the hash value is changed as well.

In Fig. 1:

1. Current Hash of Block 01 and Block 02 is generated using data present in the respective block
2. Previous hash points to the hash of the previous block

Blockchain [2] provides features such as immutable data, decentralized technology, enhanced security and distributed ledgers. Many of these features are implemented not only in the financial domain but also in various industrial, government bodies as well as management systems [3]. One such application is Grievance Redressal systems.

A grievance can be considered as an official statement of a complaint about something believed to be unfair or unjust. Grievance Redressal refers to receiving and processing complaints or issues from grievant/consumers and action taken on these issues to avail services more effectively. Grievance Redressal Systems are most implemented in institutions, government bodies and private Industries.

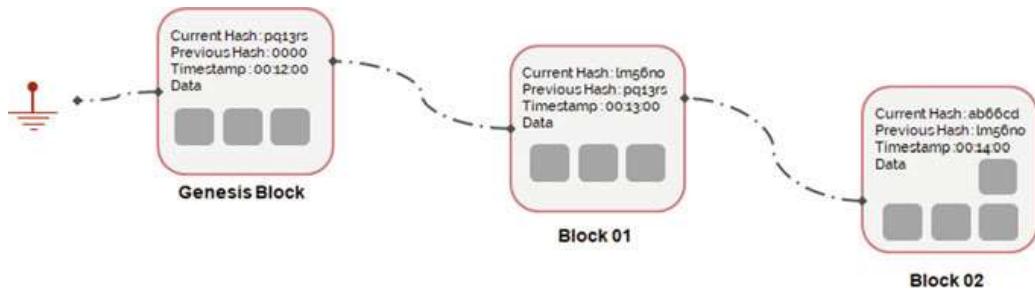


Fig. 1 Structure of blockchain

An institutional grievance redressal system that is built on a centralized architecture requires a constant backup of data. Although there are many optimal methods to reduce the backup process overhead, such systems are still susceptible to vulnerabilities that would affect the entire system. In case of such a security breach, the server will reflect the malicious data, affecting all the systems throughout the network. Thus, data integrity is one such feature that must not be overlooked. Some other features that are underestimated are data consistency and ignorance of the grievance by an authority. Moreover, ignorance of authority or malpractices needs to be removed.

As a result, we propose a system that would empower students keeping their satisfaction as a top priority. In this paper, we provide a solution that uses blockchain technology to overcome all the shortcomings of the traditional system. We provide a system where the entire grievance process is transported on the blockchain network. This can be done by making each grievance thread a single blockchain associated with the issue generated by the student. This is followed by passing the thread through each level and appending all the updates to the thread accordingly. Now, to make this efficient and ignorance proof there are different rules and actions associated with the levels defined, which are further discussed in this paper.

2 Benefits Over Current System

The primary drawbacks in the current traditional system are mishandling and ignorance of grievances. These drawbacks often lead to the inefficiency of the system. To overcome these shortcomings, a blockchain network is used. The data in the blocks can't be tampered with and thus, the higher authorities are forced to comply with the responsibility of managing the grievances. However, there is no such timeline that the grievant can follow. Hence, we can say that the data updated in the blockchain doesn't change and is kept authentic. Moreover, the middleware algorithm is written in such a way that the grievance is forwarded to the higher level of hierarchy after a specific time if left unattended or unresolved.

At present, there is no common interface between the committee, Head of the Department and the Principal. Hence, this application bridges the gap between the students and the respective authorities in a hierarchical manner by providing complete information related to the grievance which was uploaded.

Since the hierarchical network is made completely transparent, the student knows the authority to which his/her grievance is currently addressed to. This transparent nature helps in empowering the students (Fig. 2).

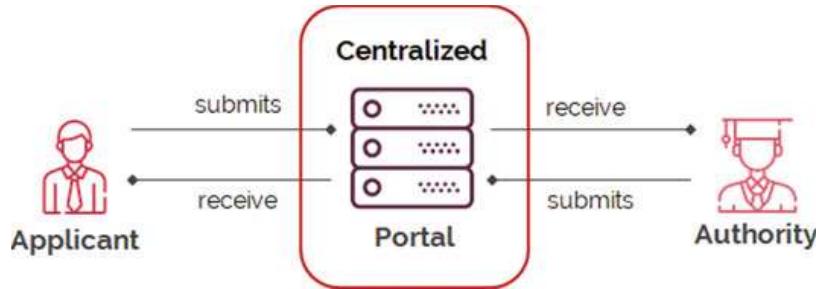


Fig. 2 Existing system

3 Approach

We limit the application scope of the redressal system to institutes. This redressal system addresses the grievances submitted by students through their dashboards. The submitted grievances flow through a three-level system. These three levels are the college Grievance Committee, Head of Department and Principal. These levels of hierarchy may vary from institute to institute. When a student submits a grievance, it first reaches the Institute Grievance Committee. If the committee is unable to cater to the grievance in a fixed time or is out of its scope and requires higher authority intervention, then the committee can either forward it to the next level or it is automatically forwarded after a fixed time. The higher authority can either resolve the grievance or it can ask the lower level authority to address it. This process continues to the last level.

The system that we propose has the features described as follows. It is a platform for students to submit a grievance and features a real-time level-wise progress tracker. The system is immune to various attacks and overcomes the shortcomings of the traditional online grievance portal. Even when the grievance is ignored by one level, it gets forwarded to the next level, they can send back the grievance to an appropriate level for resolution. The system removes the need for a mediator and replaces it with a highly secure and distributed network. The probability of an unexpected risk to occur is substantially diminished as any third-party intervention is not present in the system. Our system also shows a statistical analysis depicting the number of grievances resolved and unresolved in a particular department of the institute or any other analysis as per requirement. The higher level authority gets informed if the lower level authority has not attended to the grievance. A student may exercise his right to upvote or downvote any grievance in the particular institute. The student has the choice to keep his/her identity concealed whilst submitting a grievance. In such a case, the identity on the authority dashboard will be displayed as 'Anonymous User'.

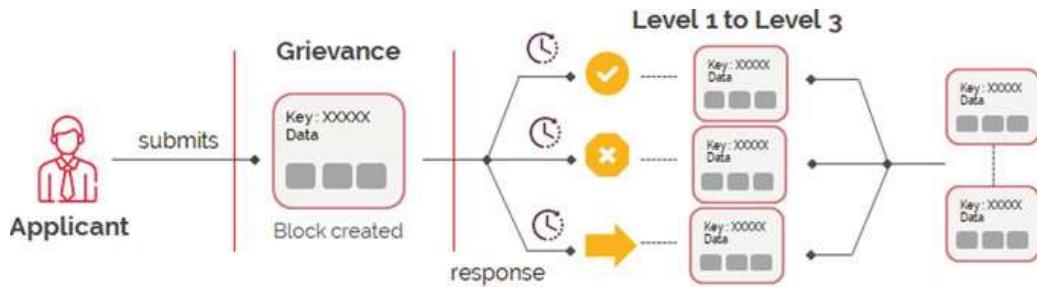


Fig. 3 Working of our system

3.1 Working

The student signs up or logs in his/her respective dashboard of the web portal. After logging in, the student can submit a grievance or can view the status of ongoing grievance and view previous grievances (Fig. 3).

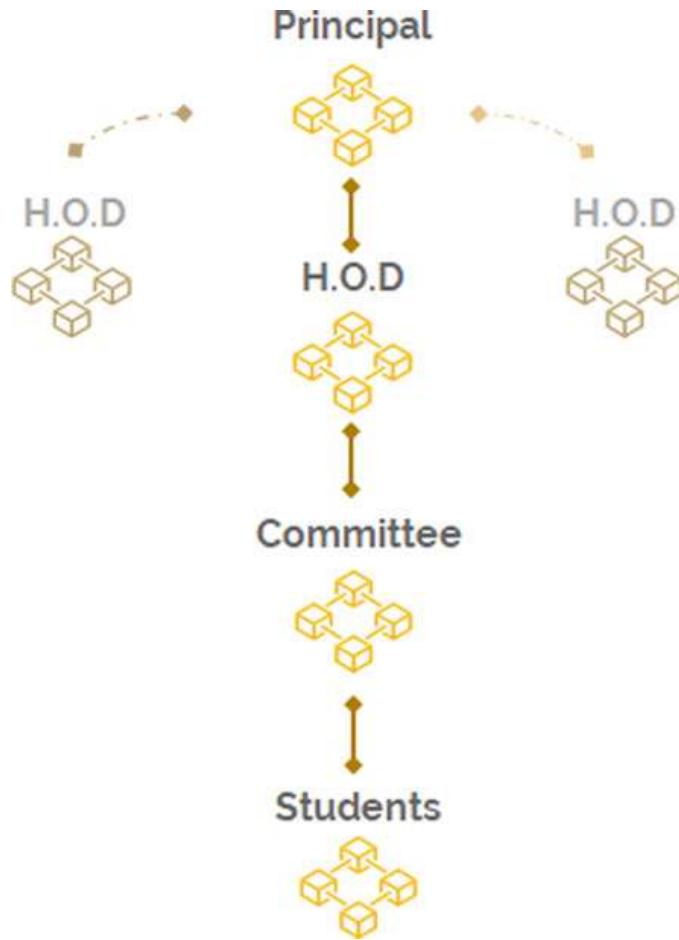
Every new grievance submitted will be designated as a genesis block in the blockchain network. This new grievance is sent to the first level of screening, the Grievance Committee. The committee has added bonus to debar the grievance if it is insignificant. If the grievance is severe and requires the immediate attention of superior authority, then it is forwarded to the proceeding level, i.e. Head of the Department. If the grievance isn't attended to, then it will be automatically sent to the proceeding level with superior authoritative capabilities. The HOD may perform any action which he/she may deem fit. If the grievance is forwarded by the HOD, it is submitted to the Principal which is the highest authority in this system. After the grievance is solved by any level, the student can see the solution and can either accept the solution or can revert the solution if not complied or unsatisfied with.

3.2 System Architecture

Following are the stakeholders and their corresponding actions (Fig. 4):

Student: It is the entry point for the system where life cycle of the submitted grievance begins. It provides an integrated web-based interface for submitting, viewing and responding grievances.

Level 1—Grievance Committee: The Grievance Committee may include several members in the decision-making process. Hence submitted grievances require the agreement of the majority if it needs to be forwarded or debarred. This is catered with the help of the consensus protocol that is followed during implementation. On the other hand, the grievance can be responded if found solvable at this level. Hence, the actions involved at this level are as follows:

Fig. 4 System architecture

- Resolve
- Forward
- Debar.

Level 2—Head of Department: Grievances received from the committee as well as grievances which are automatically forwarded (due to the threshold time) are reflected on the dashboard of the respective departmental H.O.D. Similar to the Grievance Committee, this level also involves the forwarding of the grievance if found severe or ignored. On the other hand, if deemed solvable it can be resolved. Whereas, if a particular grievance could be resolved at the committee level but was still forwarded due to some reason, it can be reverted to the committee for resolution along with the reason. Hence, the actions involved at this level are as follows:

- Resolve
- Forward
- Revert

Level 3—Principal: Grievances that require attention by the highest authority of the Institution are reflected on the dashboard of the Principal. These grievances are the ones raised by the HOD or which are automatically forwarded (due to the threshold time). Similar to the previous levels, this level too can provide the following actions:

- Resolve
- Revert

It is important to note that the students who receive a response can mark the response as satisfied/not satisfied and can provide feedback regarding the same. If the response received from any level is marked not satisfied then, the grievance is again forwarded to the Grievance Committee. If the response received from any level is marked satisfied then, the grievance is then marked resolved and closed (Fig. 5).

3.3 *Mathematical Model*

Inputs:

Students:

- D—Details:
 - n—Name
 - yb—Year and Branch
 - em—E-mail
 - num—Phone no.
- G—Grievance:
 - t—Title
 - c—Category
 - sc—Sub-category
 - d—Description
 - doc—Supporting Documents

The sets are represented as follows:

S—{D, G} where
 D—{n, yb, em, num}
 G—{g₁, g₂, ..., g_n}
 g—{t, c, sc, d, doc}

Functions:

Student:

1. Submit_Grievance()
2. Submit_Feedback()
3. Track_Progress()

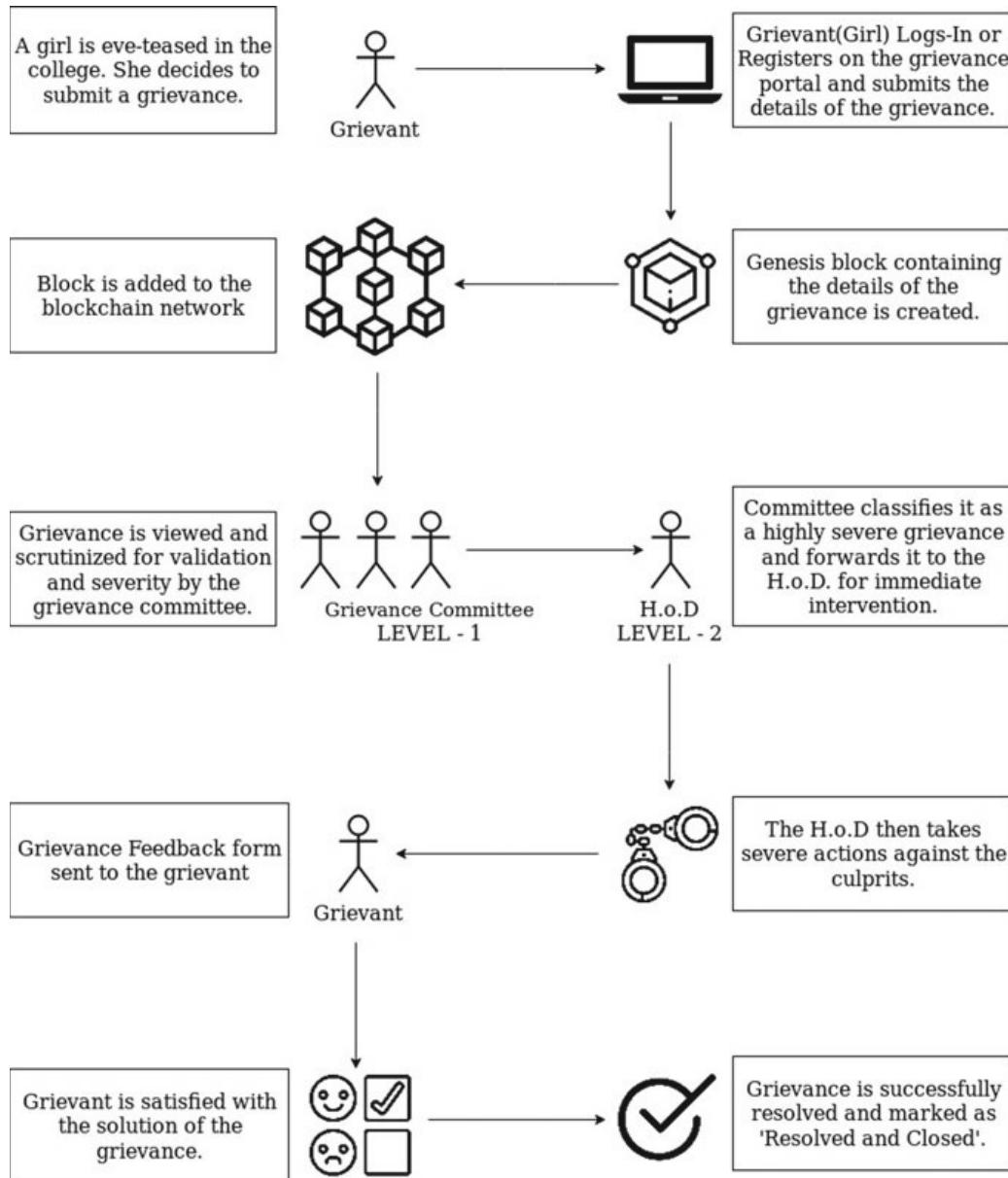


Fig. 5 Hypothetical scenario

4. SignUp()
5. Login()

Committee/HOD/Principal:

Let 'A' be the set of actions, defined as follows:

- fwd()—ForwardTheGrievance()
- rvt()—RevertBackTheGrievance()
- rsv()—ResolveTheGrievance()
- dbr()—DebarTheGrievance()

Functions of individual levels are as follows:

1. **Committee:** A—{rvt()}
2. **HoD:** A—{dbr()}
3. **Principal:** A—{fwd(), dbr()}

Hierarchy:

The hierarchy is as shown below:

- College – {principal, staff, grievance_committee, students}
- staff – {teaching, non-teaching}
- teaching – {HoD, asst_prof}
- HoD – {h₁, h₂, h₃, h₄}
- asst_prof – {ap₁, ap₂, ..., ap_n}
- non_teaching – {nt₁, nt₂, ..., nt_n}
- grievance_committee – {h, a, n, ∀ h ∈ HoD, ∀ a ∈ asst_prof, ∀ n ∈ non_teaching}
- student – {s₁, s₂, ..., s_n}

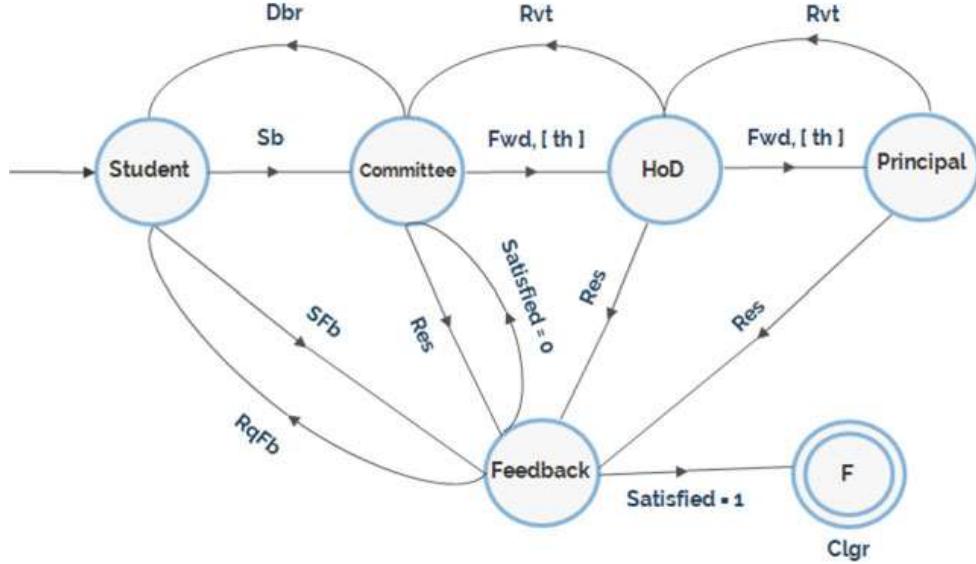
Finite-State Machine Diagram: The finite-state machine diagram depicts the different states where control may end up upon executing a specific sequence of inputs. Success state can be formed consisting of a certain input segment which ends up at the final stage.

Failure states consist of sets of input sequences that do not find themselves at the final state upon completion.

$$S = \{Q, \Sigma, \delta, Q_0, F\}$$

$$Q = \{\text{Student, Committee, HoD, Principal, Feedback, F}\}$$

$$\Sigma = \{\text{th, Sb, Fwd, Rvt, Res, SFb, RqFb, Dbr, S}_1, \text{S}_0\}$$



- $$\delta =$$
- 1) Student = Sb.Committee + SFb.Feedback
 - 2) Committee = (Fwd + Th).HoD + Res.Feedback + Dbr.Student
 - 3) HoD = (Fwd + Th).Principal + Res.Feedback + Rvt.Committee
 - 4) Principal = Res.Feedback + Rvt.HoD
 - 5) Feedback = RqFb.Student + S0.Committee + S1.F

Q_0 = Student

F = Final State

th	=>	Threshold ()
Sb	=>	Submit_grievance ()
Fwd	=>	Forward_grievance ()
Rvt	=>	Revert_grievance ()
Res	=>	Resolve_grievance ()
SFb	=>	Submit_feedback ()
RqFb	=>	Request_feedback ()
Dbr	=>	Debar_grievance ()
Satisfied=>		{ 0, 1 }

Success State -

$$A = \{(Fwd + Th)^{[0..2]} . (Rvt)^{[0..2]} . Res . Rqfb . SFb . (S_0 . A + S_1 . F)\}$$

$$B = \{(Sb . Dbr)^* Sb(Fwd + Th)^{[0..2]} . (Rvt)^{[0..2]} . Res . Rqfb . SFb . (S_0 . A + S_1 . F)\}$$

F = Final State of State Machine Diagram

Success State

$$A = \{(Fwd + Th)^{[0...2]}, (Rvt)^{[0...2]}, Res, Rqfb, SFb, (S_0, A + S_1, F)\}$$

$$B = \{(Sb, Dbr)^*, Sb(Fwd + Th)^{[0...2]}, (Rvt)^{[0...2]}, Res, Rqfb, SFb, (S_0, A + S_1, F)\}$$

F = Final State of State Machine Diagram.

4 Implementation

The proposed system is a use case of permissioned blockchain which can be implemented using tools such as Hyperledger Fabric (HF) Composer. Hyperledger Fabric Composer is used for developing Distributed Ledger Technology (DLT) for organizations that need to automate their business process. In this case, the life cycle of grievance can be considered as a business process. Using HF, a blockchain network can be developed.

5 Conclusion

With the widely increasing popularization and its deepened applications, blockchain technology is becoming a hot research topic in both academic and industrial communities [4]. The varied characteristics such as decentralized nature, verifiability and enforceability enable the contract terms to be executed between untrusted parties without the intervention of a trusted authority or central server. In this paper, we propose a blockchain-based grievance redressal system. We introduce the working principle of the system and discuss the issues related to the current system and how we are to overcome them in our system. The system, being on blockchain, is immutable and untameable. The decentralized nature of the blockchain enables us to have a secure, fault-tolerant system with added security [5]. In our model, we propose a three-level escalated system with the levels being Grievance Committee, Head of the Department and the Principal of the Institute. The grievance will be automatically passed to the next level if not solved within a specified time frame. Authorities will have the privilege to forward the grievance to the next level, revert the grievance to lower level and resolve the grievance. The committee will have an added privilege to debar the grievance if deemed irrelevant. The student (grievant), if discontented with the solution, can again raise the grievance.

In future work, we plan to expand our system to various other institutions, government agencies, private firms and add some more features/add-ons to the system. May the need arise, the system can be modified to add features or changes as per the requirements of the customer.

References

1. Wang, S., Ouyany, L., Yuan, Y., Ni, X., Han, X., Wang, F.: Blockchain-enabled smart contracts: architecture, applications and future trends. *IEEE Trans. Syst. Man Cybern. Syst.* (2018)
2. Miller, D.: Blockchain and the internet of things in the industrial sector. *IT Prof.* **20**, 15–18 (2018)
3. Henry, R., Herzberg, A., Kate, A.: Blockchain access privacy: challenges and directions. *IEEE Secur. Priv.* **16**, 38–45 (2018)
4. Kshetri, N., Voas, J.: Blockchain in developing countries. *IT Prof.* **20**, 11–14 (2018)
5. Pilkington, M.: *Blockchain technology: principles and applications* (2015)

Utility of Neural Embeddings in Semantic Similarity of Text Data



Manik Hendre, Prasenjit Mukherjee, and Manish Godse

Abstract Semantic similarity plays an important role in understanding the context of text data. In this paper, semantic similarity between large text data is computed using different neural embeddings. we review the utility of different deep neural embeddings for text data representation. Most of the earlier papers have studied the semantic similarity of text by using individual word embeddings. In this paper, we have evaluated the neural embedding techniques on large text data with the help of Essay Dataset. We have used recent neural embedding methods such as Google Sentence Encoder, ELMo, and GloVe along with traditional similarity metrics including TF-IDF and Jaccard Index for experimental investigation. Experimental evaluation in this research paper shows that Google Sentence Encoder and ELMo embeddings perform best on semantic similarity task.

Keywords Embedding · Semantic similarity · Natural language processing · ELMo · Sentence encoder · Neural embedding · Text similarity

1 Introduction

In natural language processing field, there has been many advancements in last couple of months. We have more powerful language models which can perform various tasks as par with humans [14]. In tasks like sentiment analysis, chatbot, question answering, automatic essay evaluation, dialog systems, parsing, word-sense disambiguation, named-entity recognition, POS tagging, and many more, we are observing good

M. Hendre (✉) · P. Mukherjee · M. Godse
Department of Analytics and IT, Pune Institute of Business Management,
Pune, Maharashtra, India
e-mail: manik.hendre@pibm.in

P. Mukherjee
e-mail: prasenjit.mukharjee@pibm.in

M. Godse
e-mail: manish.godse@pibm.in

results [1, 4, 14]. The computing resources are more available and affordable now, as compared with couple of years back. Due to this, the research in NLP using deep learning techniques is taking new leap in every field [14]. For every NLP task, the numerical vector representation of text data is very important. Most of the deep learning techniques require numeric vectors as an input to the system. Traditional method of representing text into vector form is TF-IDF. Term frequency is the number of times a particular word appears in a document. Inverse document frequency assigns a weight to a word according to how rare or common that word is in set of documents. It gives more weightage to rarely occurring words. Product of term frequency and inverse document frequency is the single number representation of the word in a document.

Survey of different word embedding methods have been performed by wang et al. [13]. In this, authors point out that currently there are no metrics available for evaluation of word embedding models. The semantic and syntactic relations captured by word embedding models are different from how human beings understand languages [13]. Most of the word embedding models are task specific because they are trained for specific natural language processing tasks. In many NLP tasks, we need to compare different set of texts. The semantics have very important role to play in natural language generation and understanding. There are many ways in which same text can be written, having the same meaning. The semantics tries to capture this meaning from different text data. Automatic essay evaluation using word mover's distance is proposed by Tashu et al. [12]. In this, semantic similarity of text is given more weightage than the syntax and vocabulary. For calculating essay score, the word mover's distance between normalized continuous bag-of-word features is calculated [13]. Semantic similarity based on knowledge graphs is proposed by Zhu et al. [15]. Most of the semantic similarity techniques uses only surrounding words while computing semantic similarity. Knowledge graphs represent concepts, and complex relationships can be extracted from them [15].

Semantic similarity between two words, sentences, and paragraphs is presented by Pawar et al. [8]. In this, sentence similarity is computed in two phases, and in first phase, the similarity is maximized using word, sentence, and word-order similarity. In second phase, the skewness is removed which was introduced because of deviation from actual similarity. Automatic evaluation of text using word and sentence embeddings is proposed by Clark et al. [3]. Authors have introduced a new metric, sentence movers similarity which is the extension of word mover's distance for multiple sentences. Sentence movers similarity metric has improved correlation with the human judgment scores on automatic text evaluation task [3]. Jaccard index is a similarity metric widely used for computing similarity of text. Jaccard index calculates similarity as the intersection over union of the words in two set of texts. According to jaccard index, if there are many words which are common in two set of texts, then those texts are more similar. In semantic similarity, context of a word is important. Context representation method using bidirectional LSTM is proposed in [5].

Utility of a particular technique shows its effectiveness in performing a specific task. Main contribution of this paper is to calculate neural embeddings for large

text data and finding its utility on semantic similarity task. The organization of the paper is as follows. In Sect. 2, we list all the studied neural embedding techniques. Proposed methodology is explained in Sect. 3. Dataset used is explained in Sect. 4. Performance evaluation techniques and experimental results are presented in Sect. 5. Finally, the conclusions are drawn in Sect. 6.

2 Neural Embedding Techniques

Widely used neural-network-based word embedding, Word2vec [6][7], is developed by Mikolov et al. at Google. In [6], the authors have proposed two novel architectures for word embeddings. First architecture continuous bag-of-words model which predicts the current word, given a context of surrounding words. Second architecture is a continuous skip-gram model which tries to predict context, given an input word. The architecture used by authors is shallow network which is less computationally intensive. Several improvements over [6] are proposed in [7] by same author. As stop words do not provide much information regarding semantics, the authors have performed sub-sampling of stop words to improve the training speed [7]. Simple mathematical operations like addition and subtraction can be performed on word vectors, which surprisingly gives interesting semantic relationships among words [7]. The neural embeddings given by Word2Vec are good at maintaining semantic and syntactic structure among words [6, 7].

Sentence-level embeddings are proposed by Cer et al. [2]. In this, universal sentence encoder takes input sentence of any length and gives its 512 dimensional numeric representation. Two different approaches for sentence encodings are presented in universal sentence encoder. First approach uses transformer networks which gives accurate results at the expense of more computational resources. The second approach makes use of deep averaging network which are less accurate as compared with transformer-based model but are efficient in terms of speed and memory [2]. Earlier transfer learning based neural embeddings were word based, but the solution provided in universal sentence encoder is sentence based, and these models can be directly used with the help of transfer learning [2]. Embeddings for language model(ELMo) is another deep learning based embedding proposed by Peters et al. [10]. The embeddings are computed by using bidirectional language models. Specifically, long short-term memory (LSTM) with forward and backward passes have been employed for training purpose. ELMo is a feature-based approach. Unlike other methods in which neural embedding is a function of top layer, in ELMo, the final vector representation is the function of all the internal layers. ELMo has shown improvements on large number of NLP tasks [10]. Another word to vector representation GloVe (Global Vectors) which takes into account global information is developed at Stanford [9]. Unlike word2vec which only considers surrounding words while calculating embedding, the GloVe takes into account the global context. In GloVe, words are projected in a space such that semantically similar words will be adjacent to each other. For global context, the word-word co-occurrence statistics are calculated. This method performs good on word analogy task [9].

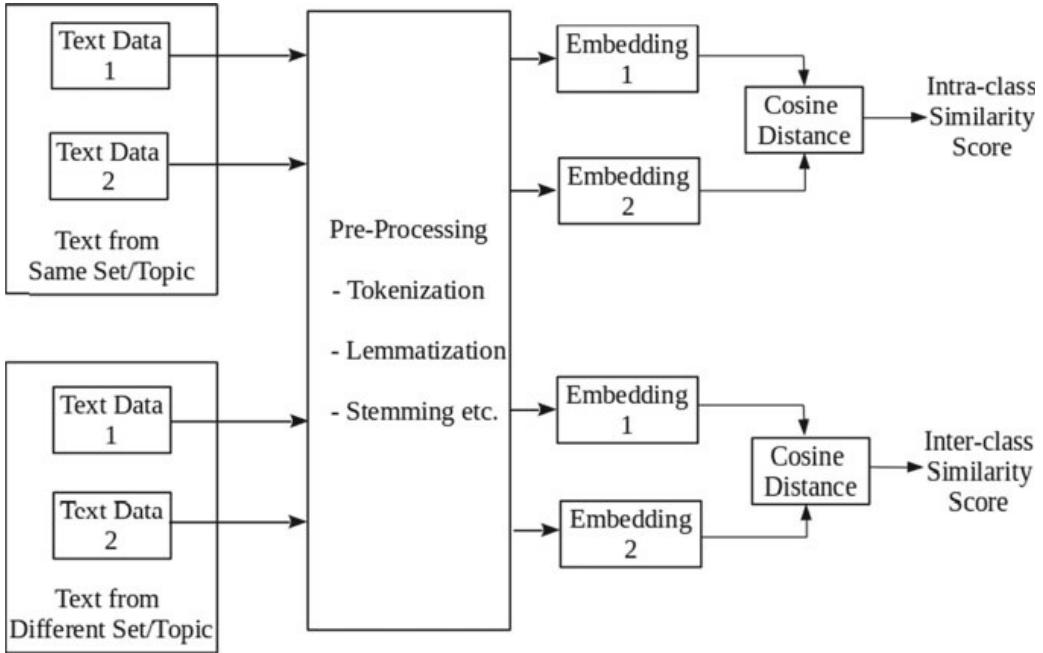


Fig. 1 Block diagram of semantic similarity system

3 Methodology

Figure 1 shows the block diagram of proposed semantic similarity system. For intra-class similarity, the input text should be from same set/topic. For calculating inter-class similarity, we have compared text from different set/topics. To calculate the semantic similarity, the text data has to be converted into its numerical vector representation. We call this vector representation as embedding. Let p_{ij} be the embedding of i th essay from j th set. We have in total eight sets containing essays on eight different topics. Let q_j be the embedding for the top scored essay from j th set. We use the cosine similarity metric to calculate the similarity between two set of texts.

Cosine similarity computes the angle between the vectors representing the embeddings of two sets of text data. When the angle between two embeddings is 0, then we get $\cos(\Theta)$ value as 1 which denotes that the embeddings are exactly similar to each other. The formula for cosine similarity is given in Eq. (1).

$$\cos(\Theta) = \text{similarity}(p_{ij}, q_j) = \frac{p_{ij} \cdot q_j}{|p_{ij}| |q_j|} \quad (1)$$

4 Data Management

We are using the dataset [11] provided under Kaggle competition, namely The Hewlett Foundation: Automated Essay Scoring. For this competition, total 12,977

essays are collected. These essays are written by students from grade 7 to 10. Essay length is not constant, each essay has typically 150 to 550 words into it. There are eight type of essays. Each type of essay has different set. They have also provided the marks scored by every student, but we are currently just focusing on semantic similarity part that is why we have not made use of scores. This dataset has good amount of variation in terms of text data. As we have eight different sets containing different essay content, we can calculate the semantic similarity between same set's essay as well as different set's essay.

5 Experimental Results

Top-scored essay from each essay set is considered as the model essay for comparison purpose. For intra-class similarity calculation, top-scored essay from each essay set is compared with all the essays from that set only. For inter-class similarity calculation, the top scored essay from each set is compared with first 100 essays each from other essay sets. So, we have 12,977 intra-class and 5600 inter-class similarity scores for each evaluated method. To compute the distance between neural embeddings, we have used cosine distance similarity metric. We have used TF-IDF, Jaccard, Glove [9], Google Sentence Encoder Large [2], Google Sentence Encoder Lite [2], and ELMo [10] methods to compute similarity between essay text.

5.1 Similarity Score Distribution

In this section, the intra-class and inter-class semantic similarity distributions are shown for each evaluated method. We have also used the box plots to show how each essay set distributions are performing for all used methods. we have used the same box plot to depict both intra-class and inter-class similarity scores. The notched box plots represents the intra-class semantic similarity scores, and the box plots without notches shows the inter-class similarity scores. Figure 2 shows the similarity distribution for TF-IDF method. In left graph of Fig. 2, we can see that overlapping region is more. In right plot of Fig. 2, we can see that there is overlap between intra-class and inter-class similarity scores for essay sets 3 and 7. For all other essay sets, we have good amount of separation. Similarity score distributions for Jaccard index method are shown in Fig. 3. In right plot of Fig. 3, we can see that there is a overlap in essay set 2 and 3.

Figure 4 shows the distribution plots for the GloVe embeddings method. In right graph of Fig. 4, we can see that most of the notched and normal box plots are overlapping, which denotes that the underlying score distributions are not significantly different. GloVe method fails in capturing the semantic similarity on essay text data as compared with other methods. We can see the performance of ELMo technique in Fig. 5. In right plot of Fig. 5, we can see that except for essay set 4, all other box

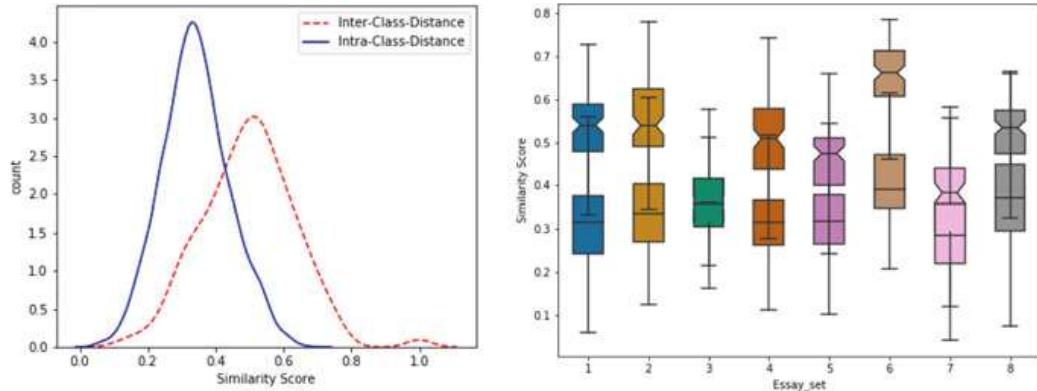


Fig. 2 TF-IDF: similarity score distribution

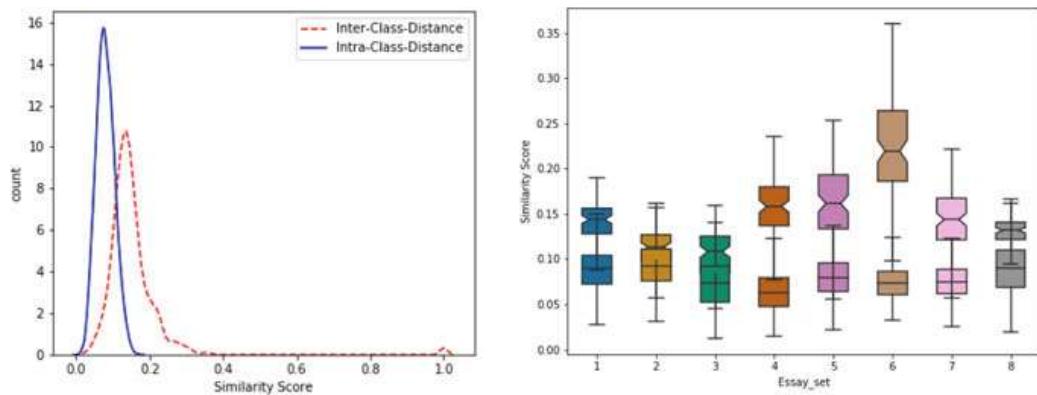


Fig. 3 Jaccard index: similarity score distribution

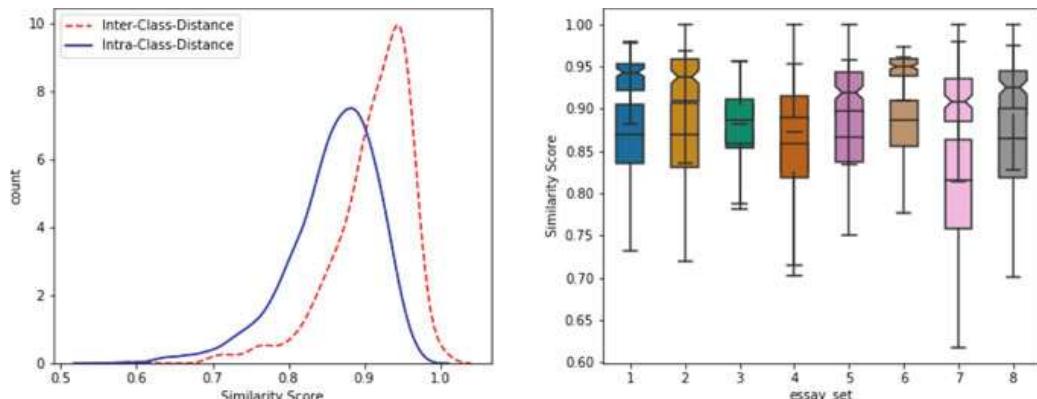
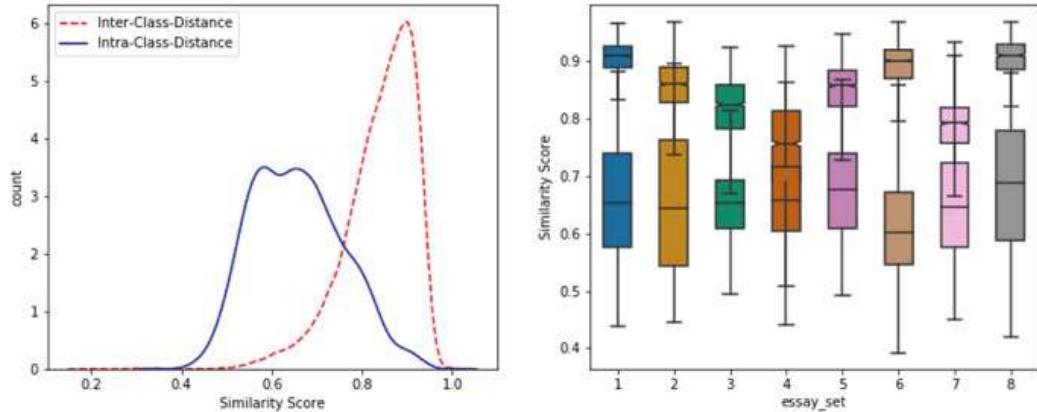
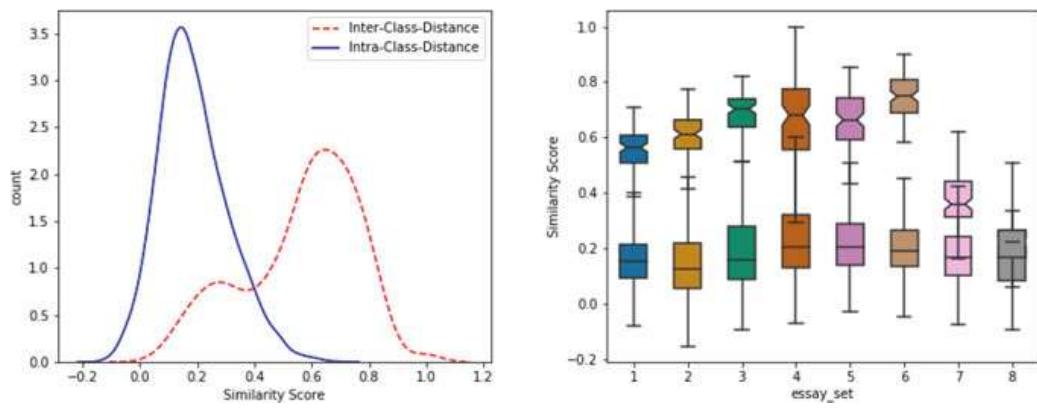
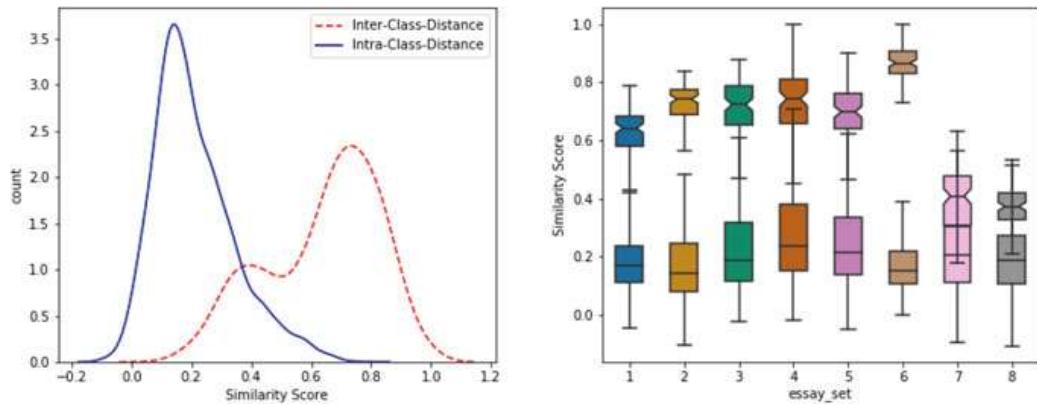


Fig. 4 GloVe: similarity score distribution

**Fig. 5** ELMo: similarity score distribution**Fig. 6** GSE-Lite: similarity score distribution**Fig. 7** GSE-Large: similarity score distribution

plots for intra-class and inter-class distances are well separated. This shows good performance on semantic similarity task.

Performance of Google Sentence encoder Lite and Large are shown in Figs. 6 and 7, respectively. Both the sentence encoder methods perform well on semantic

Table 1 Intra-class and inter-class separation

Method	GSE-Large	ELMo	Jaccard	TF-IDF	GSE-Lite	GloVe
d-prime	2.8375	2.1527	1.6013	1.2434	1.2349	0.9271

similarity task. We can see the distribution plots in which overlap between inter-class and intra-class score distributions is less. Also the box plots for maximum essay sets show the clear separation between inter-class and intra-class similarity scores. GSE-Large takes more memory and time to compute the embeddings as compared with GSE-Lite. But improvement in performance is not that significant. Results of both the GSE-Large and GSE-Lite are almost similar.

5.2 D-Prime

D-prime quantifies the separation between two probability distributions. D-prime is calculated as given in Eq. (2),

$$d' = \frac{\sqrt{2}|\mu_{\text{IntraClass}} - \mu_{\text{InterClass}}|}{\sqrt{\sigma_{\text{IntraClass}}^2 + \sigma_{\text{InterClass}}^2}} \quad (2)$$

where μ is the mean and σ^2 is the variance of similarity score distributions. Table 1 shows the d-prime values for all the methods used for evaluation. By observing distribution and box plots, we could not distinguish between the performance of GSE-Large and GSE-Lite. But by observing the d-prime values, we can see that GSE-Large performs best as compared with other methods including GSE-Lite. The traditional methods like TF-IDF and Jaccard index give better performance than GloVe embedding method. GSE-Large with d-prime value of 2.8375 has the best separation between intra-class and inter-class semantic similarity distance scores. GloVe with d-prime value of 0.9271 has the least separation between similarity score, which denotes that it could not distinguish between the essays from same set and from different sets.

6 Conclusion

In this research paper, an attempt is made to compute the similarity between two sets of large texts. Different neural embedding-based techniques have been employed for finding the semantic similarity between the essay text data. We have calculated the similarity between texts by using classical methods like TF-IDF and Jaccard index. We have also employed advanced deep learning based methods including ELMo,

GloVe, and Google Sentence Encoder. Embeddings given by ELMo and Google Sentence encoder gives good results as compared with other methods. GSE-Large with d-prime value of 2.8375 gives good performance by distinguishing text from same set and different set. Though simple and basic, the TF-IDF and Jaccard index also shows performance comparable to advanced deep learning methods. Performance given by GloVe is not good as compared with other methods. This study shows the efficacy of the neural embedding techniques in semantic similarity checking task. It also gives good indication about which technique to use when we have large text data.

References

1. Cambria, E., White, B.: Jumping NLPP curves: a review of natural language processing research. *IEEE Comput. Intell. Mag.* **9**(2), 48–57 (2014)
2. Cer, D., Yang, Y., Kong, S.Y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal Sentence Encoder. arXiv preprint [arXiv:1803.11175](https://arxiv.org/abs/1803.11175) (2018)
3. Clark, E., Celikyilmaz, A., Smith, N.A.: Sentence movers similarity: automatic evaluation for multi-sentence texts. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2748–2760 (2019)
4. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural Language Processing: State of the Art, Current Trends and Challenges. arXiv preprint [arXiv:1708.05148](https://arxiv.org/abs/1708.05148) (2017)
5. Melamud, O., Goldberger, J., Dagan, I.: context2vec: Learning generic context embedding with bidirectional LSTM. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 51–61. Association for Computational Linguistics, Berlin, Germany (2016). <https://doi.org/10.18653/v1/K16-1006>
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
8. Pawar, A., Mago, V.: Challenging the boundaries of unsupervised learning for semantic similarity. *IEEE Access* **7**, 16291–16308 (2019)
9. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
10. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of NAACL (2018)
11. Prize, A.S.A.: The Hewlett Foundation: Automated Essay Scoring (2012). <https://www.kaggle.com/c/asap-aes/>
12. Tashu, T.M., Horváth, T.: Pair-wise: automatic essay evaluation using word mover's distance. *CSEDU* **1**, 59–66 (2018)
13. Wang, B., Wang, A., Chen, F., Wang, Y., Kuo, C.C.J.: Evaluating Word Embedding Models: Methods and Experimental Results. arXiv preprint [arXiv:1901.09785](https://arxiv.org/abs/1901.09785) (2019)
14. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **13**(3), 55–75 (2018)
15. Zhu, G., Iglesias, C.A.: Computing semantic similarity of concepts in knowledge graphs. *IEEE Trans. Knowl. Data Eng.* **29**(1), 72–85 (2016)

Big Data and Machine Learning Analytics to Detect Epileptic Seizures with Minimum Delay Using Random Window Optimization



S. Sanila and S. Sathyalakshmi

Abstract Among the main techniques for the identification and diagnosis of epileptic seizures, electroencephalography, commonly referred to as EEG, remains the most widely used. The EEG data that is collected from the numerous patients who undergo this medical procedure is now considered to be part of big data as we know it. EEG analysis and detection of a data are finding numerous applications in today's world, and few dedicated algorithms have already been developed to deal with this data. This study is aimed at developing a lightweight algorithm, with the sole purpose of comparing the accuracy in the prediction of the seizures from the EEG data, with data volume reduction. The endeavor is to filter out the essentials and exploit the capabilities of the existing algorithms like SVM and ANN, to distill out an algorithm that can achieve the aim of the project. The method proposes the use of sampling using fixed sized windows, to reduce the volume of EEG data that is handled for analysis, and extraction of the top- K amplitude measure from each of these samples. From the windows of 2 s each, top 60 values are sorted out for further analysis. These top- K values, along with the extracted features from the selected EEG, are then fed into the classification algorithms, with the aim of predicting whether it is a seizure or not. The results are promising and found out that ANN algorithm prediction accuracy is better than SVM.

Keywords Big data · Artificial intelligence · Machine learning · Classification · SVM · ANN

S. Sanila (✉) · S. Sathyalakshmi
Hindustan Institute of Technology and Science, Chennai, India
e-mail: ssanila@gmail.com

S. Sathyalakshmi
e-mail: slakshmi@hindustanuniv.ac.in

1 Introduction

Nowadays, most records of everyday happenings are captured as electronic data through devices connected to the World Wide Web, unlike the years past where relevant data was available at very low frequency. Though the volumetric flow of this data is now high, useful data is still sparse and too widely spread out. When the volume of the data exceeds the capabilities of the commonly used computing systems and algorithms, they fall within the bracket of big data. One needs to purchase this data, organize them properly, and then commence the analysis. Analysts and practitioners have always been keen on the refinement and exploitation of the data, but the highly dynamic nature, the heterogeneity, and the voluminous and continuous nature of this data strain the capabilities of current storage management and computation [1]. Doing this is achieved using big data analytics, machine learning application, or deep learning of which requires some theoretical knowhow and a lot of experience in the actual design of qualitative strategies.

Any algorithm that aims to tackle data streams should be capable of using limited resources of time and memory, and deal with the data's nature to change its distribution with the passage of time. In data stream mining, there are three main areas of interest, namely accuracy, amount of space (computer memory) needed, and the time required in making learned predictions. All these areas are symbiotic in nature, hence adjusting the time and space used by an algorithm can influence accuracy [2]. If we gave the algorithm the luxury of referring to more preprocessed data and look up tables, the space occupied increases but the time needed for computing decreases. In short, the concept of green computing which advocates the study and practice of using computing resources efficiently is the need of the hour.

The advancements in the analytics era make it easier, faster, and more precise to analyze the data generated from EEG recordings. This is the future and offers a great example of the convergence of multiple aspects of science [3]. EEG can be partitioned as intracranial and scalp EEG based on the placement of electrodes on different positions on the scalp. Of these, scalp EEG is sensitive to noise. So, an efficient false detection removal method needs to be implemented. Different types of parallel and distributed frameworks are available to solve the complexity issues related to big data analytics. Message passing interface and Open MP APIs are available and these are all about making easy to write, shared memory multiprocessing programs during the multicore processing era. But high throughput and fault tolerance, which are the requirements of our analysis, are not available with these processing frameworks [4].

2 Big Datasets and Machine Learning

The new datasets have three main characteristics that distinguish and distance them from traditional data and these are their volume, velocity of generation, and variability. These three main areas of interest are very much independent; for example,

if the space and the time occupied by an algorithm are changed, then it will end up affecting accuracy [5]. Processing of big dataset includes partitioning of this large amount of data into chunks for seed up processing. Initially, batch processing is performed on big data. But due to the properties of this huge data, arrival time and processing speed of data streams may vary.

Methods to analyze big and alternative datasets include not only traditional statistical methods, but also machine learning techniques like supervised learning, (regressions, classifications), unsupervised learning (factor analysis, clustering), deep and reinforcement techniques, etc. Today's business enterprises and industries are operating in a highly competitive arena. One of the tools for getting an edge over the competition is the harnessing of big data streams using new generation technologies. The speed in processing the datasets through cutting down of the waiting time between a query and a response can mean better profits through better and faster decisions [6].

Intelligence can be said to be the ability to assess and make sense of things by itself and to appreciate that which other systems have understood. Artificial intelligence aims to develop concepts that enable computers to perform human task, in a far superior manner. Basically, AI gives machines a step toward cognitive interpretations that only humans had before. The concept of AI also aids us in understanding how natural intelligence works [7]. Machine learning achieves this cognitive intelligence nowadays, through the concept of deep learning (DL). DL has achieved spectacular results in the fields of image and pattern recognition [8].

3 Epileptic Seizure Detection

EEG is the procedure used to detect and record the electrical activity in the brain using multiple electrodes attached to the scalp. This data is always recorded as a multichannel time series, with range extending from 128 to 2000 samples per second per channel. Isolating and recording the electrical signals of a single neuron are extremely difficult [9]. As such, the EEG records the activities of several neurons together. Each EEG channel is then calculated by measuring the difference between that channel and one or two of the reference electrodes. In the event of a seizure, it is represented in the EEG by high-amplitude signals that are a clear indication of prospective seizures. However, it will be incorrect to infer that all the high-amplitude signals seen in the EEG are signs of seizure. Few of the features that need to be considered in seizure detection are:

1. Large variability in seizure activity exists among individuals.
2. Different occurrences of seizure activity in individuals at different times will have similarity in the seizure pattern.

4 Related Work

As of date, there does not exist an algorithm that refines and combines the advantages of extracted features along with separate sorted amplitude measure and then fed these into SVM and ANN algorithms. Previous studies have implemented Ensemble Empirical Method Decomposition (EEMD) using MapReduce, but this remains data intensive as well as computationally intensive. But this works very well in Hadoop MapReduce [10]. The latency and throughput of this study were promising, but it lacked stability.

Discrete wavelet transform (DWT) was used in big data analytics early days using MapReduce technique. Wavelet transform for processing EEG has been pursued in automatic seizure detection due to its ability to obtain non-stationary time frequency analysis of signals. Many studies have utilized wavelet and wavelet packet transform-based features with varying success in automated seizure detection. Wang et al. implement a parallel version---ensemble empirical mode decomposition algorithm using MapReduce [9, 11]. Of these two, it was noted that EEMD performs better. It is an innovative method for processing EEG signals. But it is highly computational and data intensive. Parallel EEMD also served to verify the scalability of Hadoop MapReduce.

Dutta et al. use Hadoop and HBase distributed storage for multidimensional EEG analysis. These two technologies combined and executed to provide results which are promising in terms of latency and throughput. Most of the detection methods in literature use machine learning techniques like artificial neural network (ANN) to classify the extracted features into seizure and non-seizure occurrences [12]. Shoeb et al. studied and present a patient-specific automated seizure detection method that uses support vector machine.

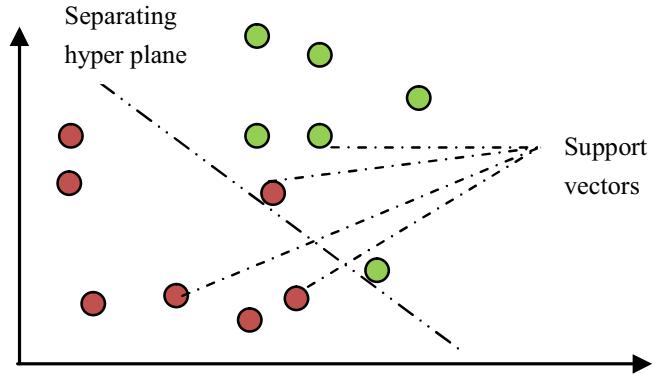
5 Methods

5.1 Support Vector Machine

SVM is a supervised machine learning algorithm used for classification challenges. After plotting each data item as a point in n dimensional space, where n is the number of features [13], classification is performed by finding the hyperplane that separates the two classes, as shown in Fig. 1.

Classification algorithm such as SVM can be implemented in Python using scikit-learn. After importing library files, we must create classification objects and labels.

$$\text{Sensitivity} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}} \quad (1)$$

Fig. 1 Hyperplane in SVM

$$\text{Specificity} = \frac{\sum \text{TN}}{\sum \text{FA} + \sum \text{TN}} \quad (2)$$

$$\text{Accuracy} = \frac{\sum \text{TP} + \sum \text{TN}}{\sum \text{TP} + \sum \text{FA} + \sum \text{TN} + \sum \text{FN}} \quad (3)$$

Equations (1–3) are characteristics of the classification procedure and are inversely related. After classification, we can calculate the accuracy, precision, recall, and F1 score as per the above equations. Its main feature is the use of hyperplane. These measures demonstrate whether SVM algorithm correctly predicts that the given class label is a seizure or not. SVM algorithm outputs such a hyperplane to classify the prescribed labels.

5.2 Artificial Neural Network

According to Howard Rheingold, neural network frameworks are a kind of technology that cannot be classed as algorithms, and deep learning is a branch of machine learning which uses these neural networks. Performance of neural network improves as they grow larger and work with higher volume of data. ANN has many processors. This operates parallel and they are arranged as tiers. The first level receives the input, processes it, and then with weight adjustments, passes it to the successive layers. The last layer processes the final output [14]. The key aspect of ANN is that they are adaptive and learn very quickly. Types of different neural networks are: feedforward, radial basis functional, multilayer perceptron, convolutional, recurrent neural network, and modular neural network.

6 Methodology

Since the EEG is a continuous stream of data, it is essential for its proper analysis, to divide it into small portions. When handling numerical data, it becomes easy to split it into manageable portions and carry out analysis as required. However, in the case of EEG, this is a continuously streaming physiological data which makes analysis difficult. In order to ensure proper handling of this data, the data stream is split or divided into windows of two second intervals (Fig. 2). It is on these two second windows of data that the analysis is effectively carried out.

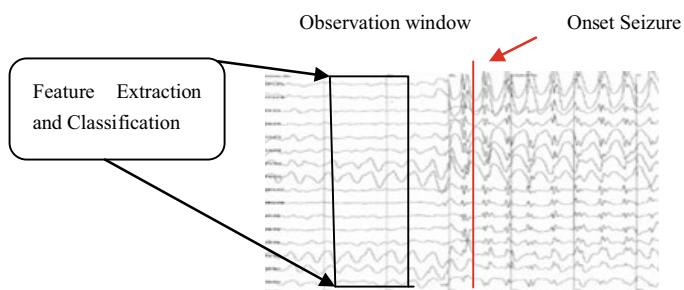
6.1 Top-K Amplitude Measure

During the normal activities of the human brain, the amplitude of the electrical waveforms that are recorded in the EEG will be of even. However, when there occurs a seizure there will be sudden spike in the amplitude of the signals. It is this aspect that makes the process of using amplitude measure, a very good means of seizure detection. This spike in the amplitude can vary to different levels and is represented by the “ k ” values, where the value of K varies depending on the samples selected from each window, and the highest of these values, referred to as the top- k values, are basically the major indicators of seizure.

6.2 Multiwindow Measure

EEG signals are extracted and split into windows of 2 s each. From every window, top- k amplitude values will be calculated and retrieved. For each top- k measure, the following features are extracted: mean, standard deviation, variance, median, percentile, skewness, and kurtosis. For evaluating the seizure detection ability, it is decided to test it for accuracy, precision, recall, and F1 score. Datasets used in our experiment are downloaded freely from CHB-MIT database [15]. These datasets after complete conversion are about 1.54 TB. In our experiment, data from a single patient is taken for training purpose. It contains 23 channels. Each second's data size

Fig. 2 Selecting window of desired size



is about 256 bits. Suppose if we consider 2 windows at a time, there will be 512 bits. First five hundred windows will be analyzed and cumulative mean will be calculated. It will be finally given for classification. Windowing is roughly described in Fig. 2.

For recognizing a single shape, isolated windows are enough, but they cannot recognize a complete pattern from the previous window which is helpful for the detection of seizures. So, the activity from previous windows is also extracted for the better detection accuracy. The window size is decided as per the rate of false positives. Large values of windows will minimize the false positives but increase latency as well.

In this work, mainly modified or combined properties of support vector machine and artificial neural network are being applied. The classifications are compared, and the aim is to find which method will accurately classify the seizure signal from the non-seizure one.

6.3 Workflow

- Step 1: Read the EEG signal
- Step 2: Split the signal into windows of 2 s
- Step 3: Select top- k values from every window
- Step 4: Extract features for the extracted top- k values
- Step 5: Combine along with features
- Step 6: Apply classifier
- Step 7: Check whether classification is successful.

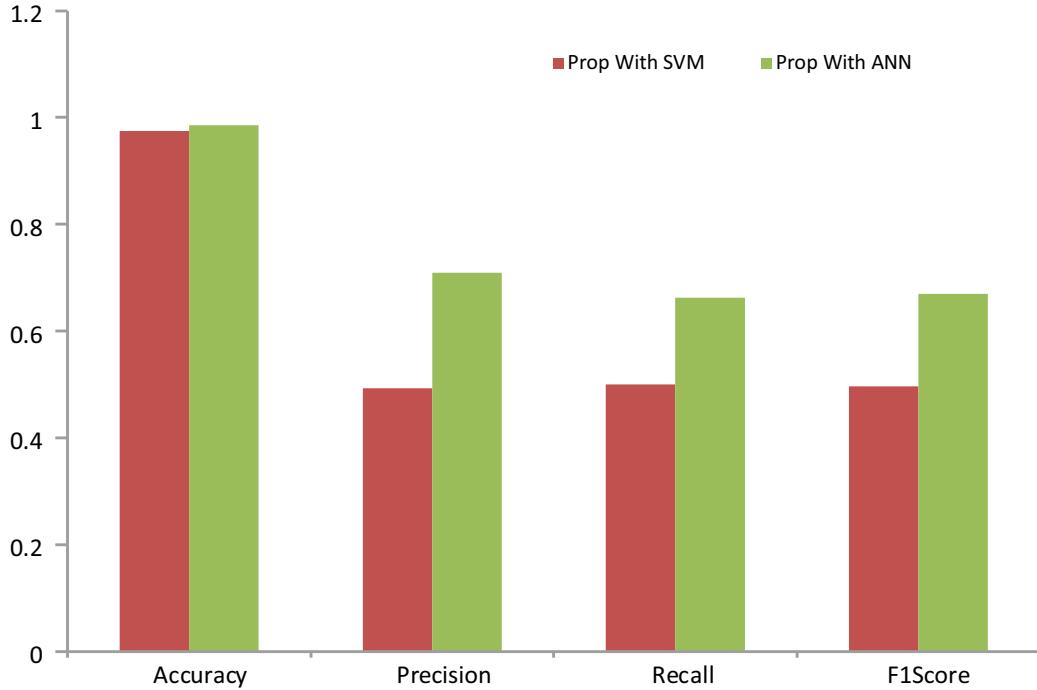
The act of writing software routines with a rigid set of instructions, solely aimed at accomplishing a task is to be replaced with the concept of machine learning. Machine learning creates algorithms with dexterity and ability to poach required data, learn from the data, and then give an output that decides and prediction of something in the world.

7 Results and Discussion

The proposed method and workflow are explained. In an experimental scenario, our SVM and ANN algorithm for classification using random window top- k measure was able to correctly classify seizure data. Table 1 represents a comparison between the results obtained after the classification of multichannel dataset, originating from a single patient. As evident below SVM lacks behind ANN in terms of the defining parameters. Graphical representation of this result is shown in Fig. 3.

Table 1 Comparative results

	Prop with SVM	Prop with ANN
Accuracy	0.975111	0.985556
Precision	0.493056	0.709121
Recall	0.5	0.662458
F1 score	0.496499	0.669669

**Fig. 3** Graphical representation of results

8 Conclusion

The above explains the comparison of two predominant algorithms for epileptic seizure detection, both of which were fed with the exact same multichannel EEG dataset. The workflow and architecture involved are explained. The data reduction is achieved with the help of windowing, and the latency reduction is done by the introduction of a new top-k amplitude feature. The future aim is to scale up the experimental algorithm by modifying the capabilities toward implementing it on the deep learning paradigm. This will enable the algorithm to simultaneously deal with multiple multichannel datasets, originating from different patients.

References

1. Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data mining with big data. *IEEE Trans. Knowl. Data Eng.* **26**(1), 1041–4347 (2013)
2. Domingos, P., Hulten, G.: Mining high speed data streams. In: Proceedings of Sixth ACM SIGKDD International Conference, Knowledge Discovery and Data Mining (KDD'00), pp. 71–80 (2000)
3. Tzallas, A.T., Tsipouras, M.G., Fotiadis, D.I.: Epileptic seizure detection in EEGs using time-frequency analysis. *IEEE Trans. Inf. Technol. Biomed.* **13**(5), 703–710 (2009)
4. Ahmed, L., Edlund, Å., Laure, E., Whitmarsh S.: Parallel real time seizure detection in large EEG data. In: Conference Paper, Research gate Proceedings of the International Conference on Internet of Things and Big Data (IoTBD 2016), pp. 214–222 (2016). ISBN: 978-989-758-183-0
5. Herland, M., Khoshgoftaar, T.M., Wald, R.: A review of data mining using big data in health informatics. *J. Big Data* **1**(1), 1–35 (2014)
6. Zhou L, Pan S, Wang J, Vasilakos AV, Machine learning on big data: opportunities and challenges. *Neurocomputing*. <http://dx.doi.org/10.1016/j.neucom.2017.01.026>
7. L'heureux, A., Grolinger, K., Elyamany, H.F., Capretz, M.A.: Machine learning with big data: challenges and approaches. *IEEE Access* **5** (2017)
8. Zhang L, Tan J, Han D, Zhu H (2017) From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Disc. Today* **22**, 1680–1685 (2017) (Elsevier, Informatics)
9. Vidyaratne, L.S., Iftekharuddin, K.M.: Real-time epileptic seizure detection using EEG. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**(11):2146–56. <https://doi.org/10.1109/tnsre.2017.2697920>
10. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Commun. ACM* **51**(1), 107–113 (2008)
11. Ayoubian, L., Lacoma, H., Gotman, J.: Automatic seizure detection in SEEG using high frequency activities in wavelet domain. *Med. Eng. Phys.* **35**(3), 319–328 (2012)
12. Biswas, D., Hossain, M.F.: Epileptic seizure detection based on selected features of different complexities using ANN. In: International Conference on Electrical Information and Communication Technology, IEEE Xplore (2018). <https://doi.org/10.1109/eict.2017.8275176>
13. Varatharajan, R., Manogaran, G., Priyan, M.K.: A Big Data Classification Approach using LDA with an Enhanced SVM Method for ECG Signals in Cloud Computing. Springer (2017). <https://doi.org/10.1007/s11042-017-5318-1>
14. Li, M., Chen, W., Zhang, T.: Classification of epilepsy EEG signals using DWT-based envelope analysis and neural network ensemble. *Biomed. Sign. Process. Control* **31**, 357–365 (2017)
15. CHB-MIT scalp EEG database. Available at <http://physionet.org/pn6/chbmit/>

An Efficient Evaluation of Spatial Search on Road Networks Using G-Tree



C. P. Shahina

Abstract Location-based services (LBS) enable users to question the purpose of interest like restaurants, hospitals with various features. Although optimal route calculation in road network supported distance may be a combinatorial optimization problem, with the advancement of varied web mapping services, it is possible to produce travel distance for LBS applications. Road network and site data easily are often represented as a graph during a spatial database. By using different spatial functions like pg-routing Dijkstra or pg-Routing A*, we will efficiently calculate the route from it. But in case of extremely large dataset, searching route and finding nearest neighbors are far more difficult. So here a completely unique indexing method for the road network is proposed. G-Tree indexing is an indexing method that efficiently supports mainly two queries, i.e., k-NN queries, single-pair shortest path query. G-Tree indexing method is inspired by R-Tree. Inspired by R-Tree, G-Tree may be a height-balanced and scalable index, namely G-Tree, to efficiently support these queries. The space complexity of G-Tree is $O(|V|\log|V|)$, where $|V|$ is the number of vertices within the road network. Finding only one shortest path to a point of interest (POI) is not sufficient for many situations like road conditions, traffic, etc. So this paper also suggests the method for calculating k alternating paths to a point of interest (POI) by using Yen's algorithm and G-Tree.

Keywords Single-pair shortest path · KNN searches · Road network · Index · Spatial database · G-Tree index · LBS

1 Introduction

A web map service [1] (WMS) may be a standard protocol for serving geo-referenced map images that a map server generates using data from a geographic data system (GIS) database. Web mapping services like Google Maps [2–9] provide distance and

C. P. Shahina (✉)

Department of Information Technology, Kannur University, Kannur, Kerala, India
e-mail: cp.shahi916@gmail.com

travel time information. It also provides location-based service (LBS). For example, in navigation systems, a user might want to seek out the shortest path between the current location and point of interest (POI). In tourist guide applications, a tourist may search for k-nearest “seafood restaurants” or “where is the nearest petrol pump” while traveling in a city. But the distance between the two points is static. So if it gets efficiently stored, information can be easily provided. G-Tree indexing method is used to store and retrieve road network information.

G-Tree [10] may be a novel indexing method, especially for the road network. The essential idea is to recursively partition the road network into subnetworks and build a tree structure on top of subnetworks where each G-Tree node corresponds to a subnetwork. The most advantage [11, 12] of using G-Tree is it only takes $O(|V|\log|V|)$ [1] as space complexity, and for G-Tree single-pair shortest path, (SPSP) time complexity is $O(|V|)$. In this paper, G-Tree SPSP query is used to find the shortest path and improvised Yen’s algorithm [13] with G-Tree is used for finding the k-shortest path from the current location to POI.

2 Related Work

Paper 1 illustrates server-side spatial mashup service (SMS) that permits the LBS provider to efficiently evaluate k-NN queries in road networks using the route information and time period retrieved from an external web mapping service.

Paper 2–9 illustrates web mapping service to urge location, distance and direction information. Location, distance and direction data is taken from these web services. The query results of web mapping services are either JSON or XML; here, we have taken JSON data and store it in our database. The distance between the two points is static. From different reviewed papers, the multigraph used for road transportation modeling. Road segments are often treated as multiple edges and junction or road endpoints are often considered as a node during a multigraph. For single-pair shortest path, G-Tree uses an assembly based algorithm to compute the shortest path distance between two vertices with $O(|V|)$ time complexity. Paper 11–13 is predicated on finding the first shortest route (one) during a road network. Paper 11 illustrates Dijkstra’s algorithm to seek out the shortest route, the most drawback of this is its high time complexity. In paper 12, R-Tree is to create an efficient tree that on one hand is balanced on the opposite hand the rectangles do not cover an excessive amount of empty space and do not overlap an excessive amount of. In paper 12, the road is split into small groups (transit nodes) and computes the space between these nodes. But here performance is poor and space complexity is high. Paper 13 illustrates a replacement technique to seek out the shortest path. This system is far better than Dijkstra’s algorithm and transit method, and therefore, the space complexity is low. Paper 14 describes Yen’s algorithm for finding k-shortest path.

3 Proposed Method

In this paper, the road network is modeled as an undirected weighted graph $G(V, E)$, where V is a set of vertices and E is a set of edges [1]. This section is further divided into G-Tree construction, SPSP search and k-shortest path to a POI.

3.1 G-Tree Indexing for Finding SPSP Based on Distance

3.1.1 G-Tree Construction

The road graph is partitioned into f equal-sized subgraphs and it will be the children of the root graph. Then recursively partition each subgraph into f equal-sized subgraphs until each child does not have more than T vertices [10]. Figure 1 shows the road graph partition, assuming f as 2 and T as 4.

Figure 1 shows how G-Tree is formed from the partitioned graph [10, 13]. In non-leaf subgraph, rows and columns are borders of that subgraph, and for each leaf, subgraph rows are borders and columns are nodes of that subgraph. For example,

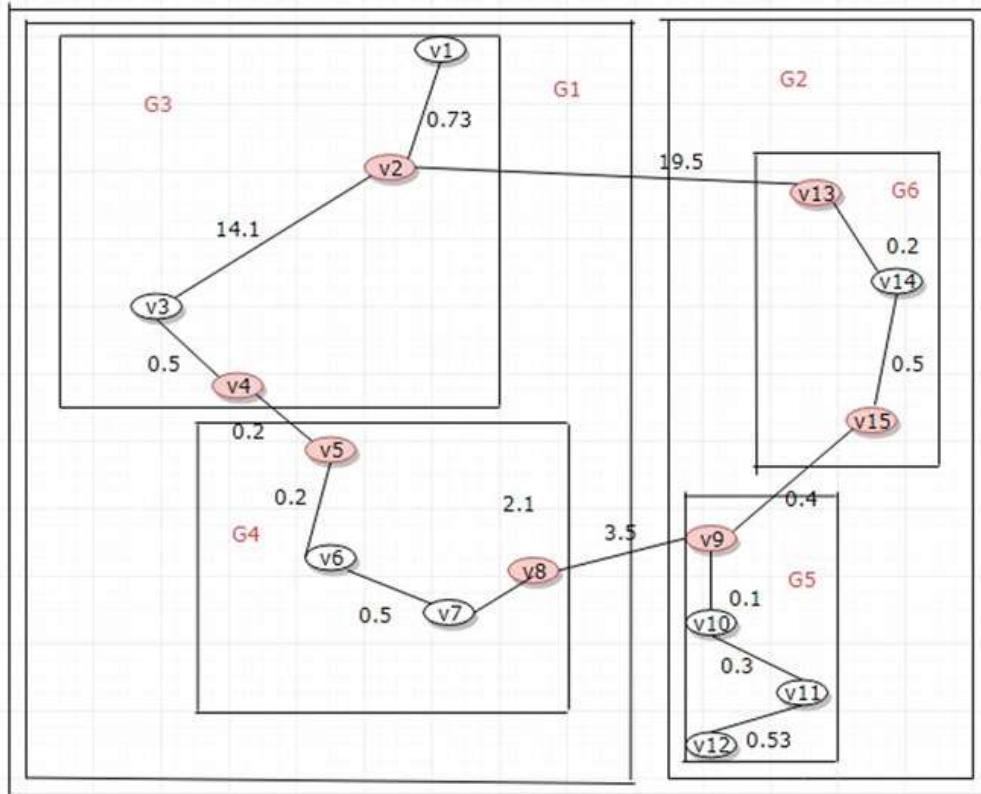
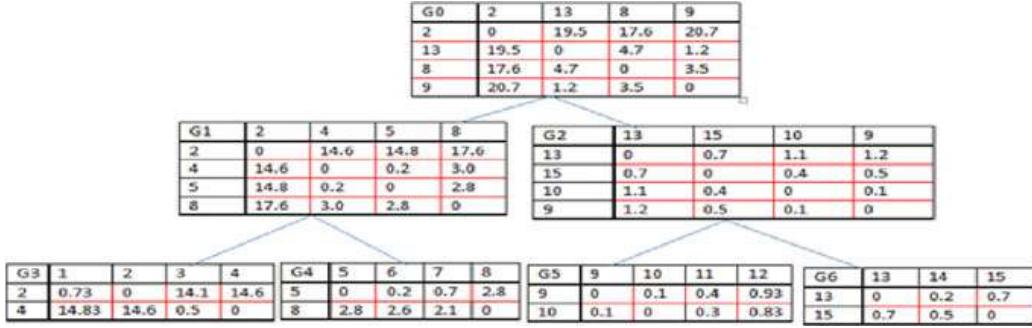


Fig. 1 Road network and graph partition

**Fig. 2** G-Tree

G_0, G_1, G_2 are the non-leaf subgraph. Borders of G_1 are 2, 4, 5, 8 and 13, 15, 10, 9 are the borders of G_2 . Leaf subgraphs are G_3, G_4, G_5 , and G_6 . Borders of G_3 are 2 and 4 and nodes of G_3 1, 2, 3, 4 similarly borders of G_4 are 5, 8 and nodes of the G_3 are 5, 6, 7, 8.

Figure 2 shows how G-Tree is formed from the partitioned graph. For each non-leaf subgraph, their vertices are borders of their children, and for each leaf subgraph, their vertices are their nodes. In the example, G_0, G_1, G_2 are non-leaf subgraphs and G_3, G_4, G_5, G_6 are leaf subgraphs. For each non-leaf graph, the distance between borders is stored, and for each leaf graph, all the nodes are represented as columns and rows are represented as borders of that graph. Separate stored procedures are developed in the application to partition nodes and for G-Tree creation while inserting the road network graph automatically.

3.1.2 Single-Pair Shortest Path (SPSP) Query

Single-pair shortest path query, $\text{SPDist}(u, v)$ identifies the shortest distance between two locations say u and v in a road network. G-Tree indexing is an indexing method for spatial queries in the road network and it outperforms the traditional methods like Dijkstra's shortest path algorithm both in terms of space and time complexity [10].

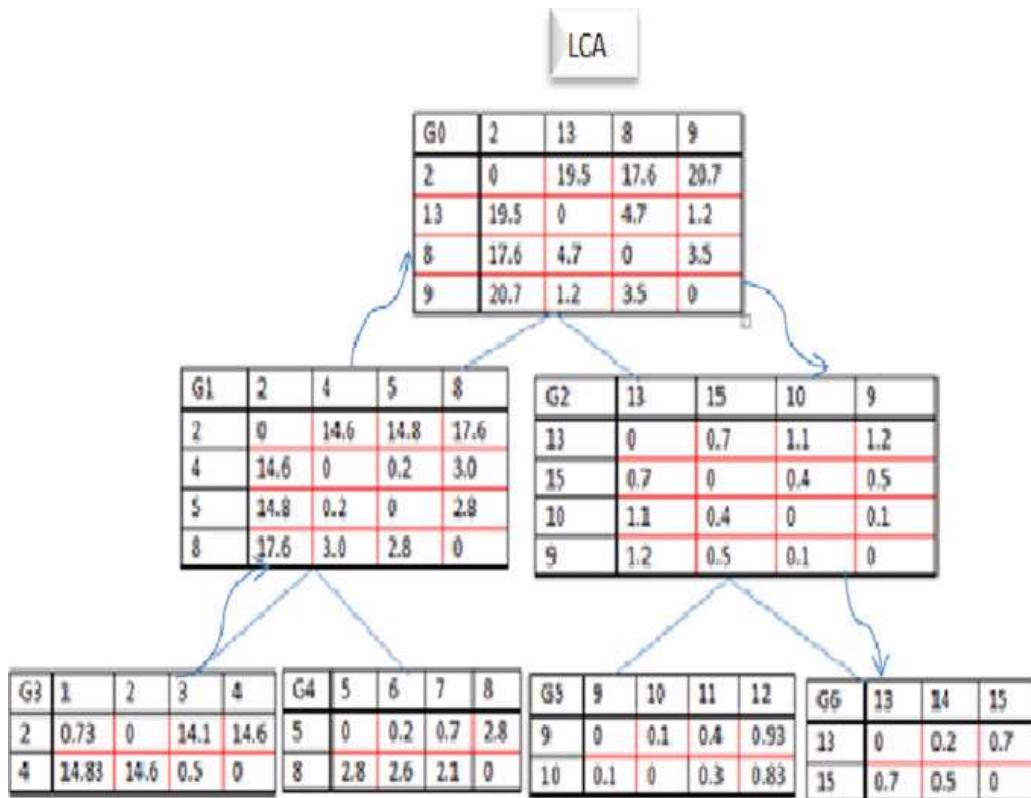
MIN-DIST OUTSIDE GRAPH

Besides considering the path from u to v , u to common border and v to the common border are calculated, i.e., (u, b) and (b, v) are calculated. Then it decomposes one path into m subpaths by m borders, i.e., $(u, b_1), (b_1, b_2) \dots (b_m, v)$, while these m borders belong to m distinct and consecutive G-Tree nodes on the path from the leaf(u) to leaf(v).

For example, consider SPSP between (1, 10). The procedure is as given in Fig. 3.

SPSP ALGORITHM

First, check whether both u and v belong to the same subgraph. If yes, Dijkstra's shortest distance [14] is calculated. Even though both the nodes belong to the same subgraph, sometimes, no direct path is found between those points. Then the border

**Fig. 3** SPSP search

distance is calculated and the shortest of both Dijkstra's and border distance is identified. It will be the shortest path between those points.

If u and v are on different subgraphs, the first step is to find out the subgraphs containing those points. The next step is to find the common parent, i.e., least common ancestor (LCA) of u and v . Then find the distance between v to its borders and take the shortest one. Repeatedly calculate border to border distance until it reaches the least common ancestor between u and v . After getting the path to LCA, find a path from LCA to its child's border (it might be the subgraph, the parent subgraph or the ancestor subgraph of u). Repeatedly calculate the border distances until it reaches the subgraph border of u . Finally, calculate the distance from the border to u . So to summarize, the path from v to LCA and from LCA to u gives the shortest path from u to v . This path will contain u , v and the shortest borders' distances between these two points. In the above example (3, 14), u and v are 10 and 1, subgraphs belong to u and v are G6 and G3, respectively. LCA is G0 and the shortest path is:

1- 2 -13- 15- 10

Algorithm 1: SPSP Search

```

1 Locate leaf(u) and leaf(v) into a temporary table
2 If leaf (u) = leaf(v), Then
3     R =min(BoarderDist(u; v), DijkDist(u, v))
4 Else
5     Find a path N from leaf(u) to leaf(v) on G
      5.1.common_parent_gtree(strt_vertx,end_vertex)           //returns strt_graph and end_graph
      5.2.subgraph_to_node_gtree(strt_graph,strt_vertex)       //returns strt_vertex to strt_graph distance
      border
      5.3.graph_graph_gtree(strt_arr[],border)                 //border is vertex returns from (5.2)
      5.4.parent_root_gtree(strt_arr[],end_arr[],boarder1)    //boarder1 is vertex returns from (5.3)
      5.5 graph_root_gtree(end_arr[],boarder2)                // boarder2 is vertex returns from (5.4)
      5.6subgraph_boarder_node_gtree(end_graph, border 3,end_vertex) // boarder3 is vertex returns from
(5.5)

```

3.1.3 G-Tree Indexing for Finding K-Path Based on Distance

The ideal path is not always the shortest distance path; it also depends on travel time (or congestion). We consider different shortest paths (k-shortest paths) and find travel time for each k-path. Here, we Yen's algorithm is used for finding the k-shortest distance path. Here G-Tree SPSP search is used rather than Dijkstra's algorithm for finding the first shortest path. Time complexity is further reduced by the use of G-Tree in Yen's algorithm.

Path 1: 1-2-13-15-10

Path 2: 1-2-9-10.

4 Experimental Evaluation and Results

This work makes use of J2EE and spatial database PostgreSQL for the implementation process. The SPSP algorithm is implemented using stored procedures created in PostgreSQL and calls this procedure from Java web service. For further optimization of the G-Tree indexing method is implemented for stored procedures.

In SPSP, the algorithm is designed in PostgreSQL stored procedures. A procedure named Common_parent_GTree() is created for finding common parent of corresponding source and destination subgraphs. Then the shortest distance of start node to its subgraph is calculated using the procedure Subgraph_to_node_gtree(strt_graph, strt_vertx). Then the shortest distance from the start subgraph to destination subgraph is calculated using Graph_graph_GTree(strt_array [], border). Then the parent to root distance is calculated using parent_root_GTree(strt_arr[], end_arr[], boarder1). The shortest distance between the destination subgraph and the root is calculated using the stored procedure Graph_root_GTree (end_arr[], boarder2). Shortest distance between destination subgraph to destination node is calculated using the stored procedure Subgraph_boarder_node_GTree(end_graph, boarder3, end_vertex). Temporary tables are used to store intermediate results. Different Tables 1, 2, 3, 4, 5 and 6 produced while implementing G-Tree indexing are shown.

Table 1 Direction table

Id	startlon	startlat	endlon	endlat	the_geom	Distance	Source	Target
1	75.35985	12.03726	12.03713	75.35919	01020000020E6100000020000000809AC 7713132840AFFF84C407D752402E4 A1AED02132840736891EDFC65240	0.73	1	2
12	75.35985	12.03726	12.03713	75.35919	01020000020E6100000020000000809AC 7713132840AFFF84C407D752402E4 A1AED02132840736891EDFC65240	0.73	1	2
11	75.35564	11.86844	11.86886	75.35542	01020000020E6100000020000000834 CD7B8A3BC2740D80FB1C1C 2D6524013769B81DBBC27402D793C2D BFD65240	0.53	7	12
19	75.35564	11.86844	11.86886	75.35542	01020000020E6100000020000000834CD 7B8A3BC2740D80FB1C1C2D6524013769 B81DBBC27402D793C2DBFD65240	0.53	7	12
2	75.35919	12.03713	11.92657	75.35356	01020000020E6100000020000000E4A 1AED02132840736891EDFC652408F B1C9D067DA27401F4B1FBAA0D65240	14.1	2	3
5	75.3523	11.92056	11.91933	75.35169	01020000020E6100000020000000641 AF27453D727402FFDA60B8CD65240 0ECC6F6479B2D627402F36AD1482D65240	0.2	4	5
10	75.35867	11.86848	11.86844	75.35564	01020000020E6100000020000001BB73 DE6A8BC2740D6A88768F4D65240 834CD7B8A3BC2740D80FB1C1C2D65240	0.3	6	7
3	75.35356	11.92657	11.92271	75.35221	01020000020E6100000020000008FB1 C9D067DA27401F4B1FBAA0D65240 1B6E765A6DD827402A15F99A8AD65240	0.5	3	8

(continued)

Table 1 (continued)

Id	startlon	startlat	endlat	endlon	the_geom	Distance	Source	Target
4	75.35221	11.92271	11.92056	75.3523	01020000020E6100000020000001B6E 765A6DD827402A15F99A8AD65240 641AF27453D727402FFDA60B8CD65240	0.2	8	4
6	75.35169	11.91933	11.91529	75.35134	01020000020E6100000020000000EC6 F6479B2D627402F36AD1482D65240 EA094B3CA0D42740DEF0715C7CD65240	0.5	5	9
7	75.35134	11.91529	11.89824	75.3495	01020000020E610000002000000EA094 B3CA0D42740DEF0715C7CD65240 96D3F94BE6CB27402F55C4445ED65240	2.1	9	10
8	75.3495	11.89824	11.86989	75.35952	01020000020E61000000200000096D3F9 4BE6CB27402F55C4445ED65240 2789809E61BD27404C07A17202D75240	3.5	10	11
9	75.35952	11.86989	11.86848	75.35867	01020000020E6100000020000002789 809E61BD27404C07A17202D75240 1BB73DE6A8BC2740D6A88768F4D65240	0.2	11	6
13	75.35919	12.03713	11.88501	75.37263	01020000020E6100000020000002E4A1AD 02132840736891EDFCDF65240966AB0 BA1FC52740DEDADA2EDD9D75240	19.5	2	13
14	75.37263	11.88501	11.875	75.36471	01020000020E610000002000000966A B0BA1FC52740DEDADA2EDD9D752404DF 96B0D00C0274050E50F6157D75240	1.6	13	14
15	75.36471	11.875	11.87323	75.36513	01020000020E6100000020000004DF 96B0D00C0274050E50F6157D75240 7DE5E6C017BF2740B8DCAA355ED75240	0.2	14	15

(continued)

Table 1 (continued)

Id	startlon	startlat	endlat	endlon	the_geom	Distance	Source	Target
16	75.36513	11.87323	11.87125	75.36091	01020000020E6100000020000007DE5E 6C017BF2740B8DCAA355ED75240796231 4514BE27405E6ADF3719D75240	0.5	15	16
17	75.36091	11.87125	11.86848	75.35867	01020000020E610000002000000796231 4514BE27405E6ADF3719D752401BB 73DE6A8BC2740D6A88768F4D65240	0.4	16	6
18	11.86848	11.86848	11.86844	75.35564	01020000020E6100000020000001BB73D E6A8BC27401BB73DE6A8BC2740834C D7B8A3BC2740D80FB1C1C2D65240	0.3	17	7

Table 2 Graph border--border table

Id	strt_boarderid	end_boarderid	Distance
1	2	13	19.5
2	2	4	14.9
3	4	5	0.2
4	5	8	2.8
5	8	9	3.5
6	9	10	0.1
7	10	15	0.4
8	15	13	0.7
9	13	2	19.5
10	4	2	14.9
11	5	4	0.2
13	10	9	0.1
15	8	5	2.8
12	9	8	3.5
14	13	15	0.7
16	15	10	0.4

Table 3 Graph vertices table

Id	Graphid	Verticeid
850	936	4
849	936	3
848	936	2
847	936	1
852	937	6
854	937	8
851	937	5
853	937	7
857	938	11
858	938	12
856	938	10
855	938	9
860	939	14
859	939	13
861	939	15

Table 4 Graph table

Id	Graphid	Verticeid
1	936	2
2	936	4
3	937	5
4	937	8
5	938	9
6	938	10
7	939	13
8	939	15

Table 5 Graph border table

Id	Name	parentId
933	G0	0
934	G1	933
935	G2	933
936	G3	934
937	G4	934
938	G5	935
939	G6	935

Table 6 Vertices table

Id	Latitude	Longitude
1	12.0372579	75.3598491
2	12.0371317	75.3591875
3	11.9265733	75.35356
4	11.9227093	75.3522098
5	11.9205586	75.3522977
6	11.9193304	75.3516895
7	11.915285	75.3513404
8	11.8982414	75.3495037
9	11.8698854	75.3595244
10	11.8684761	75.3586675
11	11.8684366	75.355637
12	11.8688622	75.3554185
13	11.8850077	75.3726308
14	11.8750004	75.3647082
15	11.8732281	75.3651251
16	11.8712484	75.3609142

5 Conclusion

This paper proposed method an efficient and scalable index on road networks. This illustrates an assembly based method to efficiently calculate shortest path distance. Using the G-Tree indexing single-pair shortest path algorithm. Experimental results show G-Tree theoretical and practical superiority over state-of-the-art methods. Searching made easily in large datasets having road networks because of the splitting of large road graphs into much smaller subgraph. K-shortest path is calculated using Yen's algorithm. Besides of Dijkstra's algorithm for finding first shortest path in Yen's algorithm, it is calculated using G-Tree SPSP algorithm. So searching made easy in K-shortest path method also.

References

1. Zhang, D., Chow, C.Y., Li, Q., Zhang, X., Xu, Y.: A spatial mashup service for efficient evaluation of concurrent k-NN queries. *IEEE Trans. Comput.* **65**(8), 2428–2442 (2016)
2. Google Maps: <http://maps.google.com>
3. MapQuest Maps: <http://www.mapquestapi.com>
4. Microsoft Bing Maps: <http://www.bing.com/maps>
5. Yahoo! Maps: <http://maps.yahoo.com>
6. The Google Distance Matrix API: https://developers.google.com/maps/documentation/distance_matrix
7. The Google Directions API: <https://developers.google.com/maps/documentation/directions>
8. MapQuest Directions Web Service—Route Matrix: <http://www.mapquestapi.com/directions>
9. The Google Places API: <https://developers.google.com/places>
10. Zhong, R., Li, G., Tan, K.-L., Zhou, L., Gong, L.: G-Tree: an efficient and scalable index for spatial search on road networks. *IEEE Trans. Knowl. Data Eng.* **27**(8), 2175–2189 (2015)
11. Guttman, A.: R-Trees. A Dynamic Index Structure For Spatial Searching. <http://pages.cs.wisc.edu/>
12. Bast, H., et al.: Transit—Ultrafast shortest path queries with linear-time pre-processing. In: Proceedings 9th DIMACS Implémentation Challenge, pp. 175–192 (2006)
13. Yen, J.Y.: Finding the K shortest loop less paths in a network. *Manage. Sci.* **17**(11), 712–716 (1971)
14. Barbehenn, M.: A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. *IEEE Trans. Comput.* **47**(2), 263 (1998)

Investigation into the Efficacy of Various Machine Learning Techniques for Mitigation in Credit Card Fraud Detection



S. R. Lenka, M. Pant, R. K. Barik, S. S. Patra, and H. Dubey

Abstract An effective credit card fraud detection model is the most challenging issue for the financial organizations. Statistical and machine learning (ML) techniques are widely explored in financial applications. But there is no thumb rule which technique gives better performance. Recent studies conclude that ensemble learning may be the right approach in this problem domain. In this paper, we aim to develop a novel fraud detection system using an ensemble model. In the proposed model, initially the imbalanced credit card dataset is balanced using random undersampling technique, then the performance of the model is evaluated using both single base classifiers and ensemble of classifiers. In the proposed model, AdaBoost, random forest (RF), extreme gradient boosting and gradient boosting decision tree (GBDT) are used as ensemble models. The experimental result shows that the combination of RF and GBDT for the ensemble model is superior in performance as compared to other combinations.

Keywords Credit card · Fraud detection · Majority voting · Ensemble model · Imbalanced ratio

S. R. Lenka

Department of CSE, Trident Academy of Technology, Bhubaneswar, India

e-mail: sudhansulenka2000@gmail.com

M. Pant

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

e-mail: meenakshikandpal14@gmail.com

R. K. Barik · S. S. Patra (✉)

School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India

e-mail: sudhanshupatra@gmail.com

R. K. Barik

e-mail: rabindra.mnnit@gmail.com

H. Dubey

Electrical Engineering, The University of Texas at Dallas, Richardson, USA

e-mail: harishchandra.dubey@utdallas.edu

1 Introduction

With the rapid growth in technologies, the number of fraudulent activities through the use of credit card transactions also increases. A credit card transaction becomes an easiest way for the illicit people to commit frauds. It becomes very difficult to detect those frauds because the fraudsters always use different methods to commit fraud. The financial losses due to such fraudulent transactions not only affect the card issuing banks and merchants but also the card holders. Recently, fraud detection becomes one of the major challenges in the field of financial applications. An automated fraud detection model is needed that not only effectively detect fraud but it should be cost effective [1]. Machine learning (ML) techniques can be used to build a cost-effective fraud detection model.

An effective fraud detection model (FDS) must not raise too much false alarm, i.e., predicting the genuine transactions as fraud should be reduced and must handle imbalanced class distributions. In credit card dataset, the class distributions are not in equal proportions. The fraudulent transactions are generally very few in numbers as compared to genuine [2]. In such imbalanced class distributions, minority class instances need to be classified correctly. Normally, ML algorithms are able to classify most of the majority (negative) class instances correctly but fail to achieve better accuracy in case of minority (positive) class instances. But FDS needs to achieve high degree of accuracy for the positive class instances rather than negative instances. Another major challenge with imbalanced class distribution is the performance evaluation of classifiers. Accuracy and error rate are irrelevant measures to evaluate the performance of the imbalanced dataset. These two metrics do not justify the performance of the model since the probabilities of both the classes are unequally distributed.

Data sampling is most widely used to handle the imbalanced class distributions. Sampling technique either under-samples the number of genuine transactions or oversamples the number of fraudulent transactions to make class distribution balanced.

The classification model for the FDS can be build either by employing single classifier or ensemble of multiple classifiers. It has been experimentally observed that ensemble classifiers perform better than single classifier model [3]. In ensemble approach, multiple models are collectively used in the classification model to build an accurate model. In this paper, an effective ensemble model is designed in order to achieve high predictive result. Initially, the imbalanced credit card dataset is balanced using under-sampling technique. Then, we implement individual classifier as well as ensemble models on the balanced data subsets. Ensemble models are build using AdaBoost and tree base models like random forest (RF), gradient boosting decision tree (GBDT) and extreme gradient boosting (XGBoost). Finally, all these models are trained on the balanced subsets and predicted on the test dataset.

The rest of the paper is organized as follows: Sect. 2 includes the brief descriptions about single and ensemble models. Section 3 provides the work flow of the proposed methodology. Section 4 presents the experimental analysis which includes credit card

dataset, experimental set-up and different evaluation measures. Section 5 provides the results and comparative analysis of different base and ensemble models and finally conclusion and future work are discussed in Sect. 6.

2 Related Works

2.1 Machine Learning Algorithms

A total of six base classifiers are used in the experimental study. Additionally, it includes different sampling techniques to handle imbalanced class distributions. The base classifiers are used in conjunction with different ensemble techniques, like bagging, boosting, random forest and stacking.

2.1.1 Base Classifiers

Decision tree (DT) is most widely used to build the classification models [4]. It is a hierarchical tree like structure with multiple nodes, branches and leaves. Each feature or attribute is represented as a node which is then split into different branches and a class value is assigned at the terminal node. DT splits the samples based on some cut-off values for each attribute and the attribute with minimum cost is selected as the splitting attribute.

K-nearest neighbors (KNN) model is used to predict the instance by finding k -nearest data points. KNN estimates the samples of the test dataset based on the class with highest majority in the k set.

Support vector machines (SVMs) are statistical learning methods implemented successfully in various classification problems. SVMs are very suitable for binary classification problems like fraud detection. SVMs are linear classifiers that converts a linear problem into in a high-dimensional feature space. The main advantage of SVMs working principle is that the nonlinear classification task in the original input space converts to a linear classification task in the high-dimensional feature space.

Naïve Bayes (NB) uses probabilistic rule to build the classifier models. It uses Bayes rule to do the predictions of the samples. Bayes rule computes the probability of class y_i given each attribute A_j and the class with the highest posterior probability is predicted as output.

Artificial neural network (ANN) model consists of interconnected neurons (nodes) which are connected to the next layer of neurons through synaptic weights. Each node takes its input from the previous layer nodes. Each neuron j receives input signal as given below:

$$U_j = \sum W_{ij} * X_j \quad (1)$$

where W_{ij} is the weight associated between neurons i and j and X_i is the input signal to neuron i . If the result is greater than a threshold value, then the neuron j fires and the output of j becomes input for the next layer.

2.1.2 Ensemble Learning Methods

In order to increase the performance of the classification model, ensemble learning methods are employed. Ensemble methods combined the hypotheses of different base classifiers into a better hypothesis and able to make better predictions [5]. Through empirical analysis, it is shown that ensemble learning approaches achieve better results than single classifiers [5]. The most common ensemble methods are bagging, boosting, random forest and stacking.

In bagging, multiple classifiers are trained in parallel on different data samples, each classifier predicts for each test sample, and finally, the final decision is made based on majority voting.

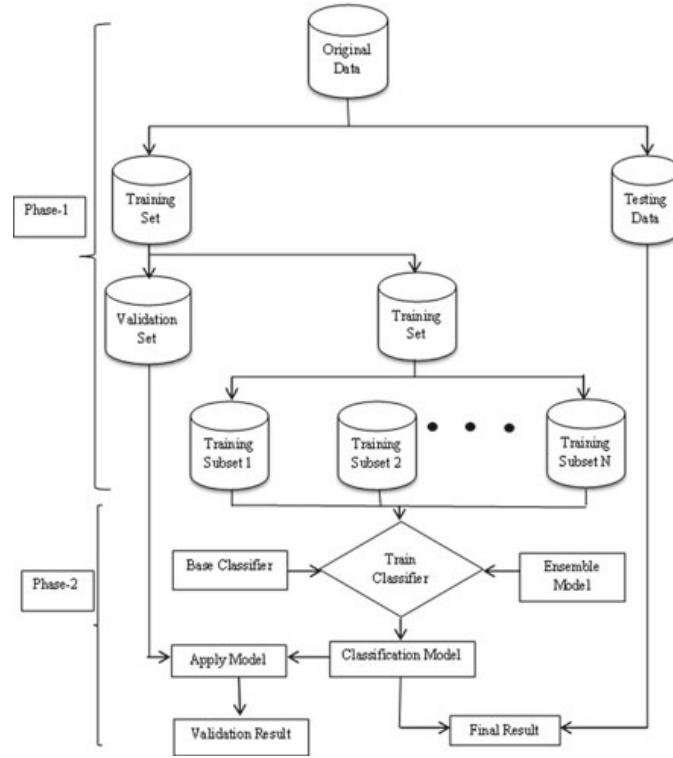
Boosting employs different classifiers on different training dataset. But, in boosting, a combination of weak classifiers are implemented serially rather than parallel. Adaboost, gradient boosting decision tree (GBDT) and extreme gradient boosting (XGBoost) [6] are the different boosting techniques in which DT is used as the base classifier.

In random forest (RF), various DTs are build by randomly picking the samples from the training dataset with replacement. Additionally, each tree in the RF is build by randomly selecting the features. As a result, the model leads to low variance and slightly more bias but the overall performance of the model is accepted.

3 Proposed Methodology

In our proposed methodology, we work on two phases. In the first phase, we deal with imbalanced datasets, and in the second phase, we work on different ensemble approaches by combining multiple classifiers to make the predictions better. Initially, we split the entire dataset in the ratio 75:25, 75% of the data used for training set and 25% for testing set. Then, on the training dataset, we implement 5-fold cross-validation to create training and validation sets. Figure 1 shows the graphical illustrations of the proposed model. Initially, the imbalanced credit card transactions are balanced using under-sampling technique. The training dataset is balanced by dividing into a set of training subsets and each subset must contain equal proportions of fraud (class 1) and normal (class 0) transactions. Each subset is generated by randomly selecting the samples from the majority (normal) class instances and equal proportion of minority (fraud) class instances is evenly contributed to the subsets.

In the second phase, the data subsets are used to train the base classifier C_i . The classifiers C_i are then used to build the ensemble models. In case of bagging, a sample x is predicted based on the majority voting. In case of boosting, the model

Fig. 1 Proposed model

gets trained sequentially by assigning more weights to the misclassified samples. Ensemble models and the base classifiers are validated through the validation set. Finally, the individual base classifiers and ensemble models are evaluated through a series of measures on the testing dataset.

4 Experimental Study

A series of experimental analysis is addressed in order to reflect the performance and robustness of the proposed model. In this section, we focus on three aspects: credit card dataset, design of experiments and performance evaluation metrics.

4.1 Credit Card Dataset

For the experimental analysis, we used the publicly available credit card dataset [7]. It consists of 284,807 credit card transactions made by European card holders. In the entire dataset, only 492 transactions are fraudulent and the remaining's genuine. The dataset consists of 30 features and feature 'Class' is the target with values 1 and 0 referred as fraudulent and genuine, respectively. The class distributions of the credit card dataset are highly imbalanced as the number of fraudulent transactions is very

few as compared to normal transactions. In order to design an effective classification model, the dataset needs to be balanced. Under-sampling approaches are widely used to make the class distribution balanced. In this paper, under-sampling approach is used to tackle the imbalanced class distributions.

4.2 Design of Experiments

Initially, we split the entire credit card dataset in the ratio 75:25. 75% of the dataset used as training set and the remaining as testing set. The 75% of the training set is further split using 5-fold cross-validation. In cross-validation scheme, the algorithm gets trained and validated on each subset. In 5-fold-cross-validation, onefold is used for validation and the remaining 4-folds used for training and we repeat the process five times. In each iteration, different fold is used for validation and the subsequently the other remaining folds for train set. Finally, we obtain the result by averaging five experiments.

For the experimental study, five traditional ML algorithms are used as base classifiers. It includes ANN, DT, KNN and SVM. Additionally, we implement ensemble models like, gradient boosting decision tree, AdaBoost, XGBoost and random forest. The default parameters are set for each base classifiers and ensemble models using 'sklearn' function in python.

4.3 Performance Evaluation Measures

The performance of the fraud detection model must be validated through a series evaluation measures. In our proposed model, we evaluate the performance of the algorithms on the basis of accuracy (ACC), f1_score, Area Under Curve (AUC) score, G-Mean, Matthews Correlation Coefficient (MCC) [8] and Logistic Loss. In binary classification models, the confusion matrix (shown in Table 1) is used to compute different measures. True negative (TN) indicates the number of normal or genuine transactions correctly predicted as normal transactions. False positive (FP) indicates the number of genuine transactions incorrectly predicted as fraud. Similarly, the other two parameters, false negative (FN) and true positive (TP) are

Table 1 Confusion matrix

Predicted	Actual	
	Normal (0)	Fraud (1)
Normal (0)	True negative (TN)	False negative (FN)
Fraud (1)	False positive(FP)	True positive (TP)

defined. Accuracy measures the number of transactions correctly predicted by the model. But, in case of imbalanced dataset, accuracy metrics do not provide necessary information to check the performance of a classifier. It is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (2)$$

For imbalanced class distributions, the following measures are needed to be considered:

$$F - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

where

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \text{Sensitive} = \text{True Positive rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$G - \text{Mean} = \sqrt{\text{Sensitivity} * \text{Specificity}} \quad (6)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (7)$$

In order to overcome such issues, receiver operating characteristic (ROC) is another approach to evaluate the performance of classifier by using the evaluation metrics, like, TPR and false positive rate (FPR). AUC represents the area under ROC curve and it is most commonly used evaluation measure. MCC measures the performance of the model considering each field of the confusion matrix. MCC metric is much suitable for imbalanced dataset. It is defined as:

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (8)$$

A perfect model results in +1 MCC value, whereas -1 for worse model.

Logistic Loss (Log_loss) is used to measure the effectiveness of the model. Suppose for the transaction i , y_i and p_i represents the actual and predicted value, respectively, and N represents the total number of transactions. It is defined as:

$$\text{Log_loss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log p_i + 1 - y_i \log(1 - p_i)) \quad (9)$$

5 Results and Discussions

The performance evaluation results of all the base classifiers, i.e., DT, KNN, SVM, NB and ANN, along with the ensemble models, like AdaBoost, RF, XGBoost and GBDT are shown in Table 2. The classifiers are ranked on the basis of average rank as proposed in [8], rank 1 indicates best performer and the classifier with higher rank value indicate worst performer. Avg_Rank computes the rank of each classifier by considering the results of each evaluation measures. It shows that XGBoost is the best classifier, whereas DT and NB are the worst performers. Similarly, Fig. 2 shows ROC curve of each classifier. SVM performs best whose ROC curve value is 0.9744 followed by GBDT whose ROC curve value is 0.9710.

The performance of the ensemble model depends upon the base classifiers. Based on the performance analysis of different models, ensemble models are designed.

Table 2 Results of base classifiers

Model	ACC	AUC	<i>F</i> _measure	G-mean	MCC	Log_loss	Avg_Rank
RF	0.922	0.92324	0.92409	0.92302	0.8455	2.8005	3.67
NB	0.899	0.90258	0.89436	0.89874	0.811	3.8506	9.33
KNN	0.926	0.92807	0.92517	0.92668	0.8568	2.9171	2.33
DT	0.892	0.89229	0.89542	0.89225	0.7839	3.5006	9.67
GBDT	0.916	0.91583	0.91856	0.91581	0.831	2.6838	5.17
ANN	0.912	0.91196	0.91613	0.91195	0.8239	2.6838	7
XGBoost	0.926	0.92679	0.92715	0.92649	0.8526	2.5671	1.83
AdaBoost	0.919	0.91937	0.921576	0.91932	0.838	2.5671	3.67
SVM	0.919	0.91654	0.90977	0.91383	0.8447	2.8004	5.5

The best performer is shown in bold fonts whereas the worst performer in italics

Fig. 2 ROC curve of each classifiers

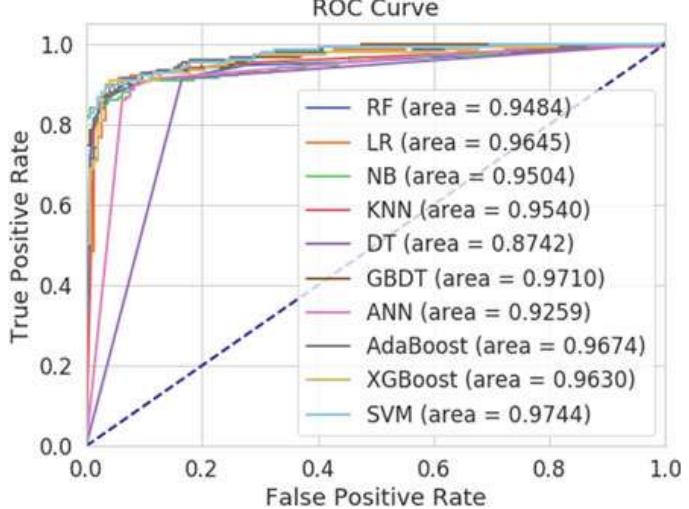


Table 3 Results of ensemble models using majority voting

Model	ACC	AUC	F_measure	G-mean	MCC	Log_loss	Avg_Rank
RF and XGB	0.9155	0.9168	0.9117	0.9147	0.8384	2.9171	3.5
RFand SVM	0.9088	0.9103	0.9032	0.9072	0.8281	3.1505	<i>6</i>
GBDT and AdaBoost	0.9189	0.92	0.9161	0.9185	0.843	2.8005	1.83
KNN and XGBoost	0.9122	0.9134	0.9085	0.9114	0.8308	3.0338	<i>5</i>
RF and GBDT	0.9223	0.92	0.9139	0.9176	0.8509	2.6838	1.5
SVM and BDT	0.9189	0.9165	0.9098	0.9138	0.8447	2.8004	3.17

The best performer is shown in bold fonts whereas the worst performer in italics

Ensemble models are implemented using the best performer models. It includes, XGBoost, RF, KNN, AdaBoost, SVM and GBDT. Two best base classifiers are selected from the above analysis report. Bagging is then implemented taking different combination of the selected base classifiers. We compute the results of the ensemble models using majority voting approach. A total of six models are implemented using bagging technique. The result (Table 3) shows that RF and GBDT ensemble model is the best approach as compared to other ensembles.

The overall performances achieved by the ensemble models using majority voting approach are much better than individual classifiers. The best rank (Avg_Rank as shown in Table 3) of the ensemble model is 1.5 and the worst rank is 6, whereas in case of single classifiers, the best rank (shown in Table 2) is 1.83 and the worst is 9.67.

6 Conclusion and Future Work

In this paper, we implemented machine learning techniques on the credit card dataset. In the proposed model, five different individual classifiers, such as DT, NB, SVM, ANN and KNN are used in the experimental study. In this work, we implement the model using both individual classifiers and ensemble of classifiers. In the experimental study, six different performance evaluators like, ACC, AUC, F_measure, G-Mean, MCC and Log_Loss were used to measure the performance of the model. We conclude that RF and GBDT are the best classifiers for the ensemble models on the publicly available credit card dataset. In the future, we will work on different approaches to tackle imbalanced dataset and we will work to optimize the number, combination and the parameters of the individual classifiers for the ensemble models.

Finally, to improve the performance of the model further, we will incorporate the deep learning techniques to the ensemble models.

References

1. Quah, J.T., Sriganesh, M.: Real-time credit card fraud detection using computational intelligence. *Expert Syst. Appl.* **35**(4), 1721–1732 (2008)
2. Lenka, S.R., Ratha, B.K., Nayak, B.: A review on novel approach to handle imbalanced credit card transactions. *Int. J. Eng. Trends Technol. (IJETT)* **62**(2), 80–95 (2018)
3. Tsai, Chih-Fong, Chen, Ming-Lun: Credit rating by hybrid machine learning techniques. *Appl. Soft Comput.* **10**(2), 374–380 (2010)
4. Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M.S., Zeineddine, H.: An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access* **7**, 93010–93022 (2019)
5. Nascimento, D.S., Coelho, A.L., Canuto, A.M.: Integrating complementary techniques for promoting diversity in classifier ensembles: a systematic study. *Neurocomputing* **138**, 347–357 (2014)
6. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining. ACM (2016)
7. Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A.: Credit card fraud detection using machine learning techniques: a comparative analysis. In: 2017 International Conference on Computing Networking and Informatics (ICCNI). IEEE (2017)
8. Lessmann, S., et al.: Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. *Eur. J. Oper. Res.* **247**(1), 124–136 (2015)

Temporal Modeling of On-Street Parking Data for Detection of Parking Violation in Smart Cities



Shiv Kumar Sahoo, Niranjan Panigrahi, Debasis Mohapatra, Asutosh Panda, and Arvind Sinha

Abstract With the increase in number of vehicles, the requirement of intelligent parking management is indispensable in smart cities. One of the major requirements in smart parking system is handling parking violations efficiently. The parking violation generally includes parking beyond allowed time. To detect parking violations and to manage it efficiently, the parking data collected through field sensor devices need to be analyzed intensively and thoroughly. To this end, this paper has presented temporal analysis of on-street parking data of Melbourne city and proposed a novel mathematical model and curve-fitting algorithm using quasi-Newton method to detect parking violation. The proposed model is validated with real dataset through simulation with a sum of squared error of 4.888×10^{-7} .

Keywords Parking violation · Smart cities · Violation model · Curve-fitting · Quasi-Newton method

1 Introduction

Evolution of Internet of things (IoT) and cloud paradigm has enabled intelligent solutions to many public and governance issues, thus making smart cities a reality. Smart parking is an indispensable requirement of smart cities to manage resources,

S. K. Sahoo · N. Panigrahi (✉) · D. Mohapatra · A. Panda
Parala Maharaja Engineering College, Berhampur, Odisha, India
e-mail: niranjan.cse@pmec.ac.in

S. K. Sahoo
e-mail: maths10sks@gmail.com

D. Mohapatra
e-mail: devpmec@gmail.com

A. Panda
e-mail: asutosh765@gmail.com

A. Sinha
National Institute of Technology, Raipur, Chhattisgarh, India
e-mail: aksinha.maths@nitrr.ac.in

i.e., fuel, limited parking space and time effectively. Recently, smart parking system has attracted both industry and research community because of its inherent issues and open challenges [1]. A smart parking system broadly involves on-street and off-street solutions for parking reservation, navigation, parking lot searching, and allocation [2]. The major objectives considered while designing such systems are: (i) minimizing delay in searching for parking lot, cost of parking and (ii) balancing parking lots. In other words, most of the research works have focused on providing solutions from drivers perspective.

The other aspect of smart parking solution, i.e., governance perspective is a least explored area. One of the important aspects from administration point of view is handling on-street parking violation which involves parking beyond allowed time and parking in the restricted zone. Some recent works in this context are proposed to manage on-street parking violations using cloud and IoT technology [4]. But, designing an efficient parking violation management system requires a priori analysis of historical data for accurate modeling and prediction. A number of works have reported in the literature which are based on analyzing parking data [3]. To the best of our knowledge, few works only focused on analyzing parking data for violation detection [7].

The major contribution of this paper are:

- (i) An extensive temporal analysis is carried out for parking violation on a real dataset from Melbourne city
- (ii) A novel mathematical model is proposed considering the behavior of average parking violation frequency
- (iii) An iterative curve-fitting algorithm (QNCF) is proposed using quasi-Newton method to detect average parking violation frequency.

The rest of the paper is organized as follows. Section 2 gives a summary of literature survey in smart parking. Section 3 highlights the parking data collection phase and a preliminary empirical analysis of parking dataset. Section 4 presents the problem formulation, followed by proposed approach in Sect. 5. Section 6 presents the simulation results and analysis. Finally, Sect. 7 concludes with future work.

2 Related Work

In recent years, smart parking management has been studied in interdisciplinary and diversified way due to its heterogeneous requirements and challenges [1, 2]. Nicola et al. have analyzed the real-time parking data to study parking lot occupancy of Northern Italy with objectives to detect outlier parking sensor data and to cluster data with similar parking behavior. The algorithms used for this purpose are: k-means clustering, DBSCAN, and SOM [3].

Yanxu et al. have investigated farthest first and EM clustering schemes to subdivide parking traces into clusters based on their statistical features and then use support vector data description (SVDD) to identify extreme behaviors (two classes), i.e., either abnormally high or abnormally low occupancy locations [6].

Shi et al. have proposed a crowd-sensed parking system, namely ParkCrowd, to aggregate on-street and roadside parking space information reliably, and to disseminate this information to drivers in a timely manner [5].

Thanh et al. have performed spatial and temporal analysis of on-street parking data of Melbourne city to design a location-centric parking violation management system [4]. Rajabioun et al. [12] have proposed a multivariate autoregressive model to predict temporal and spatial correlations of parking availability, using real-time parking data from San Francisco and Los Angeles. Considering vehicles arrival as Poisson's distribution, Klappenecker et al. [13] have proposed a model for parking lot using a continuous-time Markov chain. With the predicted occupancy status, each parking lot can provide cruising vehicles with the availability information via vehicular networks. Vlahogianni et al. [14] studied on-street parking data from SmartSantander and show that the occupancy and parking periods of four parking areas follow a Weibull distribution.

Wu et al. [15] proposed a cost model to influence users' parking choice. The cost model is based on a probability of parking successfully, within a certain distance from the current location. Ji et al. [16] have modeled parking availability data of some off-street parking garages in Newcastle, using wavelet neural network method to investigate the changing characteristics of short-term available parking spaces. It is compared with the largest Lyapunov exponents method using metrics like accuracy, efficiency, and robustness.

3 Data Collection and Preliminary Analysis

Collection of data is a challenging task and requires a technological infrastructure like IoT deployment [10, 11]. Recently, the Victoria government has released a massive on-street parking dataset of 34 cities in Melbourne from year 2011–2017. It consists of around 10 millions records with attribute like sensor id, arrival time, departure time, street name ,area name , in _violation, etc. [8]. Many recent parking solutions and analysis are carried out on this dataset [4, 7]. But, a thorough analysis of parking violation is still lagging on this dataset. So, this dataset is considered as reference for analysis and performance comparison.

The referred dataset consists of a binary attribute, in _violation, which stores 1 *if* there is violation else store 0 *if* there is no violation. From this, the following temporal analysis is carried out to understand the pattern of violation distribution in all 34 cities of Melbourne. Figure 1 shows the average violation frequency at different hours in a day. Though the violation frequency values are different at different hours in a day, they follow a similar pattern of distribution. Hence, the average of all 34 cities at different hour intervals is computed to infer a unified pattern of violation distribution as shown in Fig. 2.

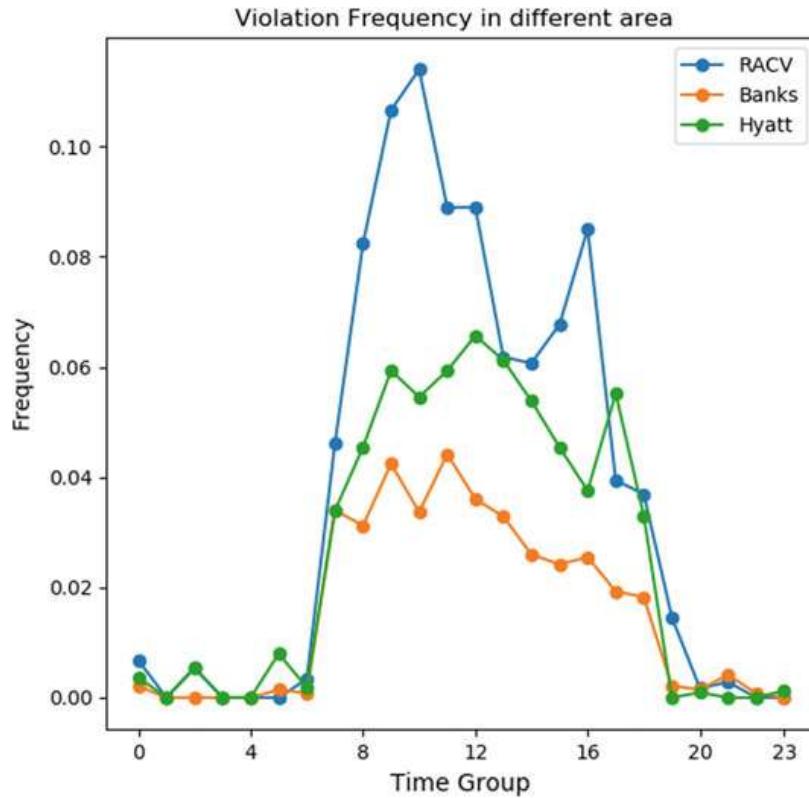


Fig. 1 Hour versus average frequency of violation

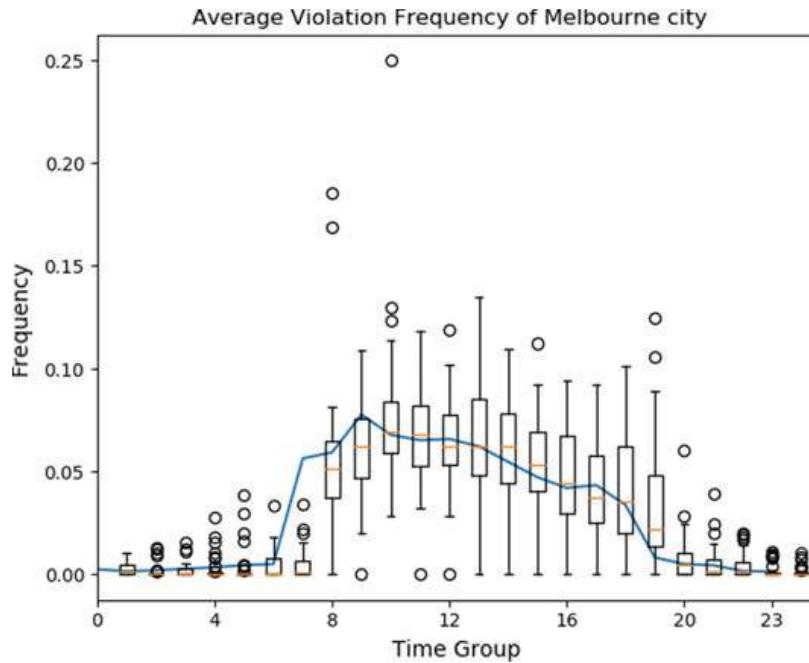


Fig. 2 Hours versus average frequency of violation for all 34 cities

4 Problem Formulation

From parking violation frequency distribution as observed in Fig. 2, it is required to design a model which can approximate the data and able to predict parking violation frequency at any time instance in different area with minimum prediction error. Hence, the problem can be formulated as a nonlinear least square unconstrained optimization problem which can be expressed as follows.

Let, the collected data which are plotted in Fig. 2 is denoted as $\{(t_i, y_i)\}_{i=1}^m$ where y_i is the average violation frequency at time instance t_i of i th observation. It is required to design the parking violation model $f(c_1, c_2, \dots, c_k)$ such that

$$\min \sum_{i=1}^m (f_i(\cdot) - y_i)^2 \quad (1)$$

The behavior of the model function $f(\cdot)$ is greatly dependent on $f(c_1, c_2, \dots, c_k)$ and it is desired to estimate the constants which minimize the error for a selected function. This problem can be best approximated as curve-fitting problem. The following section presents the proposed approach using a curve-fitting method.

5 Proposed Approach

From the study of well-known methods of curve-fitting [9], it is observed that quasi-Newton method has advantages over other methods. Motivated by this, the following section presents proposed mathematical model for parking violation detection and its parameter estimation using quasi-Newton method to define the exact model for the given problem with minimum error.

5.1 Proposed Detection Model

From Fig. 2, it is observed that the average frequency of parking violation shows a rise in first-half interval of the day and falls in the second-half of the day. Hence, the following mathematical function in Eq. 2 is formulated looking into the behavior of average frequency violation.

$$f = c_1 \sin(c_2 x + c_3) + c_4 \quad (2)$$

where c_1, c_2 , and c_3 are constants.

The above-defined constants determine the shape of the curve which are calculated using the following proposed iterative method of random constants generation and quasi-Newton method of curve-fitting.

5.2 Proposed Algorithm

This section presents the proposed iterative method of curve-fitting using quasi-Newton method and random seed generation for optimal parameters selection. The details of the algorithm are as follows.

Algorithm 1 Iterative_random_seeding_QNCF(t_i, y_i)

```

1: Set  $r_{min}$  and  $r_{max}$ 
2: for  $i=1$  to  $I_{max}$  do
3:   /****  $I_{max}$ =maximum number of iteration ****/
4:   Compute random seed  $s_0$  between  $r_{min}$  and  $r_{max}$ 
5:   ( $c_i, er_i$ )=QUASI_NEWTON(ERRORF, $s_0$ )
6: end for
7:  $c_i^{opt}=c_i:\min(er_i)$ 
8: p=DATA_MODEL( $t_i, c_i^{opt}$ )
9: function ERRORF( $c_i$ )
10:    $er_{val}=y_i-\text{DATA\_MODEL}(c_i)$  return  $er_{val}$ 
11: end function
12: function DATA_MODEL( $t_i, c_i$ )
13:   val= $c_1 \sin(c_2 x + c_3) + c_4$  return val
14: end function

```

The algorithm presented above starts with the initialization of the range $[r_{min}, r_{max}]$ for random seed generation s_0 . Two functions are defined, namely DATA_MODEL in line 9 and ERRORF in line 12 where DATA_MODEL estimates the proposed candidate function (parking violation model) for different values of parameters and the function ERRORF estimates the error between the empirical average frequency of violation data stored in y_i and the estimated value from proposed model. This function is passed as an argument to the QUASI-NEWTON function in line 5 to store the parameters c_i and error err_i for different iterations. Then, the optimal parameters c_i^{opt} are selected for minimized value of error in line 7.

6 Simulation Results and Analysis

To evaluate the proposed algorithm, it is tested in SCILAB 6.0.1 on Windows 8.1 pro with the system configuration as: Intel(R) Core(TM) i5-4590 CPU, 3.30 GHz, and 4 GB RAM. The range for random seed generation is set as $[r_{min}, r_{max}]=[-5, 5]$. Maximum iteration is set as $I_{max} = 15$, and increasing the iterations result in no improvement in the parameters. So, this will act as the stopping criteria for the proposed algorithm. To simulate, quasi-Newton algorithm, the library function *datafit* is used with function argument *algo* as *qn*.

To evaluate the error of prediction, a standard evaluation metric, sum of squared error (SSE), is taken into consideration. After reaching the maximum iterations, i.e.,

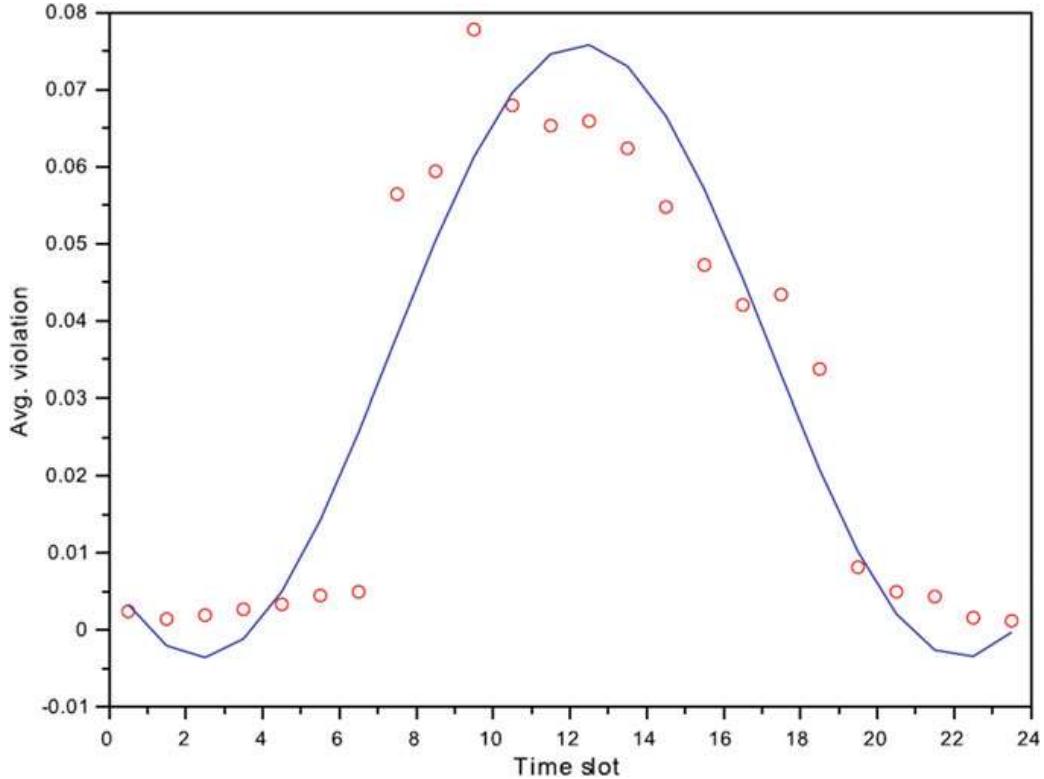


Fig. 3 Curve-fitting with proposed QNCF algorithm

convergence of the algorithm, the estimated value of parameters are: $c_1 = -0.890$, $c_2 = 0.555$, $c_3 = 1.497$, and $c_4 = 0.831$. Hence, the predicted model shown in solid line in Fig. 3 along with the empirical data in dots for the parking violation can be formulated as:

$$f = -0.890 \sin(0.555x + 1.497) - 0.831 \quad (3)$$

The estimated SSE is 4.888×10^{-7} .

7 Conclusion and Future Work

On-street parking violation detection and prediction is a major concern for city administration in smart cities to effectively manage it. For this purpose, a temporal prediction model is proposed using quasi-Newton curve-fitting method, based on real dataset of Melbourne city. With the comparative analysis of empirical data, the proposed model predicts the average frequency of violation with a sum of squared error of 4.888×10^{-7} . Our future work will include comparative analysis with other statistical methods.

Acknowledgements This research work is supported under NPIU,TEQIP-III sponsored Collaborative Research Scheme(CRS).

References

1. Turjman, A., Malekloo, A.: Smart parking in IoT-enabled cities: a survey. In: Sustainable Cities and Society, vol. 49, pp. 2210–6707. Elsevier (2019). <https://doi.org/10.1016/j.scs.2019.101608>
2. Lin, T., Rivano, H., Le Moul, F.: A survey of smart parking solutions. IEEE Trans. Intell. Transp. Syst. **18**(12), 3229–3253 (2017). <https://doi.org/10.1109/TITS.2017.2685143>
3. Piovesan, N., Turi, L., Toigo, E., Martinez, B., Rossi, M.: Data analytics for smart parking applications. Sensors (Basel, Switzerland) **16**(10), 1575 (2016). <https://doi.org/10.3390/s16101575>
4. Dinh, T., Kim, Y.: A novel location-centric IoT-cloud based on-street car parking violation management system in smart cities. Sensors **16**(6), 810 (2016). <https://doi.org/10.3390/s16060810>
5. Shi, F., Wu, D., Arkhipov, D.I., Liu, Q., Regan, A.C., McCann, J.A.: ParkCrowd: reliable crowdsensing for aggregation and dissemination of parking space information. IEEE Trans. Intell. Transp. Syst. (2018). <https://doi.org/10.1109/TITS.2018.2879036>
6. Yanxu, Z., Rajasegarar, S., Leckie, C., Palaniswami, M.: Smart car parking: temporal clustering and anomaly detection in urban car parking. In: Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, pp. 21–24, April 2014
7. Shao, W., Salim, F.D., Song, A., Bouguettaya, A.: Clustering big spatiotemporal-interval data. IEEE Trans. Big Data **2**(3), 190–203 (2016). <https://doi.org/10.1109/TBDA.2016.2599923>
8. On-street parking Bay sensor data. <https://data.melbourne.vic.gov.au>
9. Gopalkrishnan, S., Bourbakis, N.: Curve fitting methods. Int. J. Monit. Surveill. Technol. Res. **4**, 33–53 (2016)
10. Satapathy, S.C., Bhateja, V., Raju, K.S., Janakiramaiah, B.: Data engineering and intelligent computing. In: Proceedings of IC3T, 542 (2016)
11. Satapathy, S.C., Tavares, J.M.R.S., Bhateja, V., Mohanty, J.R.: Information and decision sciences. In: Proceedings of the 6th International Conference on FICTA (2017)
12. Rajabioun, T., Ioannou, P.A.: On-street and off-street parking availability prediction using multivariate spatiotemporal models. IEEE Trans. Intell. Transp. Syst. **16**(5), 2913–2924 (2015)
13. Klappenecker, A., Lee, H., Welch, J.L.: Finding available parking spaces made easy. Ad Hoc Netw. **12**, 243–249 (2014)
14. Vlahogianni, E., Kepartsoglou, K., Tsetsos, V., Karlaftis, M.: A realtime parking prediction system for smart cities. Intell. Transp. Syst. **20**(2), 192–204 (2016)
15. Wu, E., Sahoo, J., Liu, C., Jin, M., Lin, S.: Agile urban parking recommendation service for intelligent vehicular guiding system. IEEE Intell. Transp. Syst. Mag. **6**(1), 35–49 (2014)
16. Ji, Y., Tang, D., Blythe, P., Guo, W., Wang, W.: Short-term forecasting of available parking space using wavelet neural network model. IET Intell. Transp. Syst. **9**(2), 202–209 (2015)

Performance Optimization of Big Data Applications Using Parameter Tuning of Data Platform Features Through Feature Selection Techniques



Tanuja Pattanshetti and Vahida Attar

Abstract Big data application performance can be optimized by identifying the most impactful set of system parameters of big data platforms. This paper focuses on the identification of optimal system parameter set of Hadoop and Spark data platforms by applying different feature selection techniques. The main objective of the research work is to reduce the job execution time by identifying and tuning only these identified system parameters. The parameters deemed to be less relevant and redundant get eliminated during the feature selection process. The parameters identified using different feature selection algorithms are compared, and empirical analysis is carried. The statistical analysis is used as a cross-validation technique to evaluate the relevance of the identified parameter set and the dependency of platform performance on system parameters.

Keywords Big data platforms · Feature selection algorithms · Statistical analysis · Hadoop · Spark · Parameter tuning

1 Introduction

The data which have huge volume ranging from terabytes to exabytes, which exists in different formats, gathered and contributed through heterogeneous systems with huge inundation and veracities is coined as big data. The above features have helped in adding one more dimension, the “value” to big data making it important from commercial perspective. Thinking about the different highlights of big data, it is not feasible for any customary hardware/software to fulfill the client’s needs. In order to sustain with these real-world requirements and to cater to the customer’s needs, big data platform(s) play a significant role in providing the cluster topology

T. Pattanshetti (✉) · V. Attar
College of Engineering Pune, Pune, Maharashtra, India
e-mail: trpattanshetti@gmail.com

V. Attar
e-mail: vahida.comp@coep.ac.in

or horizontal scaling. Scaling is the capacity of the framework to adjust to expanded requests as an aspect of cloud service. The basic qualities provided by the cloud are harnessed by making use of big data processing platforms. They are many in types and offer platform as a service (PaaS) in cloud computing. Prominent platforms which support horizontal scaling include shared systems, Apache Hadoop, Spark, Storm, and so forth, while HPC, multicore processors, GPUs, and FPGA bolster vertical scaling [1]. Apache Hadoop is an open-source framework for batch processing and handling enormous data storage. The Hadoop platform makes use of different stack components like HDFS, Hadoop YARN, MapReduce, Apache pig, etc. Apache Spark is created with principle target to conquer disk I/O constraints in Hadoop through built-in memory calculations.

The platforms used for processing big data have several system parameters, few ranging from 100 to 200 and are used for customizing and controlling the memory, network and operating system, infrastructure-related operations. In case of any issue related to application processing or its failure, identifying the underlying cause by inspecting every individual parameter is a tedious task and time consuming. The proposed system helps in selecting the relevant system parameters through feature selection techniques and its approaches. Tuning the relevant parameters to the best possible values will reduce the execution time required for job processing.

This paper also applies statistical analysis and normality tests to support the research findings. Hypothesis testing is part of the statistical analysis, and it is generally done to validate the assumptions made by the distribution. Normality tests in this analysis are used to test whether the dataset pursues a normal distribution or not. If the bell-shaped curve is obtained, then dataset generally follows a normal distribution. D'Agostino Skewness, D'Agostino Kurtosis, D'Agostino Omnibus and Anderson Darling are few of the statistical tests which predict the goodness of fit of the bell shaped curve used here to ensure the normal distribution. This paper has considered various statistical factors such as mean squared error, correlation coefficient, and a multiple linear regression to validate the hypothesis—the correct identification of precise system parameter set will enhance the performance of big data platforms.

The research paper is organized further as described below. Section 2 discusses the state of art carried in this research area. Section 3 proposed system is introduced which discusses various tools used for the calculations of a multiple linear regression and tests performed on the datasets. Section 4 discusses experimental results, and in Sect. 5, the paper is concluded with a remark of application of mean-based multiple linear regression as a robust algorithm for feature selection.

2 Related Work

A review of different big data systems and strategies used for sharing and transfer of information is carried, with identification of the cause of impediment and techniques to improve the performance through parameter tuning [2]. The analysis of

large volumes of data is carried with focus on the challenges of scaling and heterogeneity at all phases of the information analysis [3]. Parameter tuning is suggested by the authors to amplify the performance of big data applications [4] as not all parameters contribute toward the performance of applications and have proved that in Hadoop around 25 parameters have high impact. H-Tune method is proposed [5] to tune the MapReduce applications and reduce performance overhead by 2%. Genetic algorithm and software-defined network are used for tuning the Hadoop platform parameters which have helped in significant improvement of the MapReduce job execution over existing methods [6].

MapReduce cloud administration model is proposed [7] which shows benefits over existing models for harnessing the benefits of cloud. Relief feature set measure is proposed based on relief algorithm as a feature estimator [8]. The proposed method makes use of binary and multi-classification technique of machine learning for auto-tuning the configuration parameters of Spark. It has showed better accuracy over the existing methods [9]. As stream processing systems involves vast number of features, linear regression model is used to identify the relationship for optimization of overall system [10]. Various kernel estimators are used to estimate the locales for identification of ideal design using tuning methods [11]. Different statistical methods are discussed to study the non-normality of data and the correction methods to address this issue [12].

Author has in detail discussed the statistical analysis with the help of the statistical strategies including correlation, regression, t-tests, and variance to identify the type of distribution [13]. Author has given a thorough description about different types of graphical and numerical plots to assess the types of distribution with comment that normality test is a more formal procedure to test normality. Stem-and-leaf representation is similar to histogram and gives information about original data without any loss. When the sample is taken from non-normal distribution, the test which has ability of rejecting the null hypothesis of normality seems to be significant [14–16]. Cluster independent parameters of Spark which are common to all clusters are identified using trial and error method and are tuned for performance optimization [17]. Authors have proposed a model for predictive analysis in which dependent variable can be predicted from the independent variables. But as the number of independent variables decrease the prediction effectiveness of the entire model also gets affected [18]. Mean representation classification model proposed by researchers uses the mean vector of each class to constitute the global-model for classification and performs better than existing techniques [19].

After exploring the limitations and challenges from the state of art survey, the research work has proposed the model to reduce the feature space of system parameters of data platforms to increase the accuracy and predictability.

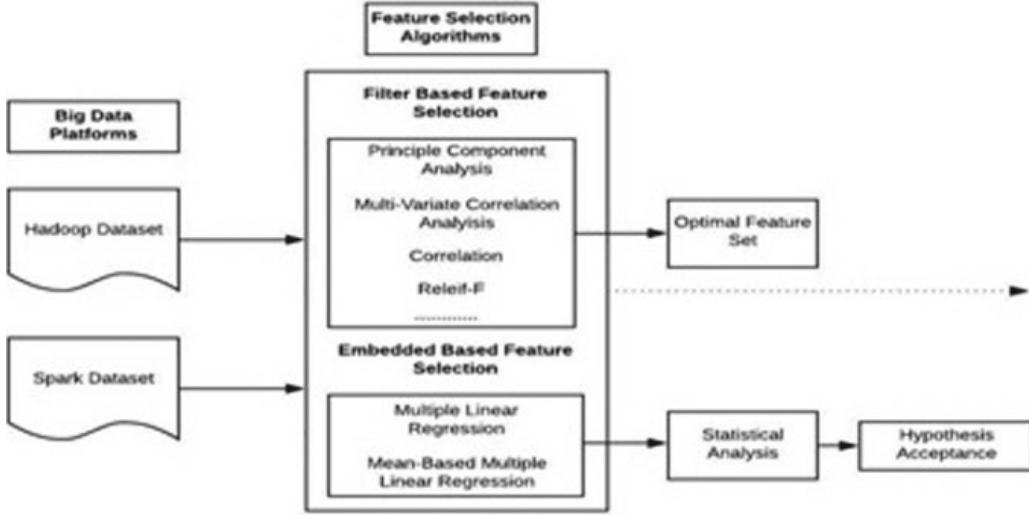


Fig. 1 Architectural view of the proposed system

3 Proposed System

In the proposed work, by applying filters and embedded methods of feature selection technique, small subset of features upon whom the performance depends is identified by eliminating redundant and irrelevant features. Empirically, 152 system parameters of Hadoop and 30 system parameters of Spark are identified by using benchmarked TeraSort application. After preprocessing and standardization process, the features space of Hadoop and Spark is reduced to 44 and 26 parameters, respectively.

Further to identify the precise parameter set, various filter-based feature selection algorithms of filter approach are applied, and thus the feature set is reduced by eliminating the redundant and combining possible parameters into derived ones, to a feature set of 26 and 18 parameters for Hadoop and Spark, respectively.

The big data platform parameters identified using novel mean-based technique for feature selection are also compared with the parameter set identified by legacy-based feature selection methods like PCA, CFS, MLR, ReliefF, correlation, standard deviation, multilinear regression, MCA [20], and MBMLR [21]. The final subset is made up of the parameters by applying intersection on the subset identified by individual algorithm. Thus, the final subset consists of only commonly identified parameters by all algorithms. The architectural view of the proposed system is shown below in Fig. 1.

4 Experiments and Results

In the scatters plots shown below, the x -axis shows the data platform parameter and its corresponding feature weight on y -axis. In Fig. 2a, b, the parameters of Hadoop and Spark platforms with their corresponding feature weights are shown, respectively.

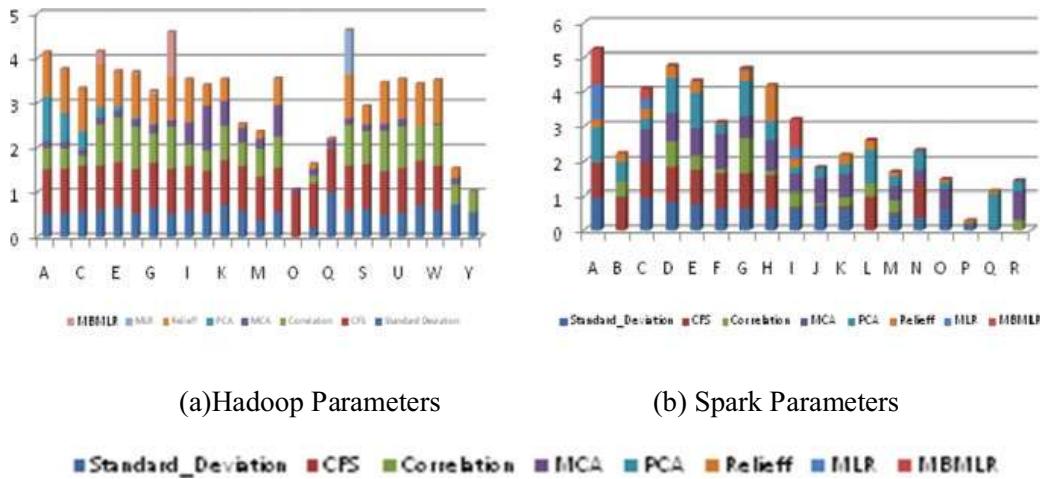


Fig. 2 Parameters and feature weights identified by feature selection algorithms using filter approach

Thus, the final feature set identified for Hadoop and Spark platforms consists of ten most relevant and common parameters identified by all algorithms. From the original feature space of Hadoop (152 features) and Spark (30 features), ten features marked with “*” in Table 1, are identified as the ones having high relevance with the target variable “y” (total execution time).

Embedded approach also helps in identifying the outliers in a precise manner indicating their irrelevance. The mean-based algorithm [10] further improves the precision accuracy as compared to multilinear regression method by identifying the optimal set consisting of only the most relevant parameters. This is obvious from the results of the normality tests as shown in Table 2.

The results show reduction in mean square error to a great extent. After application of MBMLR, it can be observed that coefficient of correlation has increased and mean square error has got reduced significantly indicating its applicability as a feature selection technique. It has helped in precise outlier detection, offering better model accuracy, and predictability.

5 Conclusion

Even if there exists a large number of system parameters of big data platforms, not all system parameters play significant role in defining their performance. Few system parameters are redundant and non-relevant and thus act as outliers. Detection and elimination of these outliers by the proposed method have helped in reducing the error. With the help of statistical analysis, the parameters identified through the proposed method are validated. This ensures that the performance of data platforms depend on the system parameters. The research work has helped in identifying the

Table 1 Hadoop and Spark parameters identified by filter approach

Hadoop parameters				Spark parameters			
<i>A</i>	File bytes read*	<i>I</i>	Physical memory*	<i>Q</i>	Maps completed	<i>A</i>	Number Of tasks*
<i>B</i>	File bytes written*	<i>J</i>	Namenode Capacity remaining	<i>R</i>	Avg. map time	<i>B</i>	Disk size*
<i>C</i>	HDFS bytes read*	<i>K</i>	Namenode blocks total	<i>S</i>	Avg. reduce time*	<i>C</i>	Disk bytes spilled
<i>D</i>	Maps*	<i>L</i>	Yarnam launch delay Avg. time	<i>T</i>	Virtual cores *	<i>D</i>	Shuffle write time*
<i>E</i>	Mb millis reduces	<i>M</i>	Yarn Am register delay avg. time	<i>U</i>	Avg. merge time	<i>E</i>	Local blocks fetched*
<i>F</i>	Spilled records*	<i>N</i>	Yarn Available Mb	<i>V</i>	Successful map attempts	<i>F</i>	Remote bytes read*
<i>G</i>	GC time *	<i>O</i>	Yarn aggregate containers allocated	<i>W</i>	Avg. data node remaining	<i>G</i>	Replication *
<i>H</i>	CPU time *	<i>P</i>	Dfs block size	<i>X</i>	Total time	<i>H</i>	Executor run time*
						<i>P</i>	Total time

* Indicates the parameters commonly identified by all feature selection algorithms and are the ones having high relevance with the target variable 'y' (total execution time).

Table 2 Statistics and hypothesis testing for original dataset and by using MBMLR dataset of Hadoop and regression Spark platforms

	1	2	3	4	5	6	7	8
Hadoop platform	R^2	1.000	1.000	D'Agostino Skewness	0.0006	Yes	0.0848	No
	Adj. R^2	1.000	1.000	D'Agostino Kurtosis	0.0019	Yes	0.0004	Yes
	Mean square error	0	0	D'Agostino Omnibus	0.000	Yes	0.0004	Yes
Spark platform	R^2	0.9111	0.9936	D'Agostino Omnibus	0.8648	No	0.5497	No
	Adjacent R^2	0.8859	0.9917	D'Agostino Kurtosis	0.8302	No	0.0024	Yes
	Mean square error	8.440359E+12	4.315293E−11	D'Agostino Omnibus	0.9632	No	0.0085	Yes

1. Item. **2.** Value for original dataset. **3.** Value for mean dataset using MBMLR. **4.** Test name. **5.** Probability level. **6.** Reject hypothesis for original dataset. **7.** Probability level. **8.** Reject hypothesis for mean dataset using MBMLR

precise parameter set of Hadoop and Spark data platforms. Tuning these parameters and the ones on which these parameters depend may definitely help in improving their performance.

References

1. Singh, D., Reddy, C.K.: A survey on platforms for big data analytics. *J. Big Data* **2**(1), 8 (2015)
2. Kamtekar, K., Jain R.: Performance Modeling of BigData—The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley Interscience, New York. ISBN: 0471503363 (1991)
3. Jagadish, H.V., Labrinidis, A.: Challenges and opportunities with big data. *ACM* **5**(12), 2022–2023 (2012)
4. Chen, X., Liang, Y., Li, G.R., Chen, C., Liu, S.Y.: Optimizing performance of Hadoop with parameter tuning. *ITM Web of Conferences* **12**, 30–40 (2017)
5. Hua, X., Huang, M.C., Liu, P.: Hadoop configuration tuning with ensemble modeling and metaheuristic optimization. *IEEE Access* **6**, 44161–44174 (2018)
6. Khaleel, A., Al-Raweshidy, H.: Optimization of computing and networking resources of a Hadoop cluster based on software defined network. *IEEE Access* **6**, 61351–61365 (2018)
7. Palanisamy, B., Singh, A., Liu, L.: Cost-effective resource provisioning for mapreduce in a cloud. *IEEE Trans. Parallel Distrib. Syst.* **26**(5), 1265–1279 (2015)
8. Arauzo-Azofra, A., Benitez, J.M., Castro, J.L.: A feature set measure based on relief. In: Proceedings of the Fifth International Conference on Recent Advances in Soft Computing, pp. 104–109 (2004)

9. Wang, G., Xu, J., He, B.: A novel method for tuning configuration parameters of spark based on machine learning. In: IEEE, 18th International Conference on High Performance Computing and Communications, pp. 586–593 (2016)
10. Prasad, B.R., Agarwal, S.: Performance analysis and optimization of spark streaming applications through effective control parameters tuning. In: Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 99–110. Springer, Singapore (2018)
11. Jamshidi, P., Casale, G.: An uncertainty-aware approach to optimal configuration of stream processing systems. In: IEEE, 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 39–48 (2016)
12. Aldor-Noiman, S., Brown, L.D., Buja, A., Rolke, W., Stine, R.A.: The power to see: a new graphical test of normality. *Am. Stat.* **67**(4), 249–260 (2013)
13. Ghasemi, A., Zahediasl, S.: Normality tests for statistical analysis: a guide for non-statisticians. *Int. J. Endocrinol. Metab.* **10**(2), 486 (2012)
14. Razali, N.M., Wah, Y.B.: Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *J. Stat. Model. Anal.* **2**(1), 21–33 (2011)
15. Das, K.R., Imon, A.H.M.R.: A brief review of tests for normality. *Am. J. Theor. Appl. Stat.* **5**(1), 5–12 (2016)
16. Yap, B.W., Sim, C.H.: Comparisons of various types of normality tests. *J. Stat. Comput. Simul.* **81**(12), 2141–2155 (2011)
17. Petridis, P., Gounaris, A., Torres, J.: Spark parameter tuning via trial-and-error. In: INNS Conference on Big Data, pp. 226–237. Springer, Berlin (2016)
18. Park, N.J., George, K.M., Park, N.: A multiple regression model for trend change prediction. In: International Conference on Financial Theory and Engineering, pp. 22–26. IEEE (2010)
19. Feng, Q., Zhu, Q., Yuan, C., Lee, I.: Multi-linear regression coefficient classifier for recognition. In: IEEE Congress on Evolutionary Computation, pp. 1382–1387 (2016)
20. Pattanshetti, T., Attar, V.: Unsupervised feature selection using correlation score. In: Computing, Communication and Signal Processing, pp. 355–362. Springer, Singapore (2019)
21. Pattanshetti, T., Attar, V.: Mean Based Robust Multilinear Regression for Feature Selection (2019 Accepted)

Development of Emotional Decision-Making Model Using EEG Signals



Mitul Kumar Ahirwal  and Mangesh Ramaji Kose

Abstract In this paper, an attempt has been made to explore the effect of incidental emotion on decision making. For this, first of all, a conventional emotion classification system based on electroencephalogram (EEG) signal is implemented. This emotion classification system ensures the generation and presence of four basic emotions, happy, relaxing, sad and angry by stimulations. audio-visual stimulations are used to generate specific emotions. The novelty of this work is in analysis of pre- and post-decisions taken with respect to stimulation provided. For this, answers of same question were taken before and after the induced emotions by stimulation. After observation and analysis, maximum (46.67%) percentage of change in decision has been noticed during angry emotion.

Keywords EEG · Decision making · Classification

1 Introduction

At present, the concept of emotion recognition through EEG signal is very well known to biomedical signal processing community. Methods involved in the development of emotion recognition system are divided into two parts, first part is stimulation selection and recording of physiological data, second part is feature extraction techniques and classification techniques [1]. In EEG signal processing, various feature extraction and classification methods are used, selection of these methods depends upon the requirement and nature of study. Some recent and commonly well-known

M. K. Ahirwal (✉)

Computer Science and Engineering Department, Maulana Azad National Institute of Technology, Bhopal 462003, India

e-mail: ahirwalmitul@gmail.com

M. R. Kose

Computer Applications Department, National Institute of Technology, Raipur, Chhattisgarh 492010, India

e-mail: mangeshkose@gmail.com

techniques are reviewed in [2, 3]. Nowadays, EEG correlation or coherence network and graph-based connectivity methods are used for brain and EEG signal analysis and visualization [4–6].

In parallel to emotion recognition, analysis of effect of emotions on decision making is also a very new area of cognitive science. It is also normally known to all that emotions plays an important role in over human behavior, response to particular situation and also effects memory some times. Various models of human decision making have been proposed by considering different parameters [7, 8].

Emotions help to make optimal decision like social decisions and perceptions of personality of person or groups/community. It has been seen that while making decision in any situation/problem, two different emotional influences may lead to different choice for same decision-making problem [9]. Various theoretical models of decision making are available. But, till now as such no model with proper experiment exists. Therefore, in this study, an attempt has been made to model the phenomena of change in decision due to change in emotions. In this first half of the study, EEG-based emotion recognition system is implemented to ensure the emotional status of subject during the experiment. In second half of the study, the analysis of decision influence by various emotions has been done.

2 Methodology

Method followed in this paper is divided into two parts, one is observation of change in answers of same questions asked before and after the emotional stimulations. Second is emotion identification done with the help of EEG signals recorded at the time of audio-visual stimulation. Below design of experiment is explained with each stage in details. Description of dataset is also provided.

2.1 Design of Experiment

- The motive of this study is to notice the changes in decisions (answers) after change in emotions by audio-visual stimulations.
- Questions are designed in such a manner that their answers (options) are some where linked or observed in audio-visual clips. Subjects are unaware of the linking of stimulations and questions.
- Total 16 audio-visual stimulations are selected to target happy, angry, sad and relaxing emotions. For each type of emotion, 3 related questions have been designed, total 12 questions are used.

Stage 1: Questions have been provided to each subject and their answers have been taken in the form of options. Each question has four options. Subject has to tick one of the four options, according to his memory, feeling and emotions at that

time. Subjects are undergraduate and postgraduate students, each student is assigned a serial number. This process was done at day time between 11:00 AM and 04:00 PM in normal working day.

Stage 2: After some days their EEG recording was taken for emotions classification experiment. Audio-visual clips have been shown to subject as stimulation to change their emotional state. At the end of stimulations, same questions have been provided to answer.

Stage 3: Analysis of change in answers given by subject at first time and second time after emotional stimulation.

In this study, ten subjects were evaluated. They had no prior experience with BCI or EEG recording. The subjects were healthy students.

Dataset: EEG signal of total ten subjects has been recorded, including five male and five female. Signals are recorded at sampling frequency of 500 Hz and 14 channels are used. Channels names are O1, O2, P3, P4, T7, T8, Fc5, Fc6, Fc1, Fc2, Fz, Cz, Fp1 and Fp2. Stimulation is provided to subjects in the form of video clips.

Feature Extraction: A very recent graph-based network connectivity approach is used for feature extraction. In this method, EEG electrodes are considered as nodes of graph. Connectivity of each node with other nodes is considered on the basis of correlation (equal or more than 80%) between each EEG signal of each node. After creation of connectivity network, network properties are considered as features. In this study, characteristic path length, total number of triangles, assortativity coefficient, modularity, link density, degree distribution and network diameter are used as network features (total seven features).

Classification: For classification, simple model of artificial neural network is used. The above data set is of dimension 160×7 (samples x features) with four classes for classification. During training of classifier, 86.17% accuracy is achieved, while testing phase, 80.00, 72.50, 70.00 and 82.50% accuracy is achieved in respective classes of happy, relaxing, sad and angry emotions. This classification accuracy can be further improved by implementing more advanced machine learning techniques.

Analysis of Change in Answer/Decision: Answer to question has been taken twice, first time before the video stimulation and second time after few days with video stimulation and EEG recording. Change in answer after video stimulation is observed and emotion wise average of changes is provided in Table 1. Anger emotion and related question get maximum changes.

Table 1 Percentage of change in decision

Emotional video clips	No. of questions	% change in option/choice average of all subjects (%)
Happy	03	36.67
Relax	03	26.67
Sad	03	16.67
Anger	03	46.67

3 Decision-Making Model

Some researches show that incidental emotions generated from one situation affect the decisions though the decision outcome has nothing to do with the source of emotion generating situation. Here, a model for decision making is suggested with traditional/rational choice and emotional input. The suggested model named as incidental emotion decision (IED) model. In this model, decision maker is influenced by incidental emotions. There is no evaluation process to check the correctness of decision. Because, all the questions and their answers are subject to personal choice. The main focus is on change in decision due to generated emotions.

The above model as shown in Fig. 1 is developed on the basics of obtained results from the experiments performed. This model highlights the role and affect of emotions on decision making. Here, decision 1 (shown by solid arrow) is taken to select, e.g., option A in normal conditions. Now, when incidental emotion is generated by the mean of stimulation, which will affect the current emotion and this leads to change in decision. For this time, for same question, decision 2 (shown by dashed arrow) is taken to select, e.g., option B.

The parameters considered in this model are as following:

- *Characteristics of decision maker*: These characteristics are preferences, personality and other factors that changes with subject who will take decision.
- *Characteristics of options/choices*: These characteristics are likelihood or probability, time delay, interpersonal outcomes/interest and nature of options with respect to question.
- *Emotions*: These are general emotions felt at time of decision, normally relaxing is considered. But this may also change with nature and relation of question/options/choices with subject.

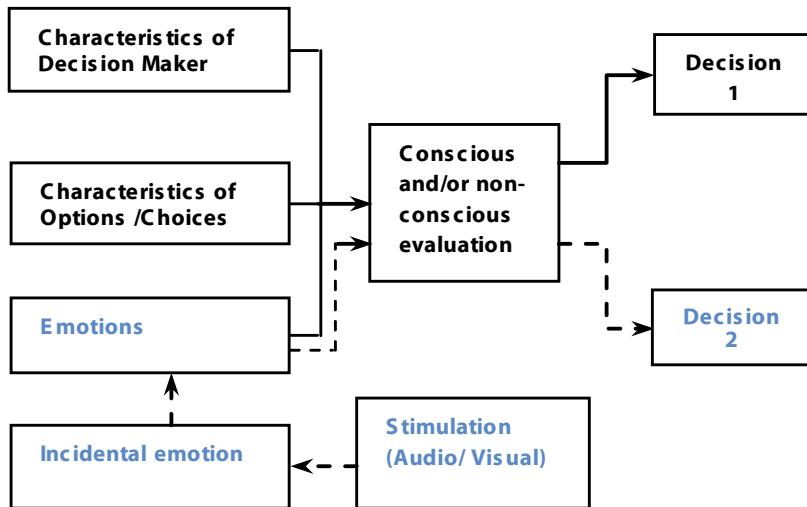


Fig. 1 Proposed model of incidental emotion decision (IED)

- *Incidental emotion:* These are emotions generated by mood, weather or stimulations like audio/visual mean. These incidental emotions may or may not be related with question/options/choices.
- *Conscious and/or non-conscious evaluation:* This is part where characteristics of decision maker, options/choices and emotions are evaluated to take decision.
- *Stimulation (audio/visual):* These are audio/visual stimulation which will intense or change the emotions from previous emotions. These are also cause of incidental emotions in this study.

4 Conclusion

EEG-based classification of emotions ensures the presence of four types of emotions during the experiment. It is concluded that the incidental emotion play role in decision making. In case of anger emotion, there is 46.67% chance of change in decision as compared to normal cases.

Acknowledgements This research and study are done under a project entitled “Development of Computational Model for Decision Making based on Emotion Recognition through EEG signal” in file no. ECR/2017/000250, funded by SCIENCE & ENGINEERING RESEARCH BOARD (SERB) a statutory body of the Department of Science & Technology, government of India.

Declaration We have taken permission from competent authorities to use the images/data as given in the paper. In case of any dispute in the future, we shall be wholly responsible.

References

1. Ahirwal, M.K., Kose, M.R.: Emotion recognition system based on EEG signal: a comparative study of different features and classifiers. In: Proceedings of Second IEEE Conference on Computing Methodologies and Communication (ICCMC), pp. 472–476 (2018)
2. Mishra, A., Bhateja, V., Gupta, A., Mishra, A., Satapathy, S.C.: Feature fusion and classification of EEG/EOG signals. In: Proceedings of Soft Computing and Signal Processing. Advances in Intelligent Systems and Computing (2019). (In press)
3. Majumdar, K.: Human scalp EEG processing: various soft computing approaches. *Appl. Soft Comput.* **11**(8), 4433–4447 (2011)
4. Ahirwal, M.K., Kose, M.R.: Audio-visual stimulation based emotion classification by correlated EEG channels. *Health Technol.* (2019). (In press). <https://doi.org/10.1007/s12553-019-00394-5>
5. Ahirwal, M.K., Kumar, A., Londhe, N.D., Bikrol, H.: Scalp connectivity networks for analysis of EEG signal during emotional stimulation. In: Proceedings of IEEE Conference on Communication and Signal Processing (ICCSP), 0592-0596 (2016)
6. Chengtao, J., Natasha, M., Maurits, Roerdink, J.B.T.M.: Data-driven visualization of multi-channel EEG coherence networks based on community structure analysis. *Appl. Netw. Sci.* **3**(41), 1–24 (2018)
7. Van Kleef, G.A., De Dreu, C.K., Manstead, A.S.: An interpersonal approach to emotion in social decision making: the emotions as social information model. *Adv. Exp. Soc. Psychol.* **42**, 45–96 (2010)

8. Andrade, E.B., Dan, A.: The enduring impact of transient emotions on decision making. *Organ. Behav. Hum. Decis. Process.* **109**(1), 1–8 (2009)
9. Harlé, K.M., Sanfey, A.G.: Incidental sadness biases social economic decisions in the ultimatum game. *Emotion* **7**(4), 876–881 (2007)

HMM Classifier Object Recognizing System in Brain–Computer Interface



**H. S. Anupama, Raj V. Jain, Revannur Venkatesh, N. K. Cauvery,
and G. M. Lingaraju**

Abstract Machine learning (ML) is the field that adds intelligence to devices providing them with capabilities to process and identify patterns in data just like human beings do. Programming devices in this manner can help in identifying those patterns which human beings often cannot. Machine learning is based on modelling data mathematically. ML has been gaining a lot of attention in the last few decades, especially in fields of interdisciplinary research. Brain–Computer Interface (BCI) is an area where Machine Learning Technology is been rapidly using. Also, Machine Learning techniques have to be used so that one can get a better result and more efficiency. Information Transfer Rate is the best way to measure the performance of the signals. The current research is mainly focused on achieving the systems with higher ITR. The focus of the proposed system is to get better and high Information Transfer Rate by merging two different approaches. The approach used in this work is (SSVEP), Visually Evoked Potential and (SSAEP) Auditory Evoked Potential by using Hidden Markova Model (HMM). The system which is to be developed checks whether the existing system has such facility if it has, does it provides accuracy which is of a higher rate and can put it in the real-world applications.

H. S. Anupama (✉)

Department of Artificial Intelligence and Machine Learning, BMS Institute of Technology & Management, Bangalore, India

e-mail: anupama_hs@rediffmail.com

R. V. Jain · R. Venkatesh

Department of Computer Science and Engineering, R. V. College of Engg, Bangalore, India
e-mail: rajjain4900@gmail.com

N. K. Cauvery

Department of Information Science and Engineering, R. V. College of Engg, Bangalore, India
e-mail: cauverynk@rvce.edu.in

G. M. Lingaraju

Department of Information Science, M. S. Ramaiah Institute of Tech., Bangalore, India
e-mail: gmlraju@gmail.com

Keywords Brain–computer interface · Hidden Markov Model (HMM) · Feature extraction · SSVEP · SSAEP

1 Introduction

All human brain has neurons within them. Neurons are responsible for the information passing from one part of the body to the other part of the body. Brain–Computer Interface is one such device that helps in getting the EEG signals from the brain which is been generated by the neural activity of the human brain. Mind Machine Interface is a device that is used to communicate with the human brain and any machines/computer to say, which uses the neuronal activity generated in the brain [1, 2].

This communication will not happen through the nerve system and through muscular activity but it has a separate mechanism for doing it. EEG signals generated in the brain can be recorded in two different ways that is an invasive method and non-invasive methods. Most of the research in BCI is devoted to the development of applications that can be used as a communication option for people with severe motor impairments. An example of a BCI system can be to control a robotic arm. BCI can be implemented in many ways. The evoked potential is one of the ways of implementing it. Stimuli that are used in evoked potential are also many. In our work audio and visual stimuli have been used.

Visually Evoked Potential (SSVEP): When a person sees an image for some time, there is a frequency generated in the brain. This frequency can be recorded by the signals that are evoked in the brain which shows a peak at the frequency which is been set for visual stimulus.

Auditory Evoked Potential (SSAEP): When a person hears audio for some time, there is a frequency generated in the brain. This frequency can be recorded by the signals that are evoked in the brain which shows a peak at the frequency which is been set for audio stimulus.

Evoked potential (EP) helps in decision-making. When a person is projected to two different stimuli with different frequencies in front of him, the object which the user sees and hears will generate the signals with higher amplitude and frequency. The obtained frequency will help to know what the user has chosen. Different frequencies are set for the stimuli to help the user to choose which stimulus he/she is focusing on. This system is proposed such that a person can enhance his decision-making capability by providing two stimuli. This will help in generating signals in different regions of the human brain. This may help the user to make a different decision simultaneously.

2 State of Art Development

Many surveys have been done on SSVEP-based BCI through which it has shown that there are three types of Repeated Visual Stimuli (RVS) that are presented in SSVEP-based BCIs [3–5].

- Light stimulus—consists of light sources like LEDs, Xe and fluorescent lights which are modulated at certain frequencies [6].
- Single graphics stimuli—are stimuli presented using display devices where the stimulus is rendered on the screen and it appears and disappears from the screen at certain frequencies
- Pattern reversal stimuli—are also rendered on display devices. They consist of patterns that alternate at a particular frequency, e.g. checkerboards.

It has been found that LED stimuli to elicit better SSVEP signals than the other types of stimuli. However, computer systems are more popular because it is easier to generate the required stimuli on such systems. The type of rendering device and the frequency of stimuli used also affects the SSVEP signals generated. Most of the papers use frequencies in the range 4–50 Hz. The best performing systems presented stimuli close to 10 Hz. Stimulus colour also affects the evoked potentials. Single graphics stimuli usually use black and white stimuli [7, 8]. However, it has been observed that the best performing BCIs have used green colour. Also, red stimuli generate the highest Information Transfer Rate (ITR).

3 Background

Brain–Computer Interface system which involves studying evoked response to a stimulus requires the stimulus to be accurate and optimal with respect to various parameters. Machine Learning [9] is a part of artificial intelligence which uses mathematical relationships between the features or characteristics to estimate the output, defined as “A machine learns with respect to a particular task denoted by T, performance metric represented by P and the type of experience represented by E, if the system reliably improves its performance P at task T, following experience E [10, 11].” Supervised machine learning algorithms are the ones that use both the features and the output values or labels to learn the relationship between the two. BCI needs machine learning because of the noisy data that is got from the instruments. For machine learning to be effective, data needs to process to get features that are fed into the machine learning algorithm. Feature extraction depends on the kind of data which is being considered. For image data, methods like Scale Invariant Feature Transform (SIFT), etc. are used, for signal data, methods like Fourier transform, etc. are used and so on.

SSVEP is a visual stimulus which is generated when a person views any of the objects. Here in this work visual stimulus is generated when a frequency of

6 Hz is been reached. That is the threshold that is been set for the visual stimulus. If it is below that value it is called Transient Visual Evoked Potential (T-VEP). The threshold for the stimulus is been set because if the frequency is more than 6 Hz and even after the stimulus is been removed one can see the signals. But in T-VEP that cannot be possible. To get better efficiency and better result the threshold for the visual stimulus is been set to 6 Hz [12, 13]. Similarly for Audio stimulus also a threshold frequency is been set to get better efficiency and better result for any object which is been projected.

4 Methodology

The methods Auditory Evoked Potential (SSAEP) and Visually Evoked Potential (SSVEP) [14] are two stimuli that are been used in our work which helps in the decision-making of the user. In SSVEP paradigm, two images are selected randomly. Those images are made to flicker at two different frequencies so that it easy for the user to observe the images. The frequencies selected for this work for the image flickering 7.5 Hz and 10 Hz, respectively. These frequencies are selected as threshold frequencies such that they do not clash with each other and are within the range of 6–24 Hz. In SSAEP signals are generated by audio input with different frequencies of the audio. To improvise and maximize the SNR (Signal to Noise Ratio) the frequency selected here is about 40 Hz. For example, the visual stimuli, the images that are selected is with respect to the food items and the auditory stimuli, the images are selected are with respect to the drink selection. Once the selection is been done, subject is made to understand the concept of how it has to be used both visual and audio simultaneously. After making them understand the training of each trial is started. Training trail is been done with the gap of 2 s such that it is called a resting period so that again when the training is been done it should not clash with the previous data which is been trained. Here a person will be asked to visualize and hear anyone object simultaneously such that signals generated for both of that will be recorded simultaneously for that person. This gives the subject to come to the decision by focusing on each one of it such that one will get baseline EEG data from the brain. Once we get the audio and video EEG data will filter the noise and extract the features using different feature extraction techniques. After the features are extracted, they will train the model for the features obtained. Machine learning algorithm is used for further classification. In our work, it has been used Hidden Markova Model for the process of classification of the features obtained from the brain signals [15, 16].

5 Implementation

The stages that are involved in BCI are acquiring the data from the headset; filtering the data once it is acquired, extracting the relevant features, classification and external application. Before the classification data has to be extracted, filtered and then classification has to be done. In this work, EEG signals are captured from the headset. The headset used in this work is non-invasive headsets. The reading obtained from the headset has to be stored somewhere in the form of any type of file. The file that is used to store the data in this work is.csv file to process further. After the signals are extracted, next is to filter those extracted signals in order to remove the noise, external and internal artifacts. After the filtering feature extraction technique is used to extract the features. Once the data is obtained from the headset, signals that are obtained consist of both relevant signals and irrelevant signals from the human brain. One has to extract only relevant data that is required for the user. So the signals obtained from the headset are the raw EEG signals. In feature extraction signals that are captured are raw EEG. From that raw EEG relevant information has to be extracted which can lead to good classification performance. Several measures have been devised for classifying object recognition. Machine learning algorithms are used for the classification of the data. Some machine algorithms use non-time-series approaches and some use time series. (HMM) Hidden Markov Model is one of the classification algorithms that has been used in this work for detecting the object.

(HMM) Hidden Markov Model is a stochastic model that incorporates the time series data by following the Markovian property [17, 18]. It is observed to be a finite state machine which has the value of the current state depending on a certain fixed number of previous states [19].

For this model, parameters are $\lambda = \{\pi_i, a_{ij}, b_{ij}\}$, where π_i is the probability that state i is the initial state, a_{ij} is the probability that transition from hidden state i to hidden state j would occur and b_{ij} is the probability that from a hidden state i there outputs the visible symbol j . The initial state probabilities help the model to choose the optimal path to predict the most probable sequence. a is called transition matrix and b is called the emission matrix.

Equation for HMM is as follows:

$$P(x_1, \dots, x_T, y_1, \dots, y_T) = P(x_1) P(y_1|x_1) \prod_{t=2}^T P(x_t|x_{t-1}) P(y_t|x_t) \quad (1)$$

The discrete HMM models require discrete symbols of hidden states [18]. For the classification procedure, one has to follow two different phases. First one is the training phase where the training of the data has to be done before starting with the experiment. Another phase is testing phase, once data is trained one has to test the data accordingly such that performance could be measured easily. The training and testing data are cross-validated before classification. In this work, HMM is used for the training phase and is also provided with the training decision sequences. In the testing phase, cross-validated part of the decision sequence is given to predict the

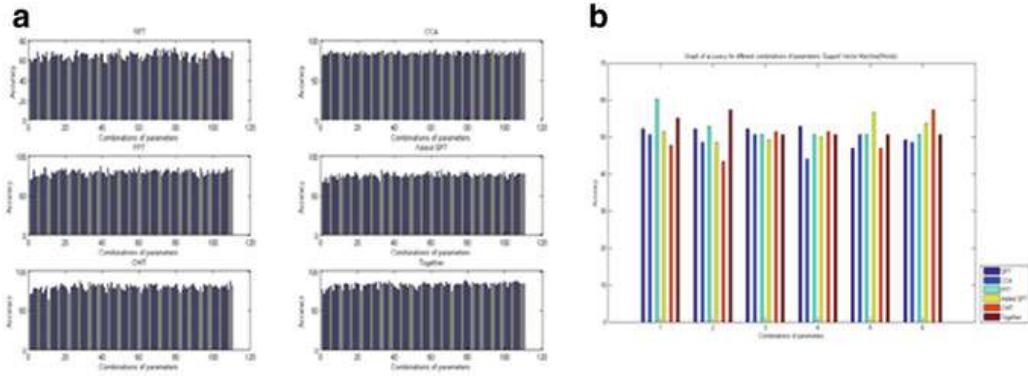


Fig. 1 **a** (Left), **b** (right): accuracy of SSVEP and SSAEP using HMM

Table 1 Accuracy results for time series machine learning algorithms

Algorithm	HMM	
	Max	Mean
Visual sine	100	87.5
Audio sine	100	93
Visual words	100	87.5
Audio words	100	90

transition and emission probabilities. Upon these results, log-likelihood is obtained to classify it to any of the classes to which it belongs.

The work was carried out in an isolated room such that no disturbance would come while collecting the raw EEG data. Noise-free room has to be selected because no disturbances should be there to get the signal from the brain. The participants who are willing to wear the EEG cap are also instructed to not wear any metals and electronic gadgets with them so that it will not disturb the accuracy of the signals. Data acquisition was done based on the above constraints for audio and video stimulus. Classification algorithm used here is Hidden Markov Model. BESS software stores the data in the EDF format. Data is collected from all the 16 channels of the electrodes. The output obtained from all the channels is evaluated comparing with the different metrics to measure the performance obtained of the system. Figure 1a, b shows the accuracy obtained for SSVEP and SSAEP using HMM model.

Maximum and mean accuracy results of HMM for visual and audio components are tabulated below in Table 1.

6 Result and Conclusion

This system was proposed to check if two BCI paradigms can be combined. The classification accuracies obtained shows that such combinations of the paradigms

can be made. The algorithm which best suits this is HMM which gives an average classification of 90%.

Another important aspect of the work was to improve the ITR. ITR obtained for the HMM model is about $ITR = 4.0858873$ bits/second or $ITR = 240.15324$ bits/min. (Considering N is the number of classes, Acc refers to the accuracy calculated in percentage, T is the time period for each trial).

Based on the results obtained the following conclusions were made:

- A combination of SSVEP and SSAEP can be used to effectively make two decisions at the same time.
- The best classification accuracy that was obtained was for HMM using a time series analysis of the data. The system was able to classify all the input data successfully resulting in 90% accuracy.
- Based on the average accuracy calculations, it was found that the classification of SSAEP for words stimulus has higher accuracy than for sine stimulus for the machine learning algorithms.

References

1. Vallabhaneni, A., et. al. Brain-computer interface. *Neural Eng.* 85–121 (2005)
2. Lance, B.J., et. al.: Brain-computer interface technologies in the coming decades. *Proc. IEEE* **100**, 1585–1599 (2012)
3. Zhu, D., et. al.: A survey of stimulation methods used in SSVEP-based BCIs. In: *Comput. Intell. Neurosci.* (2010). <https://doi.org/10.1155/2010/702357>
4. Zhang, Y., et. al.: LASSO based stimulus frequency recognition model for SSVEP BCIs. *J. Biomed. Signal Process. Control* 104–111 (2012). <https://doi.org/10.1016/j.bspc.2011.02.002>
5. Muller, S.M.T., et. al.: Incremental SSVEP analysis for BCI implementation. In: Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, Argentina, no. 32, pp. 3333–3336 (2010)
6. Stiles, W.S., Crawford, B.H.: Luminous efficiency of rays entering the eye pupil at different points. *Nature* 139(3510), 246–246 (1937)
7. Ng, K.B., et. al.: Effect of competing stimuli on SSVEP-based BCI. In: Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 6307–6310 (2011). <https://doi.org/10.1109/emb.2011.6091556>
8. Tello, R.M., et. al.: Evaluation of different stimuli color for an SSVEP-based BCI. In: Congresso Brasileiro de Engenharia Biomedica, pp. 25–28 (2014)
9. Mitchell, T.: Introduction. In: *Machine learning*. McGraw-Hill (1997). Chap. 1, Sect. 1.1, p. 2
10. Wang, Y., et. al.: Visual stimulus design for high-rate SSVEP BCI. *Electron. Lett.* **46**(15). <https://doi.org/10.1049/el.2010.9088>
11. Rueck, R., Guger, C.: A brain-computer interface based on steady state visual evoked potentials for controlling a robot. In: *Bio-Inspired Systems: Computational and Ambient Intelligence*, pp. 690–697 (2009)
12. Luo, A., Sullivan, T.J.: A user-friendly SSVEP-based brain-computer interface using a time-domain classifier. *J. Neural Eng.* **2** (2010). <https://doi.org/10.1088/1741-2560/7/2/026010>
13. Liu, Q., et. al.: Review: recent development of signal processing algorithms for SSVEP-based brain computer interfaces. *J. Med. Biolog. Eng.* 299–309 (2013). <https://doi.org/10.5405/jmbe.1522>

14. Korczak, P., et al.: Auditory steady-state responses. *J. Am. Acad. Audiol.* **23**(3), 146–170 (2012). <https://doi.org/10.3766/jaaa.23.3.3>
15. Lopez, M.A., et. al.: Evidences of cognitive effects over auditory steady-state responses by means of artificial neural networks and its use in brain-computer interfaces. *Neurocomputing* 3617–3623 (2009). <https://doi.org/10.1016/j.neucom.2009.04.021>
16. Ng, K.B., et. al.: Effect of posterized naturalistic stimuli on SSVEP-based BCI. In: Proceedings of Annual Int IEEE Engineering in Medicine and Biology Society Conference, pp. 3105–3108 (2013). <https://doi.org/10.1109/embc.2013.6610198>
17. Lee, H., Choi, S.: PCA + HMM + SVM for EEG pattern classification. In Proceedings of Signal Processing and its Applications, vol. 1 (2003). <https://doi.org/10.1109/isspa.2003.1224760>
18. Argunsahy, A. O., Cetin, M.: AR-PCA-HMM approach for sensorimotor task classification in EEG-based brain-computer interfaces. In: International Conference on Pattern Recognition (2010). IEEE <https://doi.org/10.1109/ICPR.2010.3>
19. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* **195**(1), 215–243 (1968)

Deep Learning for Stock Index Tracking: Bank Sector Case



R. Arjun, K. R. Suprabha, and Ritanjali Majhi

Abstract The current study explores the efficacy of deep learning models in stock market prediction specific to banking sector. The secondary data of major fundamental indicators and technical variables during 2004–2019 periods of two banking indices, BSE BANKEX and NIFTY Bank of Bombay stock exchange and National stock exchange, respectively, are collected. The factors impacting market index prices were analyzed using nonlinear autoregressive neural network. Preliminary findings contradict the general random walk hypothesis theory and model improvement over previous studies. The implications from practical and theoretical perspective for stakeholders are discussed.

Keywords Stock market · Banking sector index · Time-series prediction

1 Introduction

Deep learning is a fast emerging machine learning model with powerful and robust results having diverse applications. Though many models and methods were proposed in literature, skepticism arises on accuracy and efficiency due to mixed results [5]. Stock market prediction is considered challenging and wicked problem because of no concrete or comprehensive solution available till date and high stochastic nature of data involved. Moreover, there is apparent impact of stock market on future economic growth [17]. A number of factors have been investigated broadly like micro-economic conditions, macro-economic variables, political

R. Arjun (✉) · K. R. Suprabha · R. Majhi

School of Management, National Institute of Technology Karnataka (NITK), Surathkal,
Surathkal, Karnataka 575025, India

e-mail: arjrs123@gmail.com

K. R. Suprabha

e-mail: suprabha.kr@gmail.com

R. Majhi

e-mail: ritanjalimajhi@gmail.com

scenario, investment friendliness, etc., that impact market. Further, index portfolio based on deep learning has shown good performance on tracking Hang Seng index (HSI) of Hong Kong [16]. An industry analyst's uses past historic market data, current conditions and expert intuition to forecast the estimates [14]. Majority of previous works focused on models tested in broad markets/indices but specific in Indian scenario are missing; thus addressed in the current study. Also, banking at national level being a key component in service sector warrants for paramount importance considering its overall economic impact.

In this study, an effort is taken to develop deep learning model to predict the stock market indices pricing for two major indices from banking sector in India. The rest of paper is structured as follows; Sect. 2 describes related work, Sect. 3 explain the methodology adopted in study, Sect. 4 summarizes the results and implication and Section 5 outlines the limitations and future scope of work.

2 Related Work

Deep learning as effective method has already explored and experiments result in emerging markets [18, 20]. Chong et al. [4] tested such models in the Korean stock exchange prediction. Hiransha et al. [9] used tested models on datasets both from NSE and New York stock exchange suggesting convolutional neural network (CNN) is effective in both cases than autoregressive integrated moving average (ARIMA). Singh and Srivastava [22] used deep learning methods on NASDAQ stock data to predict Google stock price. Balaji et al. [2] taken the banking sector of BSE and opined that for 4-step and 1-step ahead stock directional movements, extreme learning machine (ELM)-based models gave best performance. Fawaz et al. [6] reviewed time-series classification for data mining-based applications. Borovkova and Tsiamas [3] had ensemble methods long short term memory (LSTM) for high frequency data. Specific Indian case studies also reported positive results [13]. The focus on market analysis [12] has drawn from innovations forayed to industry [11]. Recent work utilizing tick data of 1 month period is shown to be efficient over 15-min test cycle for market forecasts [21]. Thus, literature review shows a dearth of empirical studies on deep learning models for sector-specific prediction tasks applicable in Indian context.

3 Methodology

3.1 Sampling and Datasets

The secondary data of indices; BSE Bankex and NSE NIFTY Bank were taken from their official sources, https://www.bseindia.com/market_data.html and www.

nseindia.com/, respectively. The entire period of indices after creation is being collected, i.e., 2005–2019 (14 years) for NIFTY in daily frequency and BSE monthly from 2005 to 2019. The variables involved are index price, open price, high price, low price and volume of transactions used for technical analysis. Under the fundamental ratios, the price-to-earnings (P/E) ratio, price-to-book (P/B) ratio and dividend yield are considered due to availability and legal permissibility [8]. Such indicators are commonly utilized based from survey of field [15].

Figure 1 shows the visualization for dataset pertaining to BSE Bankex index. Total number of data points are 1494 including all variables. Also, no missing data was present during the analysis stage. Figure 2 shows the NIFTY Bank index dataset chart in daily frequency. The total observations are 44,581. For volume of transactions, 448 missing data were present, since data was not prior to March 29, 2007. To examine the reasons of abrupt price deviations in dataset, industry news and policy announcement events were collected. From sectorial view point, on May 18, 2009, after election results, BSE Bankex moved 20.27 points higher (18.81% change) in single day. Similarly, for NIFTY Bank, there was record high volume in October 25, 2017 due to government announcement of 2.11 lakh crore rupee stimulus package to recapitalize PSU Banks. Hence, forecasting market and mitigating crisis based on sector analysis processing news are also vital [19]. The sample size design is found to be optimum from earlier literature comparing similar models.

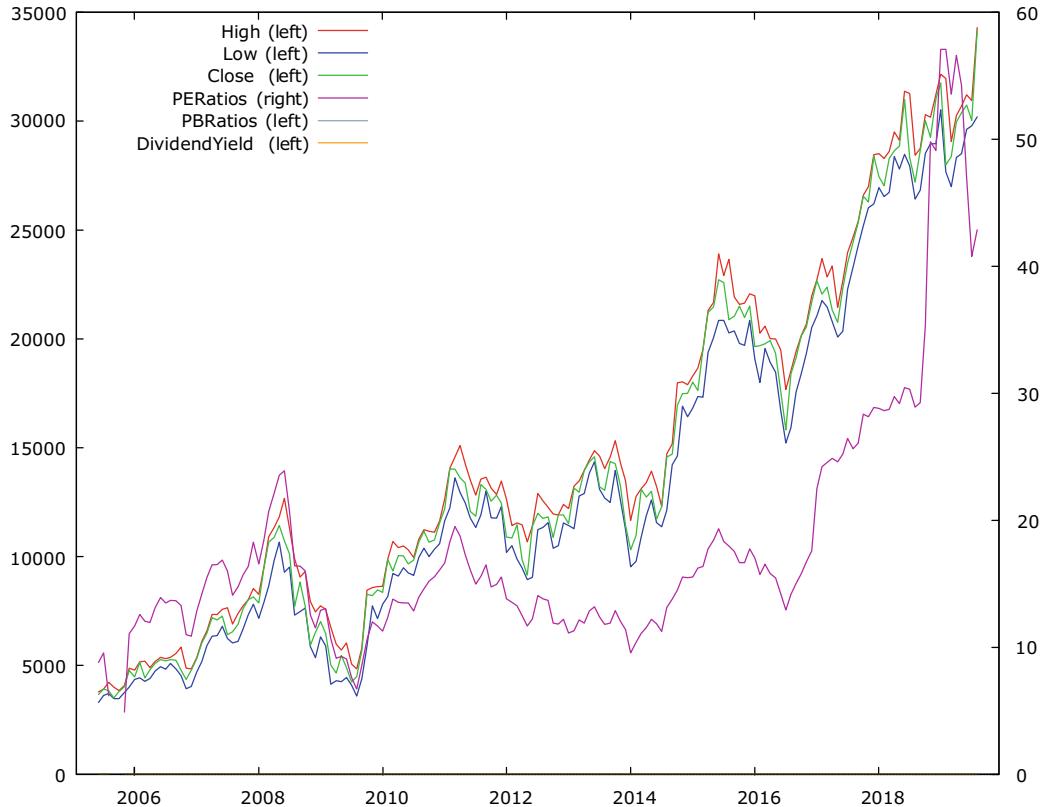


Fig. 1 BSE Bankex data visualization (*source* GRETL visualization)

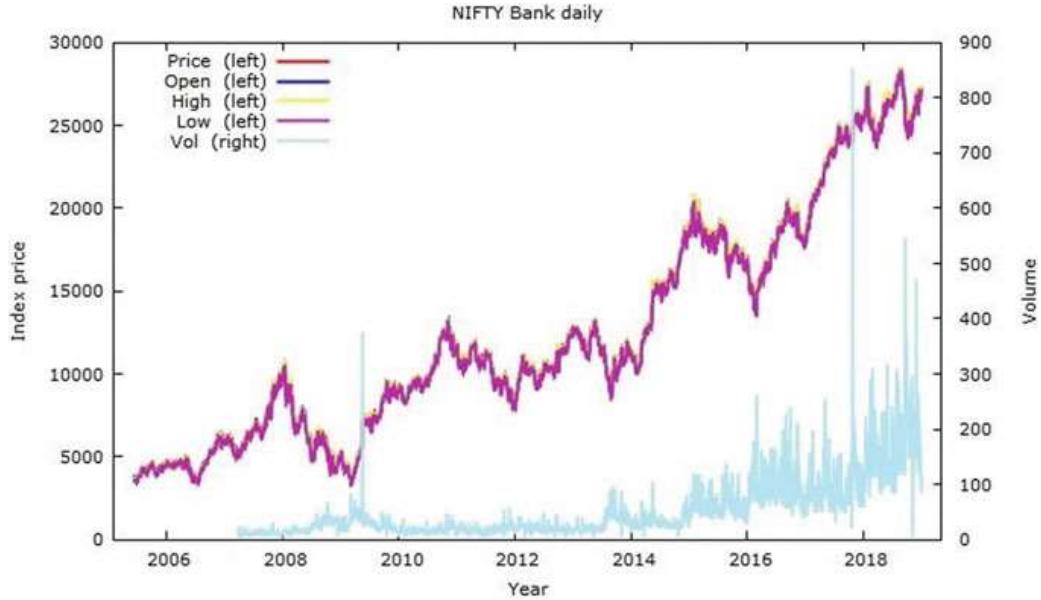


Fig. 2 NSE NIFTY Bank daily data visualization (*source* GRETL visualization)

3.2 Analysis

See Figs. 1 and 2.

3.3 Experimental Setup

In work, both MATLAB 2019a and GRETL ver. 2019c are used in simulation. The prior assumption of study is random walk hypothesis is debatable in banking sector and certain level of predictability prevails to estimate market index. To empirically state the proposition, correlation value must be 0 or by Kolmogorov–Smirnov test but results proved contrary. Among prediction performance used, R -square (R^2) model fitness statistic and mean squared error (MSE) are noted. The close price of stock is important as deep learning compared with PCA approach gave positive results [7]. Tables 1 and 2 give snapshot of BSE Bankex and NIFTY Bank datasets.

Technical indicators used in study followed are listed below:

1. Typical price (TP): Uses high, low and close price given as $(Hi + Lo + Cl)/3$
2. Volume rate of change (VROC): $(Vo-Vot-14)/Vot-14$. (14 day period is taken)
3. Exponential moving average (EMA): This is calculated as moving average = $[Price - previous\ EMA] * (2/n + 1) + previous\ EMA$. $n = 20$ days is assumed.
4. Weighted close (WC): Calculated by equation $(Hi + Lo + Cl*2)/4$.

There are total 166 records in the dataset including technical and fundamental factors.

Table 1 BSE Bankex data with computed technical indicators

Year	High	Low	Close	Weighted close (WC)	Typical price (TP)	P/E	P/B	Dividend yield
Aug-05	4789.76	4365.56	4468.51	4523.09	4541.28	11.67	2.2	1.61
Sep-05	5168.00	4440.74	5125.01	4964.69	4911.25	12.58	2.23	1.56
–	–	–	–	–	–	–	–	–
Feb-19	30,949.07	29,787.55	30,027.41	30,197.86	30,254.68	40.77	2.39	0.35
Mar-19	34,294.56	30,199.47	34,141.94	33,194.48	32,878.66	42.87	2.54	0.33

A total of 3468 records are present in dataset of NSE NIFTY Bank. After the data collection, pre-processing is done to isolate missing data, sparse values and outliers.

4 Results and Discussion

The correlation analysis and simulations are performed. For BSE, the following results are obtained from GRETL. Negative correlation is observed of typical price to div. yield (-0.7878) with highest correlation to P/E ratio (0.7887). Correlation coefficients use the observations 2005:06–2019:03. Under 5% critical value (two-tailed) = 0.1524 , for $n = 166$. From NIFTY Bank dataset, the coefficients use observations in March 03, 2007 to June 06, 2019 (missing values skipped). With 5% critical value (two-tailed) = 0.0357 , $n = 3014$. The model is implemented using MATLAB version 2019a. Only BSE monthly frequency data is used for simulation. *ntstool*, a neural network wizard is being run in experiment (Fig. 3). Following results by [1, 25], nonlinear autoregressive exogenous model (NARX) is tested due to high stochastic nature of variables in dataset. The problem is defined as to predict a series $y(t)$ given d past values of $y(t)$ and another series $x(t)$ as in Eq. 1. Input attributes are assigned $x(t)$ columns and $y(t)$ target variable, i.e., index closing. Noise/error term is denoted with ε .

$$y(t) = f(x(t-1), \dots, x(t-d), y(t-1), \dots, y(t-d)) + \varepsilon_t \quad (1)$$

The tool randomly divides total 1328 time-steps in datasets. Of this, training is 70% of data with testing and validation of 15% each. Algorithm used is Levenberg–Marquardt as in [23], no. of hidden neurons are 10 and delay units d , set to 1. Such configuration is chosen as past literature hints that increasing number of hidden layer cannot guarantee higher accuracy. Such can lead to overfitting issues [10] or trade-off situation with performance/model complexity. The iterations stopped at 13 epochs, with best performance instance in epoch 7 after retraining (Fig. 4). The training regressions are shown in validation tend to reduce R^2 (Fig. 6). The model can generate accurate prediction estimates up to using 37 data points or $3\frac{1}{2}$ years

Table 2 NSE NIFTY Bank index data with technical indicators

Date	Price	Open	High	Low	Vol	SMA	EMA	VROC	Change%	P/E	P/B	Div. yield
23-Apr-07	5550.15	5622.00	5637.10	5536.05	8.01	5326.505	5574.06	-36.1244 0191	-0.85%	15.9	2.52	1.2
-	-	-	-	-	-	-	-	-	-	-	-	-
31-May-19	31,375.40	31,678.90	31,783.60	30,623.05	230.85	29,974.305	31,524.65	-8.970820189	-0.51%	64.93	3.53	0.4

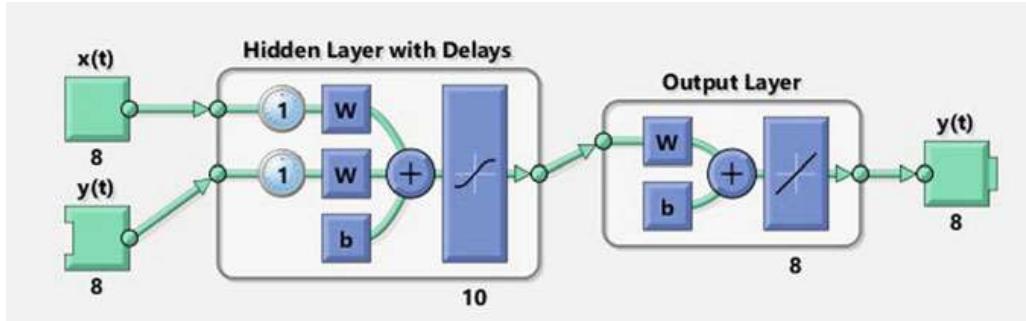
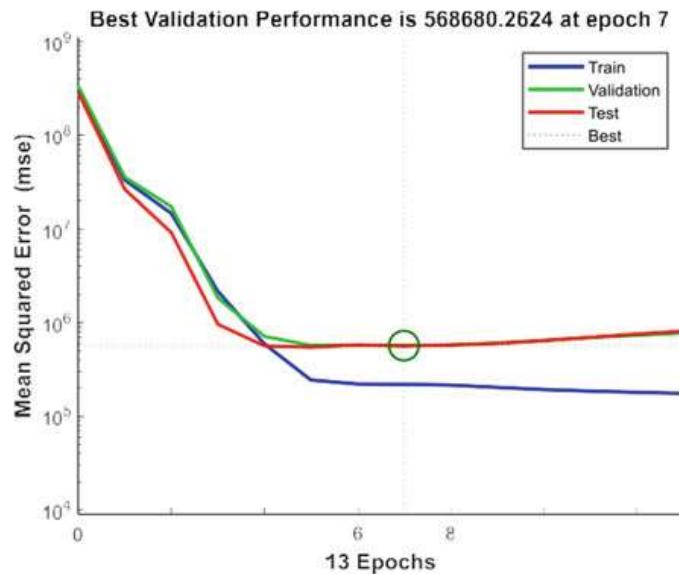


Fig. 3 NARX network model architecture (*source* MATLAB visualization)

Fig. 4 Model performance
(*source* MATLAB
visualization)



of monthly datasets. Higher error-rate is observed after 40–110 time-steps (Fig. 5). Usage of daily frequency data may enhance precision parameters of model assuming that bias and overfitting instances are negligible.

5 Conclusions

The analysis revealed that higher correlation exists of typical price technical indicator with earnings. Also, price-book value being reflected in quarterly results impacted more in BSE index price. Since volume of transactions has higher correlation than book value signifies NSE has greater impact from earnings announcement events. One of key limitations of study is that only historical market data is utilized, but industry analyst accounts real-time factors too in market estimation.

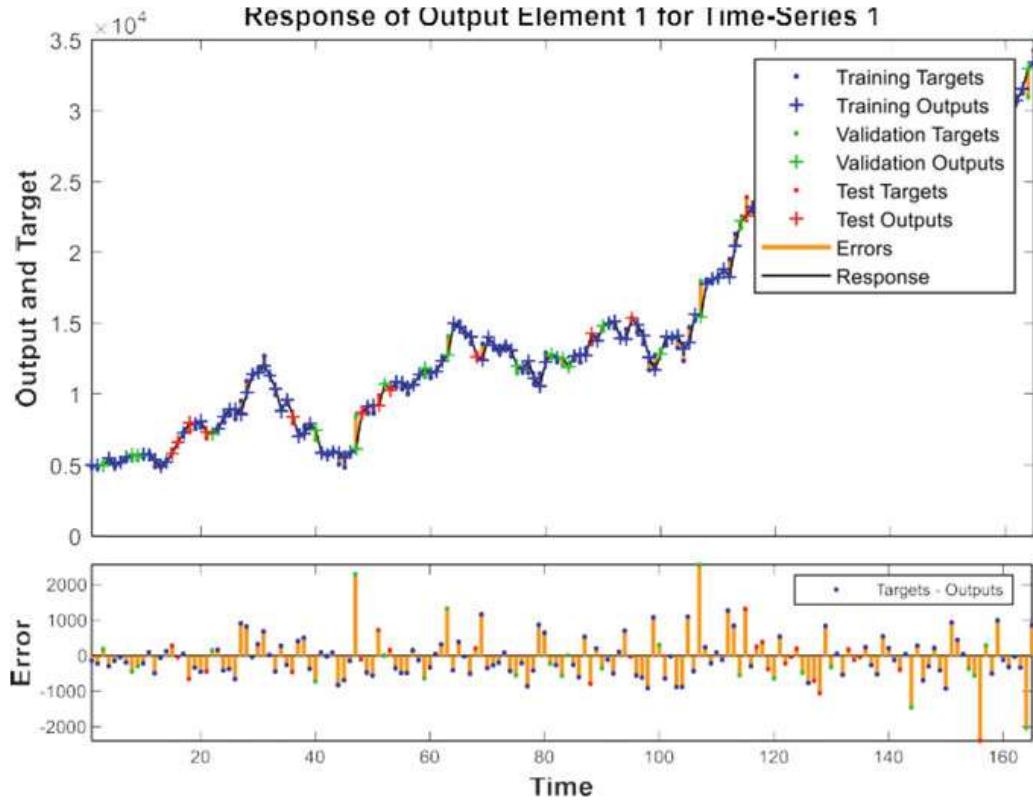


Fig. 5 Time-series response chart (source MATLAB visualization)

For example, effects arising from volatility, inflation were not assumed. Robustness may be improved by tweaking the architecture/training algorithm. A cross-validation procedure and out-of-sampling also are crucial. Current results are comparable from higher R^2 and lesser iteration epochs than reported in [1]. Adapting model with qualitative aspects of trading sentiments extracted from social media sources is an interesting direction [24]. While addressing the caveats remains future scope, focus is to develop enterprise standard applications. Finally, stakeholders and policy designers must analyze internal market structure for decision making or recommendations.

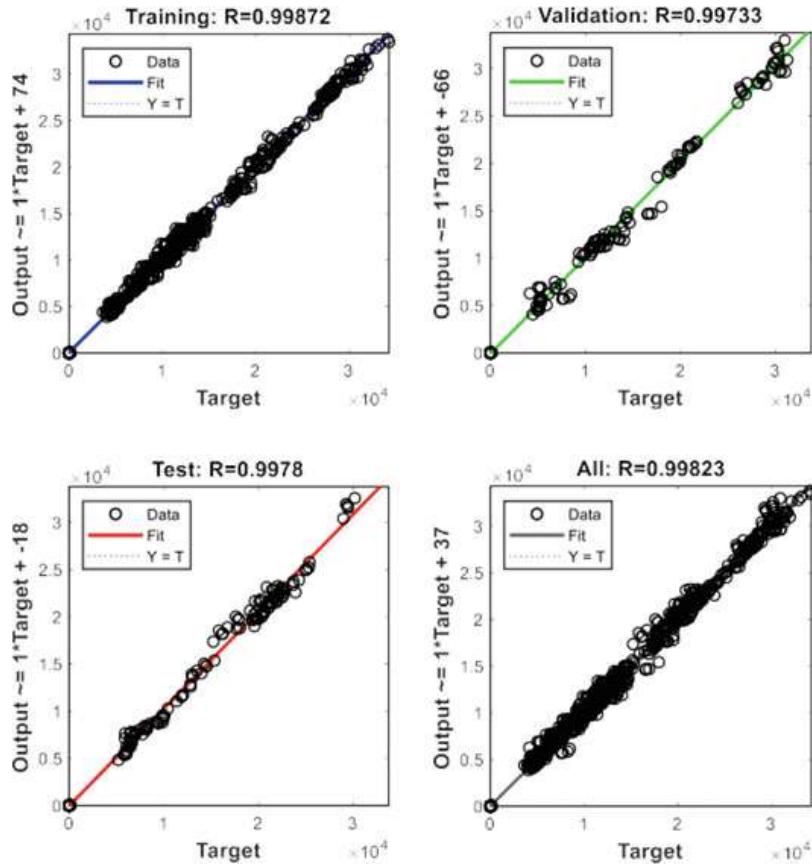


Fig. 6 Neural network training regressions (source MATLAB visualization)

References

1. Alkhoshi, E., Belkasim, S.: Stable stock market prediction using NARX algorithm. In: Proceedings of the 2018 International Conference on Computing and Big Data, 8 Sept 2018, pp. 62–66. ACM (2018)
2. Balaji, A.J., Ram, D.H., Nair, B.B.: Applicability of deep learning models for stock price fore-casting an empirical study on Bankex data. Procedia Comput. Sci. **1**(143), 947–953 (2018)
3. Borovkova, S., Tsiamas, I.: An ensemble of LSTM neural networks for high-frequency stock market classification. J. Forecast. **38**, 600–619 (2019). <https://doi.org/10.1002/for.2585>
4. Chong, E., Han, C., Park, F.C.: Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. Expert Syst. Appl. **15**(83), 187–205 (2017)
5. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Twenty-fourth international joint conference on artificial intelligence, 2015 Jun 25 (2015)
6. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Min. Knowl. Disc. **33**(4), 917–963 (2019)
7. Gao, T., Li, X., Chai, Y., Tang, Y.: Deep learning with stock indicators and two-dimensional principal component analysis for closing price prediction system. In: 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 26 Aug 2016, IEEE. pp. 166–169 (2016)
8. Hickman, K., Petry, G.H.: A comparison of stock price predictions using court accepted formulas, dividend discount, and P/E models. Financ. Manage. **1**, 76–87 (1990)

9. Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., Soman, K.P.: NSE stock market prediction using deep-learning models. *Procedia Comput. Sci.* **1**(132), 1351–1362 (2018)
10. Leinweber, D.J.: Stupid data miner tricks: overfitting the S&P 500. *J. Investing.* **16**(1), 15–22 (2007)
11. Levinson,, R.: Inventor. System and Method for Predicting Stock Prices. United States Patent Application US 10/970,892. 28 Apr 2005
12. Li, X., Cao, J., Pan, Z.: Market impact analysis via deep learned architectures. *Neural Comput. Appl.* 1–2 (2018)
13. Naik, N., Mohan, B.R.: Stock price movements classification using machine and deep learning techniques—the case study of indian stock Market. In: International Conference on Engineering Applications of Neural Networks 2019, pp. 445–452. Springer, Cham
14. Nofer, M., Hinz, O.: Are crowds on the internet wiser than experts? The case of a stock prediction community. *J. Bus. Econ.* **84**(3), 303–338 (2014)
15. Nti, I.K., Adekoya, A.F., Weyori, B.A.: A systematic review of fundamental and technical analysis of stock market predictions. *Artif. Intell. Rev.* 1–51 (2019)
16. Ouyang, H., Zhang, X., Yan, H.: Index tracking based on deep neural network. *Cognitive Syst. Res.* **1**(57), 107–114 (2019)
17. Pradhan, R.P., Arvin, M.B., Hall, J.H., Bahmani, S.: Causal nexus between economic growth, banking sector development, stock market development, and other macroeconomic variables: the case of ASEAN countries. *Rev. Financ. Econ.* **23**(4), 155–173 (2014)
18. Pun, T.B., Shahi, T.B.: Nepal stock exchange prediction using support vector regression and neural networks. In: 2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAEC), 9 Feb 2018, IEEE. pp. 1–6 (2018)
19. Rönnqvist, S., Sarlin, P.: Detect & describe: deep learning of bank stress in the news. In: 2015 IEEE Symposium Series on Computational Intelligence 2015, IEEE, pp. 890–897 (2015)
20. Samarawickrama, A.J., Fernando, T.G.: A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. In: 2017 IEEE International Conference on Industrial and Information Systems (ICIIS) 2017 Dec 15, IEEE, pp. 1–6 (2017)
21. Selvamuthu, D., Kumar, V., Mishra, A.: Indian stock market prediction using artificial neural networks on tick data. *Financ. Innov.* **5**(1), 16 (2019)
22. Singh, R., Srivastava, S.: Stock prediction using deep learning. *Multimedia Tools Appl.* **76**(18), 18569–18584 (2017)
23. Ticknor, J.L.: A Bayesian regularized artificial neural network for stock market forecasting. *Expert Syst. Appl.* **40**(14), 5501–5506 (2013)
24. Wang, Y.: Stock market forecasting with financial micro-blog based on sentiment and time series analysis. *J. Shanghai Jiaotong Univ. (Science)* **2**(2), 173–179 (2017)
25. Wibowo, A., Pujianto, H, Saputro, D.R.: Nonlinear autoregressive exogenous model (NARX) in stock price index's prediction. In: 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE) 2017 Nov 1, IEEE, pp. 26–29 (2017)

Rank Consensus Between Importance Measures in Hypergraph Model of Social Network



Debasis Mohapatra and Manas Ranjan Patra

Abstract In social network (SN), a node is considered as a social entity and a link defines the connection between social entities. In general, the link is shown as a dyadic relationship which is unable to represent a group having super-dyadic relationship. Hypergraph model of a network preserves the super-dyadic relation between the nodes. Several algorithms have been developed to measure the node importance and ranking the nodes according to importance. Some measures take less time, whereas some take more time. We propose a method to find the correlation between the different importance measures in hypergraph. By establishing high correlation, the ranking of a time inefficient importance measure can be computed from a time-efficient measure. In this paper, we present our contribution in twofold. At first, we show the construction of primal/Gaifman graph from hypergraph. Secondly, we establish the correlation between the different importance measures that are used for ranking the nodes of a hypergraph.

Keywords Social network · Hypergraph · Consensus

1 Introduction

SN is a network of social entities. Generally, the networks are represented in the form of a graph. The drawback associated with the usual graph representation is that it represents the relationship between entities as dyadic that is able to depict the relationship between only two entities. But this representation lacks in depicting

D. Mohapatra (✉)

Department of Computer Science & Engineering, Parala Maharaja Engineering College,
Berhampur 761003, India
e-mail: devdisha@gmail.com

M. R. Patra

Department of Computer Science, Berhampur University, Berhampur, Odisha 760007, India
e-mail: mrpatra12@gmail.com

the relationship that involves multiple entities like in co-authorship network, co-citation network, etc. The inefficiency of a graph model that is based on dyadic connections leads to the generalization of edge to a hyperedge and a graph to a hypergraph. A hyperedge can connect any number of nodes. Hence, a hypergraph is able to represent a super-dyadic relation. The hypergraph is a good fit for representing groups in SN. Finding most influential nodes in SN is important for understanding the reason of viral memes, product advertising, etc. But most of the findings in this area are based on graph model and its applicability to hypergraph model is a least explored area. Several methods of importance measurement are proposed in literature to discover the ranking of the nodes in a graph. In this paper, we consider three types of measures (1) measure based on centrality, (2) measure based on degeneracy and (3) measure based on information diffusion. In this paper, we at first, present algorithms that construct a hypergraph and convert the hypergraph to Gaifman graph that is a simple graph representation of hypergraph. Secondly, we find the consensus between different importance measures. The experiment confirms the consensus group by using Spearman's rank correlation. It shows two groups of consensuses, i.e., {Closeness Centrality, Degree Centrality, Betweenness Centrality} and {Core Number, Cascade Capacity}.

The rest of the paper is organized as follows. In Sect. 2, we discuss related work. We discuss problem definition and methodology in Sect. 3. The result with discussions is shown in Sect. 4 and at last Sect. 5 concludes the paper.

2 Related Work

Different types of centrality measures are discussed for graph model of social networks [1]. But relatively less amount of works is found on centrality measure in hypergraph. The affiliation network can be represented by hypergraph model. In [5], the centrality in affiliation network is explained. The paper [2] represents a general understanding on hypergraph centrality using incidence matrix. General centrality measurement of hypergraph is presented in [3]. The paper [8] proposes a co-operative game theoretic-based approach. It proposes the computation of Shapley value-based centrality without losing the super-dyadic information. The authors of [7] investigate the effects of topology and density on degree, closeness, betweenness and eigenvector centrality. The paper has reported the existence of significant correlations, i.e., degree-betweenness, betweenness-eigenvector, degree-closeness and closeness-eigenvector. Zhao et al. [10] proposed that a greedy algorithm is computationally expensive to find optimal result for k-node seed set. Hence, a k-shell decomposition algorithm for influence maximization is proposed under the linear threshold model. In [4], the objective is to maximize the cascade capacity by adding a few numbers of edges. It shows a close correlation between k-shell and eigenvector centrality. Kitsak et al. [6] have proposed a k-shell decomposition approach to find the most prominent nodes in a network. The influence propagation in hypergraph is discussed in [9].

3 Problem Definition and Methodology

Definition (Problem Statement): Given n different measures for importance. We first convert the hypergraph H to a primal graph G . Then, assigning ranking to the nodes using n different ranking R_1, R_2, \dots, R_n . We compute the consensus between all pairs of ranking, i.e., consensus (R_i, R_j) where $i \neq j$.

In this section, we propose an approach to find the consensus between five different importance measures in a hypergraph. The computation of centrality and influence is difficult in the original hypergraph representation as it maintains a high level of abstraction of the relationship between the entities in the form of a hyperedge. Hence, a low-level representation can be established by Gaifman equivalent of a hypergraph. This type of conversion is a one-way conversion as the Gaifman graph misses the subset information, and regenerating hypergraph from Gaifman graph is difficult. But it is possible only when we can provide a proper labeling of edges during the transformation process. The overall methodology consists of the following three steps.

- (i) Construction of a random hypergraph.
- (ii) Converting the hypergraph to Gaifman graph, i.e., finding the clique representation of the hypergraph.
- (iii) Applying the consensus algorithm between five different measures: Degree centrality (DC), closeness centrality (CC), betweenness centrality (BC), core number (CN) and cascade capacity (Cas_C).

3.1 Construction of a Random Hypergraph

We propose an algorithm that constructs a random hypergraph. This algorithm uses incidence matrix representation to generate a hypergraph as it is difficult to construct a hypergraph from an adjacency matrix. This Algorithm 1 takes n and m as inputs and returns a hypergraph having n vertices and m hyperedges.

Algorithm 1: RandomHypergraph (n, m)

Input:- n - Number of nodes, m - Number of edges

Output:- Returns hpergraph H

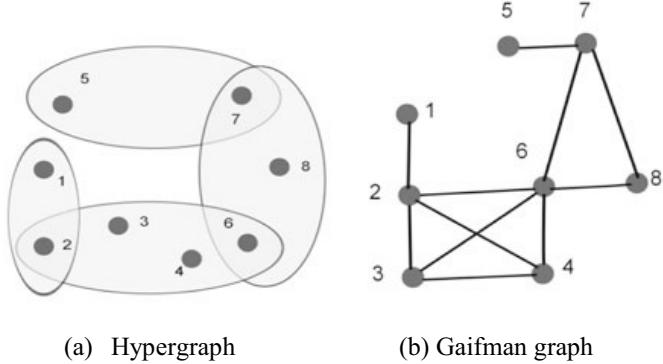
1. Create a random binary Incidence Matrix IM of order (n, m)
2. **for** each column C_i in IM
3. **if** (column sum of $C_i \leq 2$)
4. convert some 0s of C_i to 1s to make $C_i > 2$
5. **endif**
6. **end for**
7. **for** $i \leftarrow 1$ to $m - 1$
8. **if** $C_i \cap C_{i+1} = \emptyset$
9. Convert a node x from 0 to 1 in both C_i and C_{i+1}
10. **endif**
11. **end for**
12. **return**($H(IM)$)

Description: The steps 1–6 of Algorithm 1 create a random incidence matrix (IM). The steps 1–5 set the cardinality of each hyperedge to at least 3. Steps 7–11 converts a disconnected hypergraph to connected hypergraph. At last, step 12 returns the hypergraph that is represented by IM . Briefly, this algorithm returns a random connected hypergraph where cardinality of each hyperedge is at least 3.

3.2 Convert Hypergraph to Gaifman Graph

To analyze the importance of the nodes of a graph, we convert the hypergraph into Gaifman graph. This conversion is essential as the representation of the actual connectedness of two nodes is possible through Gaifman graph. In this representation, a hyperedge is represented as a clique. This conversion enables to define the importance of the nodes of a hypergraph as importance of nodes in the corresponding Gaifman graph. Figure 1b is the Gaifman equivalent of the hypergraph in Fig. 1a. The complete procedure follows Algorithm 2.

Fig. 1 **a** Hypergraph and
b Gaifman graph



Algorithm 2: Hypergraph_To_Gaifman(H)

Input:- H - The hypergraph $H = (V, HE)$ where V is the set of vertices and HE is the set of edges.

Output:- Returns Gaifman graph G .

1. $G[V] \leftarrow H[V]$
 2. **for** each h in HE
 3. $v1 \leftarrow$ set of vertices in h
 4. Form_clique($G, v1$)
 5. **end for**
 6. **return**(G)
-

Description: The set of vertices in the hypergraph and corresponding Gaifman graph is same, that is, as shown in step 1. In steps 2–5, for each hyperedge h of the hypergraph H , all vertices that constitute the hyperedge h form a clique by applying Form_clique () method on the Gaifman graph G . At last, step 6 returns G .

3.3 Consensus Between Importance Measures

At first, the algorithm performs an individual ranking by using the five different measures of importance: Degree centrality (DC), closeness centrality (CC), betweenness centrality (BC), core number (CN) and cascade capacity (Cas_C) to find the influential nodes in the Gaifman equivalent of hypergraph. The description of importance measures is given in Table 1.

Secondly, it uses a consensus algorithm that finds the frequency of matching between two different rankings. The complete algorithm is presented as Algorithm 3.

Table 1 Importance measures with description

Importance measure	Description
Degree centrality	$DC(v) = \frac{d(v)}{n-1}$, where $d(v)$ denotes the degree of node v and $n - 1$ is the maximum degree of a node as n is the number of nodes
Closeness centrality	$CC(v) = \frac{n-1}{\sum_{u \in V, u \neq v} dist(u, v)}$, where $dist(u, v)$ is the geodesic distance between u and v . V represents the set of nodes
Betweenness centrality	$BC(v) = \sum_{x \neq y \neq v} \frac{\pi_{x,y}(v)}{\pi_{x,y}}$, where $\pi_{x,y}$ is the total number of shortest paths between x and y . $\pi_{x,y}(v)$ denotes the total number of shortest paths between x and y passing through v
Core number	$CN(a)$ = Shell number of node a in k-shell decomposition
Cascade capacity	$Cas_C(a) = \frac{\sum_{i=1}^m Cas_C_i(a)}{n}$, where $Cas_C(a)$ is the number of average nodes influenced by node a

Algorithm 3: ConsensusCompute (V)

Input:- V -It is the list of nodes present in the Gaifman graph $G=(V, E)$.

Output:- Returns consensus matrix “ CM ”.

1. $L_1 \leftarrow \text{Sort_descend}(\text{FindDC}(V))$
2. $L_2 \leftarrow \text{Sort_descend}(\text{FindCC}(V))$
3. $L_3 \leftarrow \text{Sort_descend}(\text{FindBC}(V))$
4. $L_4 \leftarrow \text{Sort_descend}(\text{FindCN}(V))$
5. $L_5 \leftarrow \text{Sort_descend}(\text{FindCas_C}(V))$
6. for all pairs of ranking (L_i, L_j) , where $i \neq j$ finds the no. of matching and store the corresponding results in CM_{ij}
7. **return** (CM)

Description: In steps 1–5, it first finds the importance of the N nodes by DC , CC , BC , CN and Cas_C , then all the nodes are sorted in descending order of their importance and the rank lists L_1, L_2, L_3, L_4 and L_5 are formed. The step 6 computes the number of matching between two different rankings L_i and L_j . The step 7 returns the count matching matrix CM .

4 Results and Discussion

4.1 Experimental Setup

Our experiments are executed on a 2.40 GHz Intel(R) Core (TM) i7-4770 processor with memory support of 4 GB and Microsoft Windows 8.1 professional operating system. Implementation is done in Python using Networkx tool. The datasets are

generated by executing Algorithm 1 multiple times. Here, we have created 10 different hypergraphs (Table 2) for our experimental evaluation.

4.2 Implementation and Consensus Finding

All the algorithms are implemented in Python with network analysis support from Networkx tool. By using the Algorithm 2 all 10 hypergraphs are converted to their equivalent Gaifman graph. The CM matrix for all the hypergraphs are computed using Algorithm 3. The four prominent matching in ranking is found to be CC-DC, DC-BC, BC-CC and CN-Cas_C as shown in Fig. 2.

4.3 Spearman's Rank Correlation

The rank consensus is confirmed by evaluating Spearman's rank correlation coefficient also known as Spearman's rho. Here, it finds the degree of correlation exist between two importance measures. It can be defined as:

$$\rho = 1 - \frac{6 \sum_{i \neq j, i \in SM, j \in SM} d_{i,j}^2}{n(n^2 - 1)}, \quad (1)$$

where SM is the set of importance measures, $d_{i,j}$ is the difference between ranking of i th measure and j th measure and n is the number of nodes present in the network. The same order of consensus is preserved by Spearman's rho as shown in Fig. 3. The average computation of fraction of matching and Spearman's correlation also maintain the same order as shown in Table 3.

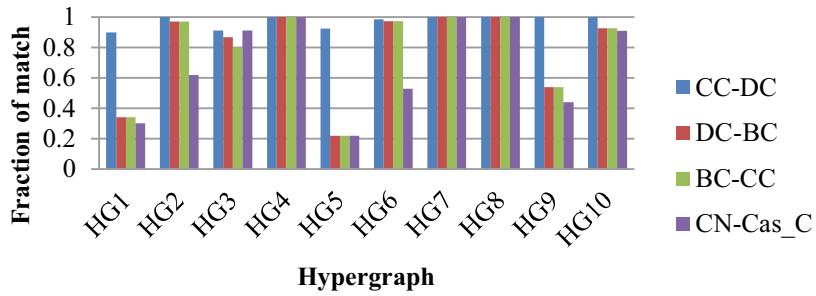
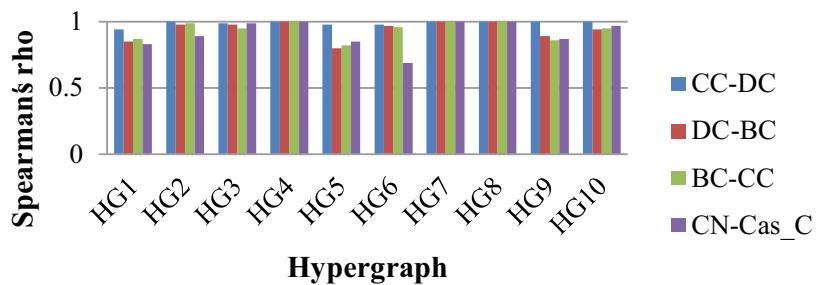
From the experimentation, we find a transitive closure dependency between *CC*, *DC* and *BC*. However, the five measures can be categorized into two consensus group, i.e., {*CC*, *DC*, *BC*} and {*CN*, *Cas_C*}. Hence, in a random hypergraph model, the local measure *DC* can be used to obtain the approximate ranking of nodes for *BC* and *CC*. Likewise, the semi-global measure *CN* can approximate the node ranking of *Cas_C*.

5 Conclusion and Future Work

This paper proposes an approach to establish consensus between five different importance measures: *CC*, *DC*, *BC*, *CN* and *Cas_C* of a random hypergraph. The experiment with match counts and Spearman's rank correlation deduces two groups of consensuses, i.e., {*CC*, *DC*, *BC*} and {*CN*, *Cas_C*}. The findings in the paper pave

Table 2 Detail of hypergraphs

HG1	HG2	HG3	HG4	HG5	HG6	HG7	HG8	HG9	HG10
$n = 50 m = 8$	$n = 100 m = 23$	$n = 150 m = 38$	$n = 200 m = 40$	$n = 40 m = 10$	$n = 70 m = 25$	$n = 90 m = 34$	$n = 110 m = 25$	$n = 500 m = 87$	$n = 1000 m = 140$

**Fig. 2** Prominent consensus**Fig. 3** Spearman's rho for the significant consensus**Table 3** Average correlation of 10 hypergraphs

Consensus	Avg. of Spearman's rho	Avg. fraction of matching
CC-DC	0.989	0.997
DC-BC	0.941	0.887
BC-CC	0.94	0.887
CN-Cas_C	0.909	0.775

a path to the approximation of global measures like *CC*, *BC* and *Cas_C* using local and semi-global measures like *DC* and *CN*, respectively. In the future, this work can be extended to different types of networks.

References

1. Babu, K.S., Jena, S.K., Hota, J., Moharana, B.: Anonymizing social networks: a generalization approach. *Comput. Electr. Eng.* **39**(2013), 1947–1961 (2013)
2. Bonacich, P., Holdren, A.C., Johnston, M.: Hyper-edges and multidimensional centrality. *Social Netw.* **26**(2004), 189–203 (2004)
3. Busseniers, E.: General centrality in a hypergraph. [arXiv:1403.5162](https://arxiv.org/abs/1403.5162) (2014)
4. D'Angelo, G., Severini, L., Velaj, Y.: Influence maximization in the independent cascade model. In: *ICTCS 2016*, Proceedings of the 17th Italian Conference on Theoretical Computer Science, pp. 269–274 (2016)

5. Faust, K.: Centrality in affiliation networks. *Social Netw.* **19**(1997), 157–191 (1997)
6. Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)
7. McCulloh, I.: Network topology effects on correlation between centrality measures. *Connections* **30**(1), 21–28 (2010)
8. Roy, S., Ravindran, B.: Measuring Network Centrality Using Hypergraphs, CoDS’15, pp. 59–68 (2015)
9. Vogiatzis, D. Influence Study on Hyper-Graphs. Social Networks and Social Contagion. AAAI Technical Report FS-13-05, pp. 24–29 (2013)
10. Zhao, Q., Lu, H., Gan, Z., Ma, X.: A K-shell decomposition based algorithm for influence maximization. In: Cimiano, P., et al. (eds.) Engineering the Web in the Big Data -Era. ICWE 2015. LNCS, vol. 9114 (2015)

Amplifying the Polarity Categorization on Twitter Data Using Tweet Polarizer Algorithm and Emoticons Score



D. N. V. S. L. S. Indira and J. N. V. R. Swarup Kumar

Abstract Mining is utilized to help individuals to separate important data from huge amount of information. Opinion Mining or Sentiment Analysis concentrates on the exploration and grasp of the feelings from the content generated in social media. It recognizes the supposition or attitude that an individual has towards a point or an article and it looks to distinguish the perspective hidden in the large content range. Knowing clients' sentiments and giving the best arrangement or administration is an outstanding business sector procedure pursued by each business. In this paper, we center around producing polarities of tweets to know the general feeling on a given word or a trump card string. An algorithm Tweet Polarizer is used in this paper to categorize the tweets. A couple of NLP procedures are used to develop a superior methodology for making the most appropriate and possible fringe for a given tweet and to imagine the few trademark features of customers like from which area he has posted the tweet and when.

Keywords Natural language processing (NLP) · Kibana · Elastic search · Opinion mining · Tweet polarizer · Emoticons

1 Introduction

In recent years, online life industry has experienced mind-blowing improvement in scoring end examination. The emotions or feelings posted by the user to user or individual to group vary from one to other on brand or any industry for evaluating. They are used to give ratings for items by considering different things on products and users. These ratings are noteworthy for any firm to build a grip on how people consider their things. This paper suggests an estimation of tweets posted continuously

D. N. V. S. L. S. Indira (✉) · J. N. V. R. S. Kumar
Gudlavalleru Engineering College, Gudlavalleru, AP, India
e-mail: indiragamini@gmail.com

J. N. V. R. S. Kumar
e-mail: swarupjnvr@gecgudlavalleru.ac.in

for a given hashtag [1, 7]. Spark streaming is used to collect twitter data to process a tremendous amount. HDFS (Hadoop Distributed File System) [2] is used to take care of data spouted from twitter. NLP Techniques like stemming, stop word removal, extracting abbreviations are used to mine the extracted tweets for the furthest point in Elasticsearch and store those data in No-SQL databases. Finally, the polarities of tweets are portrayed by the proposed algorithm. Kibana is used to envision most peripatetic tweets with the customer information and qualities of the customer.

One of the most famous microblogging web site is Twitter. In this paper, the significant jobs are played by NLP procedures, Kibana Tool, Elastic Search [4, 7, 8]. Before, NLP methods broke down just three slants (positive, unbiased and negative). In any case, presently, an aggregate of five slants are thought about. This range among the positions 2 is for very good, 1 is for good, 0 is for neutral, -1 is for bad, -2 is for very bad.

In Elasticsearch, Kibana is an open-source information perception module. It gives perception abilities on top of the substance well-organized on an Elasticsearch group. Users can draw line, bar, pie and dissipate plots and maps over huge volumes of information. Elasticsearch is a web index dependent on Lucene. It gives a circulated, multitenant-fit full-content pursuit motor with an HTTP web interface and schema-free JSON archives [4].

Major offerings of the proposed work are

- Collecting real-time tweets from microblogging web site, i.e. from Twitter.
- For a given #tag calculates the polarity and visualizes the sentiment using the Kibana tool.
- Showing feelings about genre further than polarity.
- Characteristics of the user are visualized in the latest ecosystem of elastic search.

2 Existing Work

Sentimental analysis likewise called assessment mining is very valuable in internet-based monitoring system as it enables us to increase a figure of the more extensive popular sentiment behind numerous brands and open points. For recent years, social media industry has encountered inconceivable amazing development in scoring supposition examination [7, 9]. Estimating supposition about a brand or an industry or the conclusions cited by individuals in a different person to person communication locales is significant for any firm to increase hold of how individuals consider their items, which contributes in a general rating. Subsequently, microblogging sites like twitter are ending up progressively mainstream among clients.

Information is very important and valuable for researchers, processes, methods, and organizations [11]. It is the key component for any decision-making process. Proper information is sometimes as valuable as gold. For example, proper information and public opinion, about the stock market, may save investors from millions of dollars of bad investments. Information about weather forecasting may lead to the safe landing of aircraft or ensure smooth sea travels. Sentiment is an attitude, feeling,

emotion, or opinion toward something. Sentiment can be grasped with hearing, sight, touch, smell and taste. Natural language can be churned to find better insights using sentiment analysis techniques. In the proposed algorithm, we extracted sentiments in 5 different forms like positive, very positive, neutral, negative and very negative using sentiment analysis techniques. Sentiment analysis can be categorized in the word, sentence and document level [3, 12]. This paper is using “sentence level” sentiment analysis, thereby providing polarity to a given tweet using Natural processing algorithms and other machine learning algorithms.

Having said the importance of sentiment analysis there were difficulties in achieving expected results. The Opinion Mining or Sentiment calculation from short text is tedious than traditional topic-based analysis [1, 8–10]. Available classes in topic-based classification are very high than the number of sentiments, variants or classes in sentiment analysis. The classes in sentiment analysis are categorized into two, i.e. either positive or negative. There are some other multi-valued classes that are there in opinion mining, those are positive, very positive, negative, very negative and neutral. But still, there is less number of classes in sentiment analysis. Identifying classes in sentiment analysis is more difficult than topic-based classification [5]. Because topic-based classification uses keyword search. This search mechanism is not suitable for sentiment analysis, the reason that the text contains emoticons, hashtags, and different slangs. Recommending twitter users on any class can be made based on content provided in different tweets using any recommender algorithm.

The following are other reasons for obscurity:

- Sentiments can be expressed by external negative words with little words.
- It is not as much as easy to conclude whether the given text is positive or negative and subjective or objective.
- It is a very complicated way to identify the actual opinion of the user based on the small text which contains stemming words, different slangs, and inconsistent words. Lack of predefined models to process or analyze small text for sentiment calculation is also one of the reasons.
- This paper mainly focuses on helping the end-user by a novel algorithm to decide about a product or service depending on the existing tweets over a period. The reason we considered twitter is because.
- It is a Social Network available as Open Access to all the users.
- It contains marine of Sentiments.
- The text is limited to 140 characters and with huge compactness.
- Twitter provides ease use of GUI to enter sentences and to my opinion in the dynamic and real environment.

3 Proposed Methodology

Tweet Polarizer, designed a novel algorithm to calculate people’s emotions and envisage the feelings of the user in an effective manner on twitter data. The algorithm has the following steps.

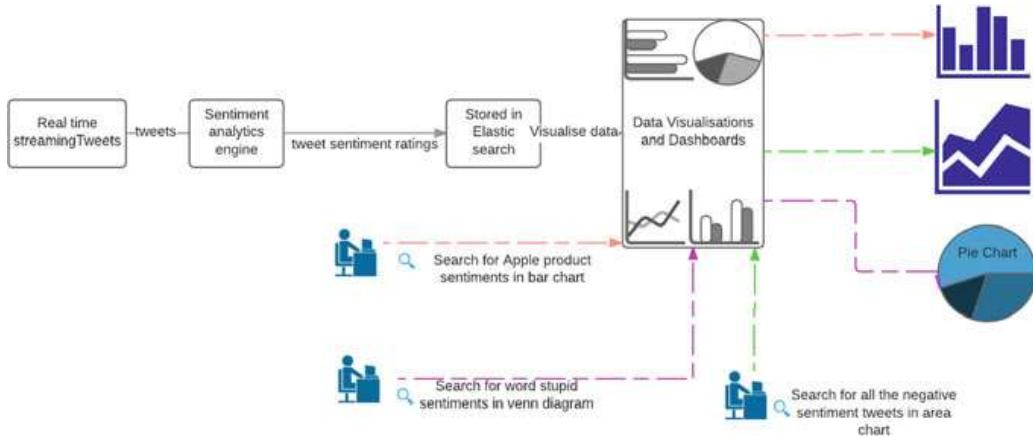


Fig. 1 Block diagram of unstructured text data analysis

1. Extraction of Tweets from Twitter API to Hadoop Distributed File System [1, 8].
2. Pre-Processing of Twitter data and calculating a score using emoticons [3, 6].
3. Sentiment Classification using Tweet Polarizer [6, 8, 9].
4. Search, Ingestion and Visualization of preprocessed tweets and sentiment in ELK stack [2, 8] (Fig. 1).

3.1 Extraction of Tweets from Twitter API to Hadoop Distributed File System

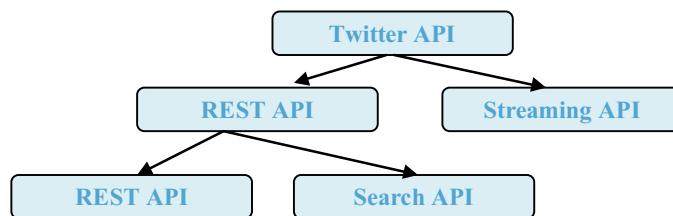
Information gathering for the examination is not as straightforward as it might appear at first idea. There are presumptions and choices to be made. There are three contrastingly gathered datasets: test information, subjective preparing information, and target (nonpartisan) preparing information. Before talking about them, Twitter API will be examined [5] (Fig. 2).

Twitter API: Twitter provides two APIs: REST and Streaming. REST API consists of two APIs: one just called the REST API and another called Search API (whose difference is entirely due to their history of development).

- #Tag based Tweets are received and stored in first file.
- Tweet Content and Trendy #tags in another file.

The above 2 files are used by the next phage to mine the necessary information.

Fig. 2 Categories of Twitter API



Code Snippet for identifying tweets for a given hashtag

```
// segregate Hash tags on a given tweet
Val hTags = stream.flatMap(status => status.getText.split(" ")).filter(_.startsWith("#"))

// popular hashtags in last 120 second window
valt120 = hTags.map((_, 1)).reduceByKeyAndWindow(_ + _, Seconds(120))
    .map{case (tc, ct) => (ct, tc)}
    .transform(_.sortByKey(false))

// popular hashtags in last 10 second window
valt60 = hTags.map((_, 1)).reduceByKeyAndWindow(_ + _, Seconds(60))
    .map{case (tc, ct) => (ct, tc)}
    .transform(_.sortByKey(false))

// Popularhashtags
T120.foreachRDD(rdd => {
    Val tList = rdd.take(10)
    println("\nPopular topics in last 60 seconds (%s total):".format(rdd.count()))
    tList.foreach{case (ct, tag) =>println("%s (%s tweets)".format(tag, ct))}}
```

3.2 Pre-processing of Twitter Data and Calculating a Score Using Emoticons

This section of the module contains the processing of all the tweets that are collected in the data collection phase. Data Pre-processing for short text is classified into five steps.

- Extracting Abbreviation
- Replacement of Slangs with actual word
- Stemming
- Spell Check
- Removal of Stop Words

Example:

The Output format is: (Cleansed Tweet, Emoticons (if any), Hashtags)

Tweet 1. @world1consult: Trump is pushing ‘Buy American.’ But do customers care? /hlfiJVEfpA @CNNMoney @NoNetzSuits #china #Customer

OUTPUT: (@world1consult Trump Buy American customers care @CNNMoney @NoNetzSuits #china #Customer)

Tweet 2. A Rs. 100 during demonetization have everyone a #100WaliHappiness

OUTPUT: (Rupees 100 demonetization #100WaliHappiness).

3.3 Sentiment Classification Using Tweet Polarizer

In this module, emoticon scoring is calculating and add the score to score obtained from the tweet polarizer. The following steps are using to calculate the polarity or sentiment or opinion of a tweet.

- Tweet Scoring based on emoticons present in the tweet
- Scoring of Preprocessed Cleansed Tweet
- Train Preprocessed Twitter Data
- Algorithm Tweet Polarizer
- Rating Tweet based on sentiment.

3.3.1 Algorithm Tweet Polarizer

Tweet Polarizer algorithm is used to opinion that can be is proposed to calculate the score of a tweet. After execution of this algorithm each tweet has a score varies from -2 to 2 (Tables 1 and 2).

Table 1 Gives the top 8 tweeted users with different polarities

Tweet author	Very negative (-2)	Negative (-1)	Neutral (0)	Positive (1)	Very positive (2)
Missxmarisa	58	380	700	380	700
Chineseleam	-	-	1000	440	380
Midesfilenegro	58	230	760	235	210
Erkagarcia	-	620	58	380	380
Tsarnick	-	180	200	460	600
Lost_dog	-	1100	300	-	-
Dogbook	-	600	500	50	-
Linnetwoods	-	600	300	48	200

Table 2 No of tweets in each polarity calculated by using tweet polarizer algorithm collected in one hour

Different polarities	Number of tweets in each polarity
Very negative	1322
Negative	8565
Neutral	16,329
Positive	4722
Very positive	9049

Algorithm : Tweet Polarizer(N,T)

// Let us assume: **Inputs:** N: Number of preprocessed tweets; T: An array of preprocessed tweets

Output:

W_i: Collection of words those are tokenized in ith tweet, which is obtained from Stanford NLP tokenizer.

F_i: Attribute List of ith tweet, from Classifier.

NR: Numerical value of rating the tweet obtained from the Tweet Polarizer algorithm.

Assumed Variables:

JJ1: Adjective in tweet; JJR1: Adjective relative in tweet

JJS1: Adjective excellent; ADJ: JJ1/JJR1/JJS1

VNEG1: Very Negative; VPOS1: Very Positive; NEG1: Negative; POS1: Positive

N1: Neutral; Score: output from MaxEnt _Classifier (POS1/NEG1/N1)

R: rating of the tweet calculated by comparing Score and ADJ

Algorithm:

for i=1 to N in steps of 1 do

 W_i= Stanford NLP tokeniser(T_i).

 F_i= $TF*IDF (W_i)$

 for j=1 to F_i in steps of 1 do

 Score= MaxEnt _Classifier (F_j)

 end for.

//Apply Parts Of Speech tagging on T_i, to recover JJ1, JJR1, JJS1.

 ADJ=POS (T_i)

//if score is N1, do not consider ADJ

 if(score==N1)

 R=score

 end if

//if score is NEG1 and if the sentence contains JJ1, JJR1, JJS1 mark it as VNEG1

 if(score== NEG1&&ADJ) R= VNEG1

 else R= NEG1

 end if

//if score is POS1 and if the sentence have JJ1, JJR1, JJS1 mark it as VPOS1

 if(score== POS1&&ADJ) R= VPOS1

 else R= POS1

 end if

//Transfer Quality Ratings to numeric values: VPOS1 to 2, POS1 to 1, N1 to 0, NEG1 to -1,VNEG1 to -2

 if(R== VPOS1) NR=-2

 else if(R== POS1) NR=1

 else if(R== N1) NR=0

 else if(R== NEG1) NR=-1

 else NR=-2

 end for

3.4 Search, Ingestion and Visualization of Preprocessed Tweets and Sentiment in ELK Stack

At first, we considered alternatives like Hive, HBase which are biological systems of Hadoop, however, we discovered downsides as a business investigator may not discover ease in utilizing these. One can even make an offer dynamic dashboards to changes in inquiries constant. Thus, we made utilization of Elasticsearch and its stack specifically Logstash and Kibana [45, 62]. There has been an examination crevice that has been distinguished in a change of information from Relational to No-SQL Document-Oriented Databases.

In this section, we used Kibana, a visualization tool to summarize the tweets, its polarity. We have also captured geo-coordinates along with the tweets. There are several different visualization types, including area chart, data table, line chart, markdown widget, metric, pie chart, tilemap, and vertical bar chart.

4 Experimental Results

This area chart gives a stratagem of total number of tweets versus sentiments (Figs. 3 and 4).

Pie Chart: Inner Pie represents different sentiments; each slice signifies the count of each polarity from the dataset. Outer Pie represents top five users belonging to a given polarity value.

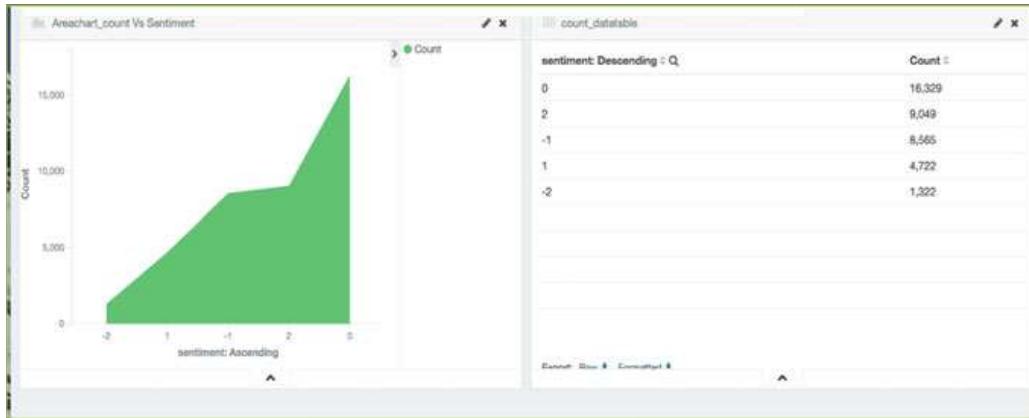


Fig. 3 Plots the polarity in the form of area versus count of tweets

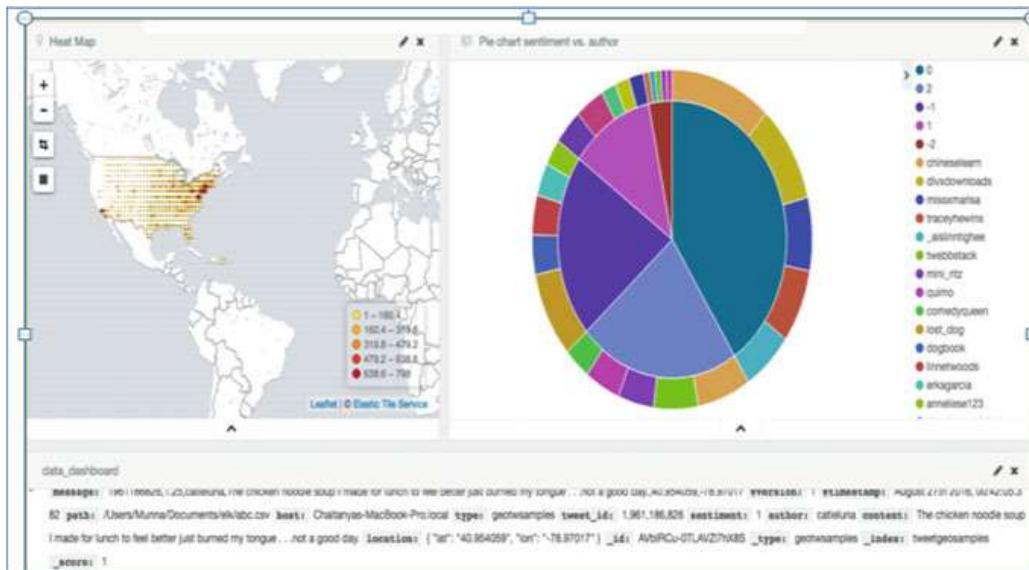


Fig. 4 Dashboard consisting of Heat map, Pie Chart and Dataset

5 Conclusion Future Work

Microblogging has turned out to be the biggest method of correspondence. Information present in these locales encourages part of the business to produce open surveys from the correspondences in internet-based life. In this paper we depicted calculation to decide and escalate notion by figuring extremity from the content information (tweets), principle center are being the most recent innovations through which we can envision and store the information for a simple outline. We centered to depict distinctive perception to abridge the information for various inquiry criteria given by clients. For a similar explanation, we have executed a versatile quest for ordering archives and associated with Kibana to give consistent association. Extremity grouping is likewise far increasingly supportive in business improvement. Notwithstanding these, we can sift through individuals with a great measure of learning, break down and classify their abilities for various types of occupation jobs and propose them with the best business openings. A wide execution of this paper should be possible and the procedures referenced can be loosened up even to other web-based life like Facebook, LinkedIn, and so on.

References

1. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of LREC 2010
2. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: USENIX Conference on Hot Topics in Cloud Computing (2010)

3. Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F., Kaymak, U.: Exploiting emoticons in polarity classification of text. *J. Web Eng.* (2013)
4. Divya, M.S., Goyal, S.K.: Elastic search: an advanced and quick search technique to handle voluminous data. In: *Int. J. Adv. Comput. Technol. (COMPUSOFT)* **2**(6) (2013)
5. Madhoushi, Z., Hamdan, A.R., Zainudin, S.: Sentiment analysis techniques in recent works. In: *Proceeding in Science and Information Conference 2015*
6. Himeno, S., Aono, M.: Tweet polarity classification focused on positive and negative term frequency ratio. In: *IEEE, 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, 16–18 Aug 2017
7. Bhanap, S., Kawthekar, S.: Twitter sentiment polarity classification & feature extraction. *IOSR J. Comput. Eng. (IOSR-JCE)* 1–3. e-ISSN 2278–0661, p-ISSN 2278-8727
8. Kalyani, D., Mehta, D.: Paper on searching and indexing using elasticsearch. *Int. J. Eng. Comput. Sci.* **6**(6), 21824–21829 (2017). ISSN:2319-7242
9. Venkatesan, N.J., Nam, C.S., Kim, E., Shin, D.R.: Analysis of real-time data with spark streaming. *J. Adv. Technol. Eng. Res.* **3**(4), 108–116 (2017)
10. Raghuwanshi, A.S., Pawar, S.K.: Polarity classification of Twitter data using sentiment analysis. *Int. J. Recent Innov. Trends Comput. Commun.* **5**(6). ISSN 2321–8169
11. Mäntylä, M.V., Graziotin, D., Kuutila, M.: The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Comput. Sci. Rev.* **27**, 16–32 (2018)
12. Indira, D.N.V.S.L.S., Kiran Kumar, R., Prasad, G.V.S.N.R.V., Usha Rani, R.: Detection and classification of trendy topics for recommendation based on Twitter Data on different genre. In: *International Conference on Smart Intelligent Computing and Applications*, vol. 105, pp. 143–153. Springer, Belin, 5 Nov 2018

Classification of Fashion Images Using Transfer Learning



Raji S. Pillai and K. Sreekumar

Abstract Computer vision, a specialization of machine learning aims to train machines to learn and interpret visual content based on its features. Many CNN based architectural models are developed to classify fashion images. The basic accuracy loss or misclassification occurs in similar fashion items. The reason for this is the minute features may not be extracted efficiently by existing models. Therefore, in the proposed study we utilize the feature concatenation property, knowledge consolidation and inductive feature transfer of DenseNet architecture for categorization of fashion MNIST dataset. The performance analysis is done by generating a confusion matrix. Further, we compare the performance of proposed architecture with the state of the art image classification architectures in literature. The analysis indicates that our proposed system outperforms the existing architectures.

Keywords Computer vision · Transfer learning · FashionMnist dataset · DenseNet · Convolution neural network · Image classification

1 Introduction

Image classification is the process of detecting or identifying the visual content of an image. Based on the result of this identification the image can be classified which is a very trivial task for humans. But the robust or accurate image classification by machine is currently a prominent research area in computer vision. Some of the reasons for accuracy lacking in image classification tasks are similar classes of apparel or accessories are misclassified into wrong classes. For example, the apparel shirt and pullover are often misclassified. The reason for this may the minute features

R. S. Pillai (✉) · K. Sreekumar
Department of Computer Science and IT, Amrita Vishwa Vidyapeetham,
Kochi Campus, Kochi, India
e-mail: raji.janin@gmail.com

K. Sreekumar
e-mail: sreekumar4@gmail.com

not be captured or while extracting features through the layers, many of the features are failed to hit the final classifier layer. So in our study, we are using the feature concatenation property of densely connected convolution neural network for feature extraction and a fully connected layer as a classifier, then it is compared with a convolution neural network with fine-tuning.

2 Background and Related Work

Classification of visual content in an image serves as a building block for many applications. A CNN based architecture with two layers of CNN, combined with batch normalization and skip connections were introduced by Bhatnagar et al. [1] for effective image classification. In a research work proposed by Kitsuchart et al. [2] suggested that CNN with dropout shows better accuracy with image augmentation and regularization techniques.

As training a CNN architecture requires an enormous amount of data and a substantially huge amount of Graphical Processing Unit, training CNN for a small dataset is a troublesome work. In a research work presented by Eshwar et al. [3] they employed transfer learning technology for the effective classification of apparel images. They retrained the final layers of GoogLeNet for classifying online fashion images. Yosinski et al. [4] studied and presented a research paper on how the features are being transferred in deep neural networks. From their work, it is proved that performance generalization can be improved by initializing the feature transferring. A comparative study of deep transfer learning strategies with a pathology dataset is presented by Mormont et al. [5].

3 Proposed Methodology

Image Classification has many major applications in every field of computer vision. In the proposed work, we have fine-tuned the network parameters of the DenseNet. The dataset we have used is a benchmarked dataset FashionMnist which is developed by Zalando research Institute. In CNN architecture feature sets are transferred from lower level layers to the layers nearer to the classifier. The first-level layers identify general features of the image and layers nearer to classifier identifies the specific features. Our main focus is to exploit the feature reuse property of the DenseNet for the effective classification of a small dataset by applying transfer learning methodology.

3.1 Densely Connected Convolution Network

D-CNN is a new CNN architecture which introduces dense connectivity between layers. In this architecture, the output of one layer becomes the input to all succeeding layers. The input from all the previous layers is combined using a channel-wise operation called concatenation. Since all the layers have direct interaction with all its previous layers, the layers are much thinner. Each layer generates k feature maps which are called growth rate. In D-CNN the features extracted at each layer are concatenated in the next layer. The feature map of each layer is of the same size. Since all the layers are connected there is a strong gradient flow, which implies implicit deep supervision.

In D-CNN number of parameters is less compared to CNN [6]. In D-CNN each layer consists of diversified features and it concatenates features from all its previous layers. So features of all complexity levels are fed into the classifier. Since the D-CNN has compact internal layers and less redundant feature transfer, it is considered as a good feature extractor for various CNN related computer vision tasks [6].

The convolution function applied between a two-dimensional image I and a two-dimensional kernel K is represented in Eq. (1),

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, (j - n)) K(m, n) \quad (1)$$

In every traditional convolution feed-forward neural network, the output of the i th layer becomes the input for $(i + 1)$ th layer. The layer transition is represented as,

$$X_i = H_{i(X_{i-1})} + X_{i-1} \quad (2)$$

3.2 Transfer Learning

Deep Learning algorithms perform remarkably well in classifying images. But with the problem with this type of architecture is that it requires a large amount of data, computational power and also it is time-consuming. If a network is trained with sufficient data and if it is a proven architecture, the same architecture training can be used for another application. This can be achieved by using a technique called transfer learning.

Application of transfer learning techniques depends mainly on the following four scenarios:

- Case 1: The test dataset is *small* and *similar* to the dataset which is used for training.
- Case 2: The test dataset is *large* and *similar* to the dataset which is used for training.

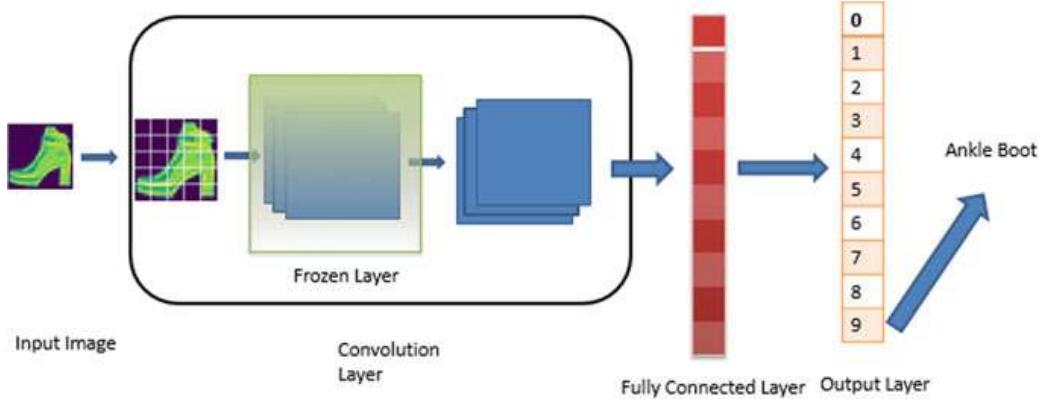


Fig. 1 Feature extraction from pretrained network by proposed methodology

- Case 3: The test dataset are **small** and **different** from the dataset which is used for training.
- Case 4: The test dataset is **large** and **different** from the dataset which is used for training.

For this FashionImage classification, the proposed architecture is based on DenseNet, which is trained with ImageNet dataset which consists of 1.2 million images for training, and 50,000 for validation, and 1000 classes. The Fashion MNIST Dataset is a small dataset with only 70,000 images. So our work comes under Case 3. Thus, we are freezing the lower level layers, that is the layers nearer to the convnets, and a dense layer is being connected to the retained pretrained layers. Then the weights of the new fully connected layers are randomized, trained and updated.

In our proposed work, for fine-tuning, the final layers are replaced by a fully connected layer with 10 neurons as our target dataset contains 10 classes of images. We freeze the initial layers and trained the new appended layers for 10 epochs, and then train the whole network for 50 epochs. The optimizer we used is Adam with a learning rate of 0.01. As loss function, we have used categorical-entropy. The training set is exclusively used for fine-tuning.

After extracting the features, a fully connected single layer is used for training the classifier. Figure 1 depicts the process of feature extraction and image classification using the proposed methodology.

4 Experimental Results

4.1 DataSet

Fashion MNIST is a novel benchmarked dataset that is used for machine learning applications. It was introduced by Han Xiao, Kashif Rasul, Roland Vollgraf as a direct substitute for the MNIST dataset. The Fashion MNIST consists of 60,000 fashion

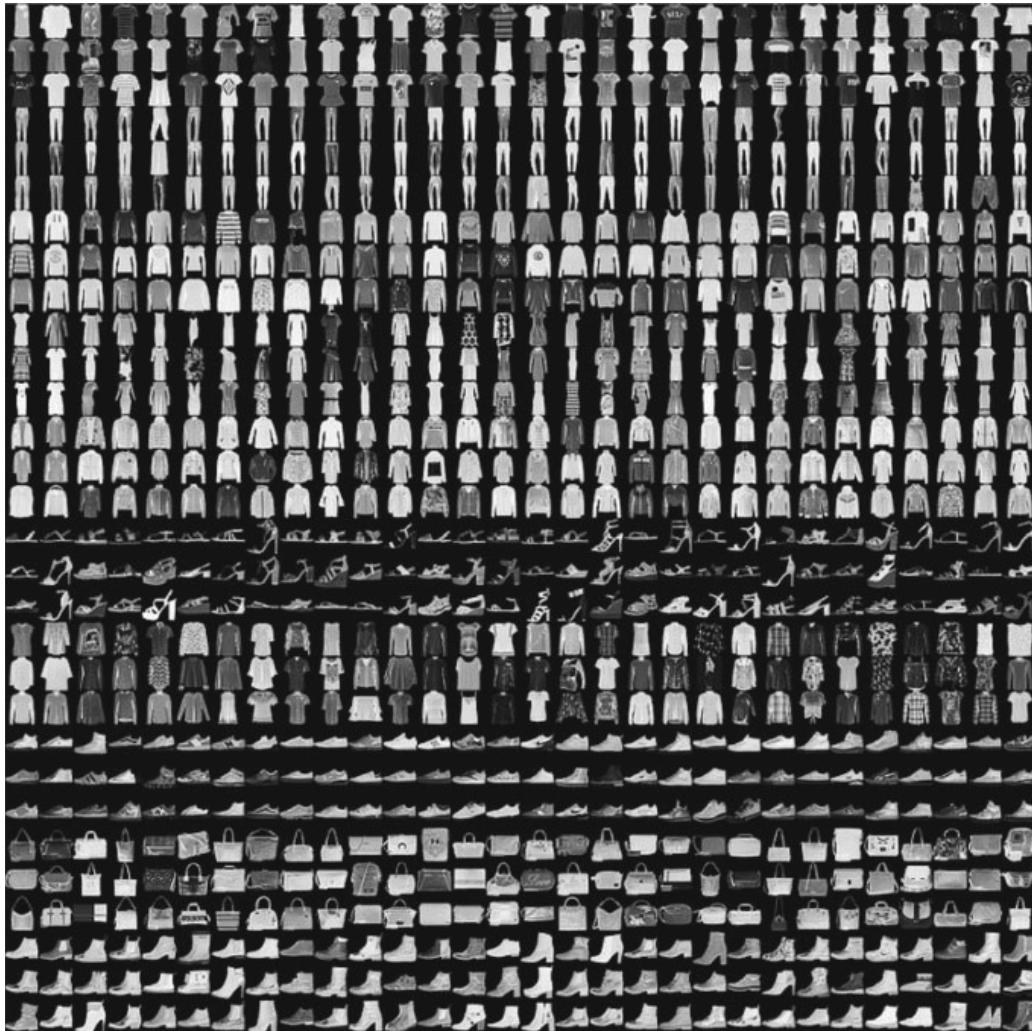


Fig. 2 Fashion MNIST DataSet

images as training set examples and 10,000 fashion images as test set images. Each image is a 28×28 grayscale image. Each image in the dataset is associated with a class out of 10 classes [7]. Figure 2 depicts Fashion MNIST Dataset.

5 Experimental Results and Analysis

The conferred work is implemented in Python. The feature extraction is done with the help of the DenseNet function available in the Keras module of TensorFlow. The experimentation is divided into two phases training and testing phase. During the training phase, the features are extracted from the training images and it is fed into the fully connected layer which is used as the classifier. These extracted features which are stored in the knowledge base of DenseNet are being used by the classifier to

make predictions. The predictions done by the classifier is verified by the test images, and the accurate analysis of the architecture is done by generating a confusion matrix. The proposed DenseNet based architecture achieves 93.95% accuracy on test images (Table 1).

The accuracy analysis of the existing literature is depicted in Table 2.

Figure 3 depicts some of the correctly classified images. All the images belonging to different classes are predicted accurately. Figure 4 depicts some of the wrongly

Table 1 Confusion matrix of DenseNet based architecture

Class	Label	0	1	2	3	4	5	6	7	8	9
T-Shirt/top	0	911	0	9	3	2	0	72	0	3	0
Trouser	1	0	997	0	1	0	0	1	0	1	0
Pullover	2	24	1	909	5	26	0	35	0	0	0
Dress	3	18	6	7	917	17	0	34	0	1	0
Coat	4	1	0	20	9	947	0	23	0	0	0
Sandal	5	0	0	0	0	0	993	0	5	1	1
Shirt	6	91	0	35	8	74	0	789	0	3	0
Sneaker	7	0	0	0	0	0	6	0	985	0	9
Bag	8	1	1	0	2	0	1	0	0	995	0
Ankle Boot	9	0	0	1	0	0	9	0	38	0	952

The number of correctly classified images of each class is given in bold

Table 2 Accuracy analysis of fashion MNIST in literature

Methodology	Accuracy (%)
HOG + SVM	86.53
CNN + SVM	90.72
CNN	92.54
Dense Net	93.95

Fig. 3 Correctly predicted classes

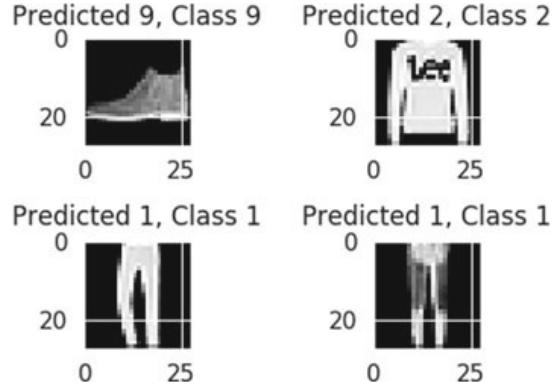


Fig. 4 Wrongly predicted classes

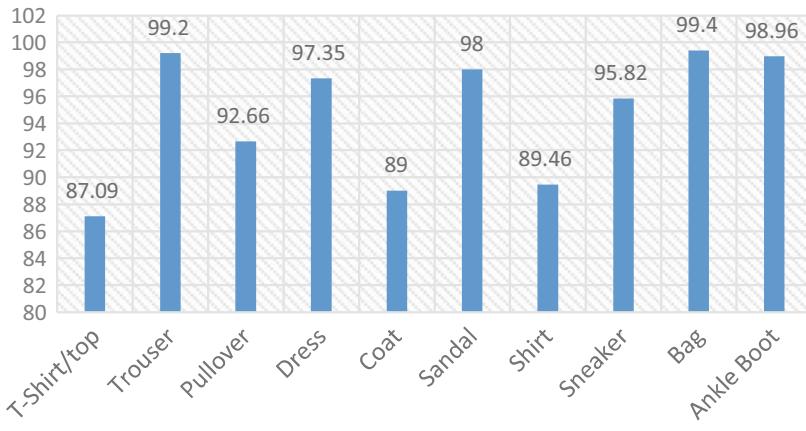
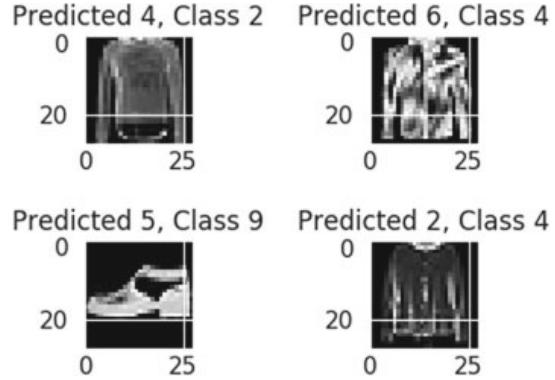


Fig. 5 Accuracy analysis of each class

predicted classes. It is very clear from the results that images with very low dissimilarities are being misclassified.

Figure 5 indicates the accuracy analysis of each class of images.

6 Conclusion

In this paper, we fine-tuned the DenseNet weight parameters over the benchmarked state of the art fashion image dataset FashionMNIST for image classification tasks. We analyzed the result by generating the confusion matrix. Regarding fashion images in the dataset, major misclassification occurs with similar-looking images such as shirt and pullover or ankle boot and sneakers, etc., this arises because of the dropping of features while transferring. In our proposed architecture we exploited the feature concatenation property of DenseNet and the results show that it invariably contributed to the improvement of accuracy. As future work, the same technique can be extended for stratified classification of images.

References

1. Bhatnagar, S., Ghosal, D., Kolekar, M.H.: Classification of fashion article images using convolutional neural networks. In: International Conference on Image Information Processing (ICIIP) (2017)
2. Pasupa, K., Sunhem, W.: A comparison between shallow and deep architecture classifiers on small dataset. In: Information Technology and Electrical Engineering (ICITEE) 2016 8th International Conference, Indonesia (2016)
3. Eshwar, S.G., Gautham Ganesh Prabhu, J., Rishikesh, A.V., Charan N.A.: Apparel classification using convolutional neural networks. In: 2016 International Conference on ICT in Business Industry & Government (ICTBIG) (2016)
4. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, Canada (2014)
5. Mormont, R., Geurts, P., Maree, R.: Comparison of deep transfer learning strategies for digital pathology. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2343–2349 (2018)
6. Huang, G., Liu, Z., van der Maaten, L.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA (2017)
7. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (2017)

A Novel Adaptive Out of Step Protection in Synchronous Generators Using Support Vector Machine Algorithm



R. Hemavathi, I. Limsha Deborah, and M. Geethanjali

Abstract The out of step condition occurs mainly due to the propagation of fault throughout the power system network. The fault may be due to power swings, phase faults, loss of excitation, under or over frequency. Particularly, in synchronous generators, the loss of excitation causes the out of step condition which leads to the wide-area blackout. In this paper, the synchronous generator's frequency, phase and magnitude of voltage are directly observed from the PMUs connected to its respective buses. The responses of synchronous generator for different operating conditions are observed directly from the PMU. This procedure is developed and tested in MATLAB/Simulink software. The PMU data is extracted to classify the loss of excitation(LOE) fault, three-phase faults and the normal operating condition using one of the premier supervised machine learning algorithms called Support Vector Machine algorithm. The propound algorithm is done in MATLAB environment and tested in IEEE 14 bus system. The results project the performance of the algorithm for fault classification in synchronous generators.

Keywords PMU · Synchronous generator · Loss of excitation (LOE) · Three-phase fault · Support vector machine (SVM) · Out of step (OOS) protection

1 Introduction

The wide-area blackout condition occurred in India, China, America, Bangladesh and some other countries consecutively in recent years. This wide-area blackout

R. Hemavathi (✉) · I. L. Deborah · M. Geethanjali
Thiagarajar College of Engineering, Madurai, Tamil Nadu 625015, India
e-mail: hemavathi11@gmail.com

I. L. Deborah
e-mail: deborahlimsha@gmail.com

M. Geethanjali
e-mail: mgeee@tce.edu

condition affected millions of people's normal routine life and work. Large number of factories and industries were shut down which affected them economically. It causes major economic impacts such as food spoilage, manufacturing plant shutdown, traffic congestion due to traffic control system failure, life support issues in hospitals, nursing homes, etc. This severe power blackout is mainly because of the power system network going into out of step condition which is crucial.

The power system network is subjected to many smaller and larger disturbances. Some disturbances may result in power swing which may be stable or unstable. In synchronous generator, the occurrence of unstable power swing may lead to losing the synchronism of the generator from the system. Sometimes loss of excitation in synchronous generators causes the rotor angle to increase and make the generator to run asynchronously. The increase or decrease in load demand also causes frequency deviations which in turn changes the rotor angle and thus turns the generator to lose its synchronism with the system. The loss of synchronism may cause the propagation of fault throughout the power network and this state is said to be out of step condition.

The UEP (Unstable Equilibrium Point) was identified from EAC (Equal Area Criterion) and Least Square curve fitting algorithm was applied for out of step condition detection [1]. The EAC is applicable only for a single machine infinite bus system (SMIB). PMU measurements after the occurrence of fault were taken. ICA (Independent Component Analysis), UKF (Unscented Kalman Filter) and Extended Equal Area Criterion method (EEAC) were used. The out of step condition detection was done by the impedance trajectory method [2]. The impedance trajectory-based out of step relay is complicated and requires many pre-settings. The LOE (Loss of Excitation) condition is detected from the reactive power output as well as the terminal voltage. The stable power swing is detected from FFT (Fast Fourier Transform) coefficients of 3-phase real power [3]. The generator armature current parameter was used for loss of excitation detection. A mathematical approach called SODAC (Second Order Differential Armature Current) was adapted to discriminate among the loss of excitation, short circuit fault and stable power swing [4]. The bus voltages were taken from the PMU and its frequencies are estimated by a Fast frequency calculation algorithm and Coordinate Rotational Digital Computer algorithm (CORDIC) [5]. These methods require a computational analytical procedure which makes the algorithm complex. From the PMUs installed in different locations, direct phase comparison among the installed PMUs helps in detecting out of step condition [6]. The conventional method of impedance trajectory method and its relay settings were tedious processes and complex in behaviour. This method detects out of step conditions after the occurrence of several prescribed number of swings which causes stress in operating conditions [7–9]. The parameters of identifying loss of excitation and frequency deviation need some analytical and computational procedures [10–15]. The relay settings were to be updated once the network parameters and the operating conditions change when disturbance occurs in the system. This method was found to be an easy and simple method. The machine learning algorithm, SVM (Support Vector Machine) was implemented to detect, classify and estimate the fault location in transmission lines [16]. The fault detection from the PMU measurements (Positive and Zero sequence current) by SVM algorithm was later implemented [17]. In this

research work, the synchronous generator's out of step condition was identified by one of the machine learning algorithms called SVM. The data obtained from the PMU of generator bus directly, the SVM algorithm classifies the fault conditions.

2 Basic Concepts

2.1 PMU

PMU stands for Phasor Measurement Unit. It gives the time-tagged synchronous phasor measurements directly. The phase, frequency, rate of change of frequency (ROCOF) along with magnitude can be directly obtained from the PMU device for monitoring, controlling and protecting the system. Two types of PMUs are available such as p-type PMU (protection PMU) and m-type (measurement PMU). The time-synchronized measurements are essential because if the supply from the grid does not meet the consumer demand, frequency discrepancies can impose severe stress on the grid, which is a potential cause for power outages.

2.2 Need for OOS Protection

When the synchronous generator loses its synchronism with system due to unstable power swing, loss of excitation, frequency deviations and other severe disturbances it should be islanded immediately. The loss of synchronism will lead to equipment damage and also the propagation of outage throughout the power network. Due to cascading of fault throughout the system the relays throughout the network operate and cause the wide-area blackout. This condition of power system network is known as out of step condition. The identification of OOS conditions should be simple, accurate and fast. Also, the protection must be adaptive to operate according to the prevailing network condition and quick operation of the relay.

3 Proposed Algorithm for OOS Detection

In this work Support Vector Machine approach has been proposed to detect and discriminate the out of step state in the generator from normal operating conditions. The IEEE 14 bus system was taken as the test system. The PMUs were connected to the test system. The PMU measurements (Frequency, Magnitude and Phase of voltage) of the system were taken for three different operating conditions (Three-phase fault, loss of excitation fault and normal operating condition in the system). The data was extracted for the classification of faults in the form of numerical data.

The Support Vector Machine (SVM) algorithm which is one of the machine learning algorithms was adopted for classification of faults. The IEEE 14 bus system was simulated in MATLAB Simulink software (MATLAB R2019a). SVM coding was done in MATLAB software. The outputs for fault classification were obtained by SVM algorithm. Figure 1 represents the flowchart of methodology.

3.1 Test System

Figure 2 shows the test system considered in this research, the standard IEEE 14 bus system. The bus system includes five generators, five transformers, fourteen buses. The PMU's were connected to all fourteen buses. The nominal frequency of PMU's was selected as 50 Hz with sampling rate 48 points/cycle.

4 Simulation Results and Discussions

The response of the generator connected to the bus 1 after the occurrence of three-phase fault and loss of excitation is directly observed from the PMU connected to that respective bus. The PMU connected to the generator bus gives the magnitude, Frequency, Phase difference of the output voltage of the generator.

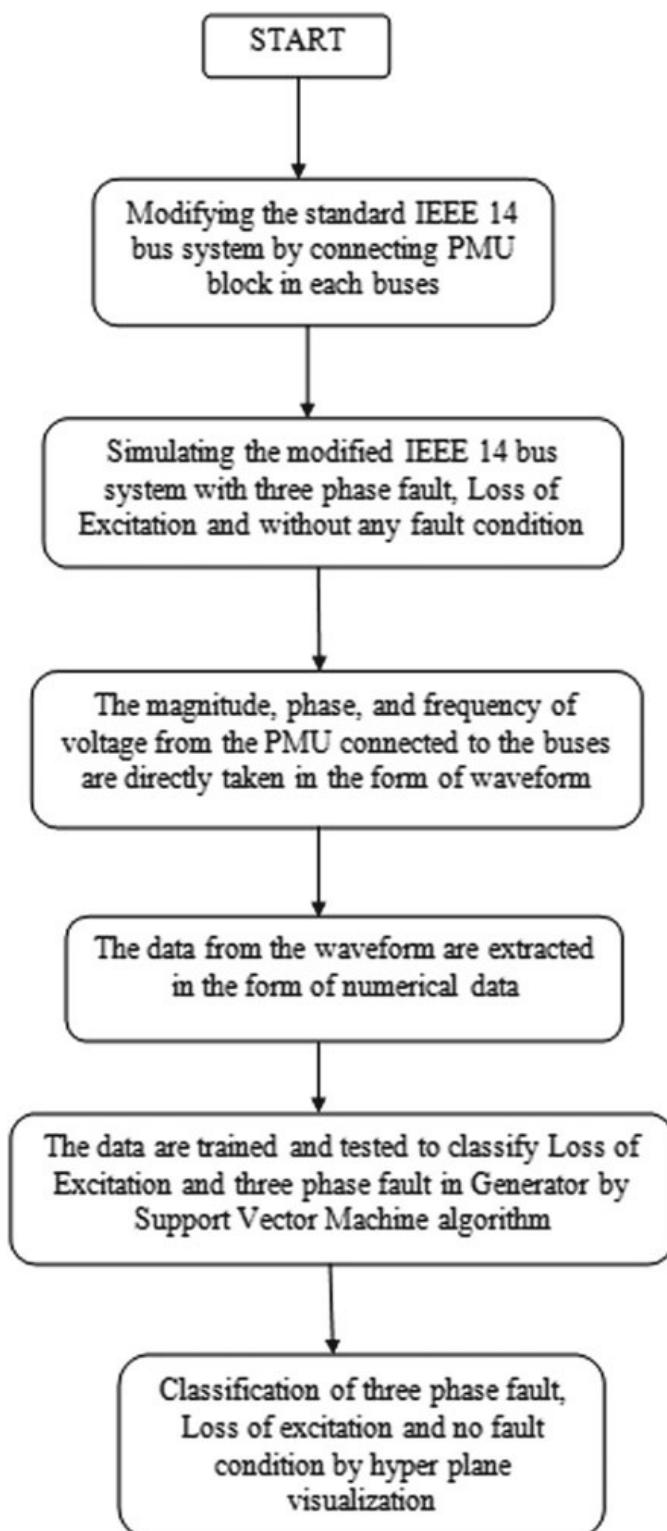
4.1 System Response Without Any Disturbances

Figure 3 shows the response from PMU. The frequency, magnitude and phase of the generator's voltage are without any deviation. The frequency is maintained at its nominal level. Similarly, the magnitude and phase are also maintained at its steady-state without much deviation.

4.2 System Response with Three-Phase Fault

The three-phase fault is given at 2 s and the following response of generator is observed from the PMU connected to bus 1. The frequency shows larger dip, magnitude and phase shows larger deviations as in Fig. 4.

Fig. 1 Flowchart of overall methodology



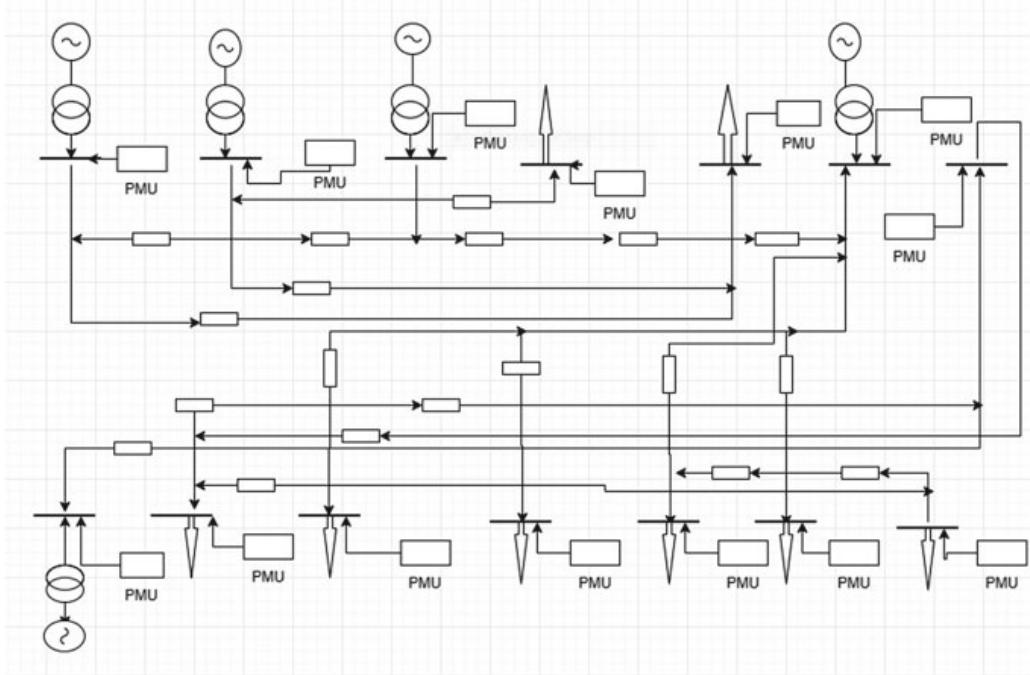


Fig. 2 Standard IEEE 14 bus system modified by connecting PMUs

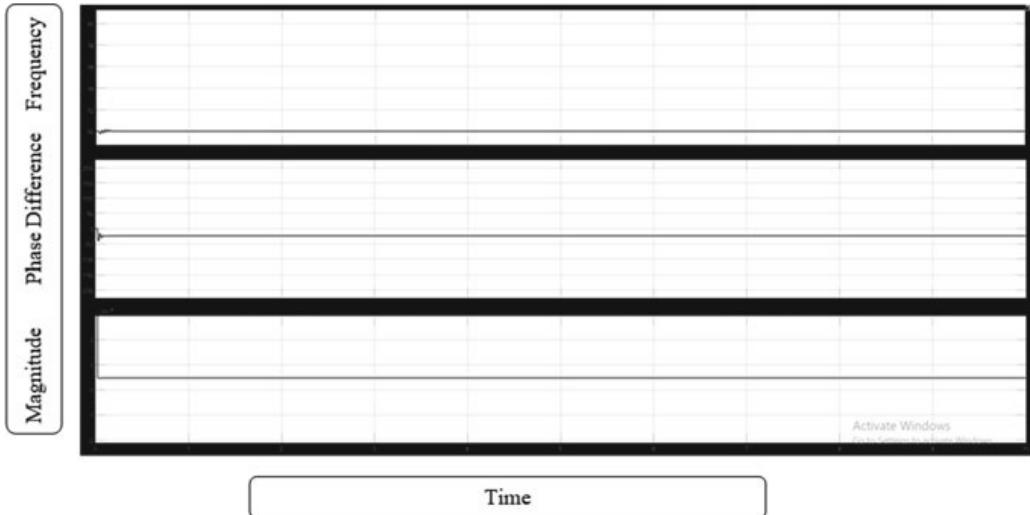


Fig. 3 PMU response of generator connected to the bus 1 in normal operating condition

4.3 Response of the System with Loss of Excitation

The Excitation loss is given in the generator right from the initial condition. Loss of excitation is applied in the generator by giving 0 as the field voltage. The loss of excitation condition leads to the drastic dip in frequency and also larger deviations in the magnitude and phase of the voltage as in Fig. 5.

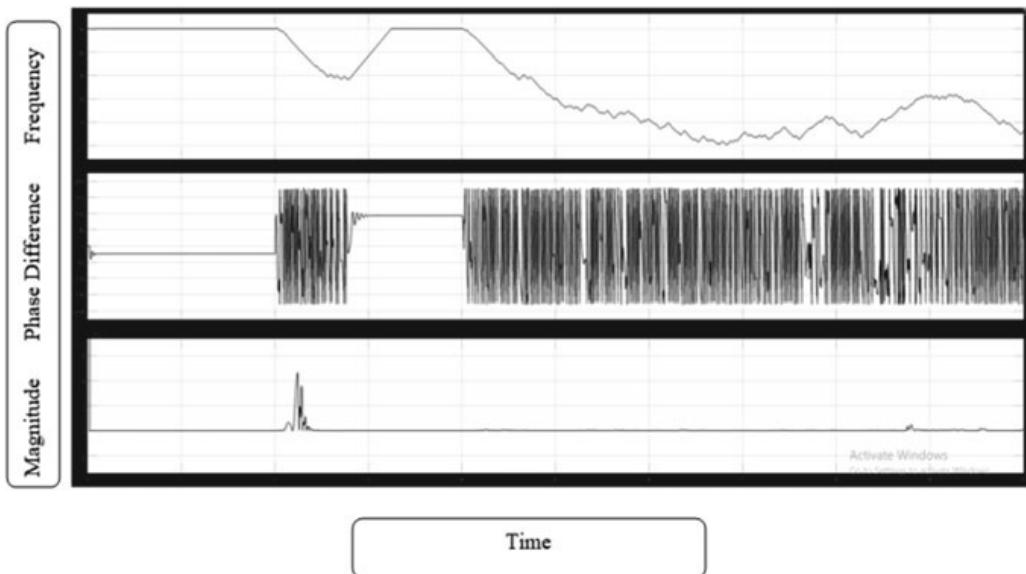


Fig. 4 PMU response of generator connected to the bus 1 for three-phase fault

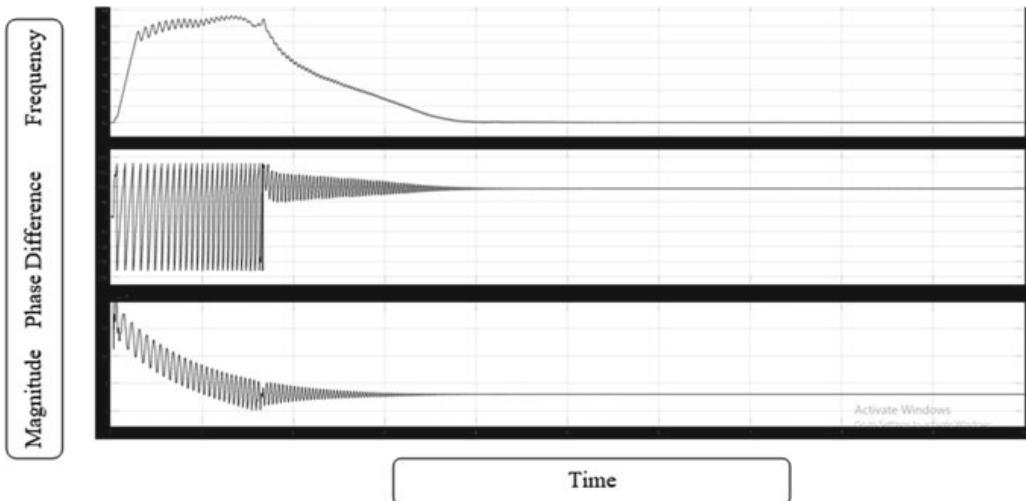


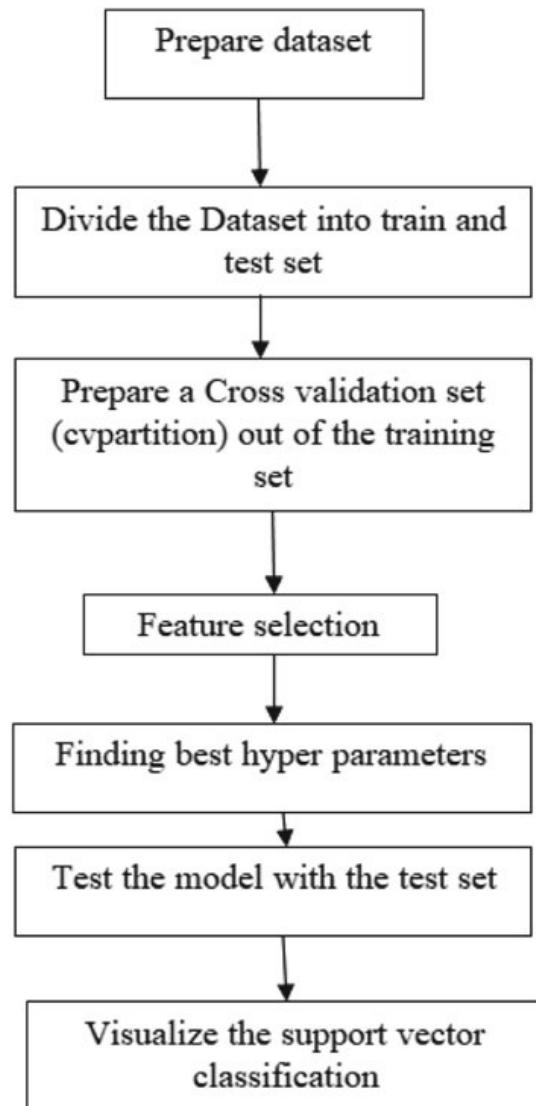
Fig. 5 PMU response of generator connected to the bus 1 for loss of excitation

5 SVM Algorithm Implementation

Support Vector Machine (SVM) is a machine learning technique which comes under supervised learning method. This algorithm maps input to output with reference to the example input–output pairs. Support Vector Machine is of two types, namely classification and regression analysis. Support Vector Machine algorithm makes a decision making optimal hyperplane such that the two-class separation in the data is maximized. The dataset is extracted from the simulation output of MATLAB Simulation software in the form of numerical data. The data extracted from the

simulation output is about 24,000 for LOE and three-phase fault conditions. It is split into training and testing data in the ratio of about 80:20. After successful training and testing of data, it is cross-validated, and the best features are selected. Based on the best test features optimized, the faults are classified. Figure 6 represents the SVM algorithm.

Fig. 6 SVM algorithm



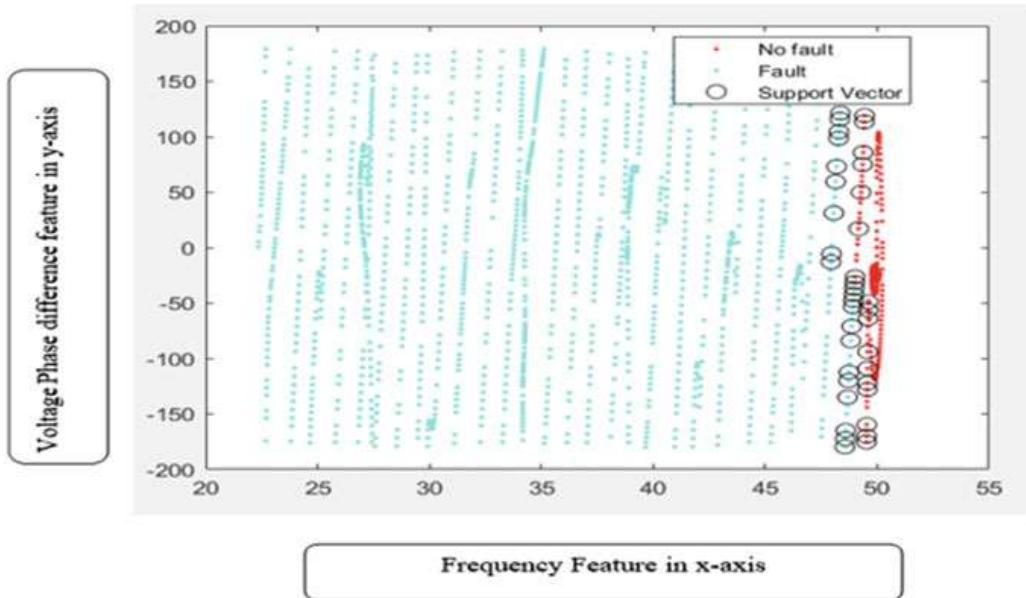


Fig. 7 SVM classification of three-phase fault from no-fault condition

6 Results and Discussions

6.1 Support Vector Machine Algorithm Classification of Three-Phase Fault from No-Fault State

The data is used for classifying three-phase fault from no-fault state. The data is trained and tested by Support Vector Machine algorithm in the MATLAB environment. According to the training of data the test data is classified as given below. The blue colour represents the fault condition and red colour represents the no-fault condition. The support vector points are circled. The accuracy of classification is 99.9% and number of iterations of optimization is 30. Figure 7 shows the SVM classification of three-phase fault from no-fault situation.

6.2 Support Vector Machine Algorithm Classification LOE from No-Fault Condition

The data is used for classifying loss of excitation fault from no-fault condition. The data is trained and tested by Support Vector Machine algorithm in the MATLAB environment. According to the training of data the test data is classified as given below. The blue colour represents the fault condition and red colour represents the no-fault condition. The support vector points are circled. The accuracy of classification

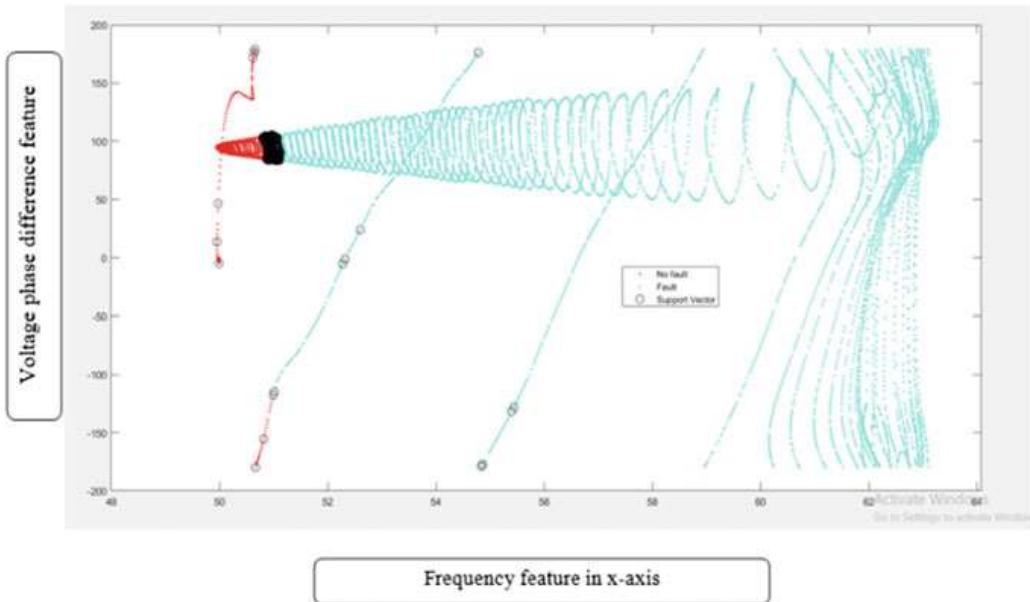


Fig. 8 SVM classification of loss of excitation fault from no-fault condition

is 99.9% and number of iterations of optimization is 30. Figure 8 shows the SVM classification of loss of excitation fault from no-fault condition.

7 Conclusion

The conventional impedance-based out of step relay requires many pre-settings and it also needs many settings to make it adaptive. In this research work, the generator's output voltage is directly taken from PMU and the out of step condition is identified by the Support Vector Machine method which is a machine learning technique. This premier machine learning algorithm is used to train and test the data for the classification of loss of excitation and three-phase fault from normal operating conditions in synchronous generators. This supervised machine learning algorithm helps in identifying the out of step condition immediately and accurately. It trains the network according to the prevailing operating condition such that making the settings adaptive. The results demonstrate that the SVM classification algorithm gives 99.9% accuracy in results. Hence it can be recommended for the practical power systems.

Acknowledgements The authors are grateful to the authorities of Thiagarajar College of Engineering, Madurai-625015, to do this research work. This work was aided by DST-WOS-A fellowship scheme under Ref DST-WOS-A File No: SR/WOS-A/ET-9/2018 (2019-2021).

References

1. Alinezhad, B., Karegar, H.K.: Out-of-step protection based on equal area criterion. *IEEE Trans. Power Syst.* **32**(2), 968–977 (2017)
2. Ariff, M.A.M., Pal, B.C.: Adaptive protection and control in power systems for wide area blackout prevention. *IEEE Tran. Power Delivery* **11**(4), 1815–1825 (2016)
3. Yaghobi, H.: Fast discrimination of stable power swing with synchronous Generator loss of excitation. *IET Gener. Transm. Distrib.* **10**(7), 1682–1690 (2016)
4. Noroozi, N., Yaghobi, H., Alinejad Beromi, Y.: Analytical approach for synchronous generator loss of excitation protection. *IET J.* **11**(9), 2222–2231 (2017)
5. Zhang, S., Zhang, Y.: Characteristics analysis and calculation of frequencies of voltages in out of step oscillation power system and a frequency based out of step protection. *IEEE Trans. Power Syst.* **34**(1), 205–214 (2019)
6. Regulski, P., Rebizant, W., Kereit, M., Hersmann, H.-J.: PMU based Generator out of Step Protection, vol. 51, issue no 28, pp. 79–84. IFAC-paperOnline, Elsevier (2018)
7. Kiaei, I., Lotfifard, S., Bose, A.: Secure LOE detection method for synchronous generators during power swing conditions. *IEEE Trans. Energy Conv.* **33**(4), 1907–1916 (2018)
8. Ostojic, M., Djuric, M.B.: Out of step protection of synchronous generators based on a digital phase comparison in the time domain. *IET Gener. Transm. Distrib.* **12**(7), 873–879 (2018)
9. Mahamedi, B., Zhu, J.G., Hashemi, S.M.: A setting free approach to detecting LOE in synchronous generator. *IEEE Tran. Power Delivery* **31**(5), 2270–2278 (2016)
10. Pandey, P., Sharma, S.K.: Detecting power grid synchronism failure on sensing bad voltage or frequency. *Int. J. Sci. Res. Dev.* **6**(1), 1170–1182 (2017)
11. Fei, T., Jian, Y., Qingfen, L., Yifei, W., Jun, J.: Out of step oscillation splitting criterion based on bus voltage frequency. *J. Mod. Power Syst. Clean Energy* **3**(3), 341–352 (2015)
12. Mojdeh, A.-K., Vittal, Vi: Modeling protection systems in time-domain simulations: a new method to detect mis-operating relays for unstable power swings. *IEEE Trans. Power Syst.* **32**(4), 2790–2798 (2017)
13. Farantatos, E., Huang, R., Cokkinides, G.J., Meliopoulos, A.P.: A predictive generator out-of-step protection and transient stability monitoring scheme enabled by a distributed dynamic state estimator. *IEEE Trans. Power Delivery* **31**(4), 1826–1835 (2016)
14. Abedini, M., Davarpanah, M., Sanaye-Pasand, M., Hashemi, S.M., Iravani, R.: Generator out-of-step prediction based on faster-than-real-time analysis: concepts and applications. *IEEE Trans. Power Syst.* **33**(4), 4563–4573 (2018)
15. Yaghobi, H.: Out of step protection of generator using analysis of angular velocity and acceleration data measured from magnetic flux. *Electr. Power Syst. Res.* **132**, 9–21 (2016)
16. Ray, P., Mishra, D.P.: Support vector machine based fault classification and location of a long transmission line. *Eng. Sci. Technol. Int. J.* **19**(3), 1368–1380 (2019)
17. Jain, A., Archana, T.C., Sahoo, M.B.K.: A methodology for fault detection and classification using PMU measurements. In: 20th National Power Systems Conference (NPSC) (2018)

Sentiment Analysis of Movie Reviews Using Support Vector Machine Classifier with Linear Kernel Function



A. Sheik Abdullah, K. Akash, J. ShaminThres, and S. Selvakumar

Abstract Sentiment analysis refers to the process of determining the opinion stated by the user corresponds to positive, and to be negative or considered to be neutral. The mechanism of sentiment analysis is said to be the process of opinion mining which in turn resembles the behavior/attitude measurement of the speaker. This is extremely useful in a place which there is a complete need for a recommendation for the user to follow a specific case of action. In public domain, the aspect of sentiment analysis is helpful for the user to state a specific nature of the action. This research work focuses on the analysis of review data to determine the aspect based on sentiments using TF, IDF, and SVM. The model extracts the textual reviews and classifies them into positive, negative, and neutral cases. The result retrieved with the proposed scheme gives an improved accuracy of about 87.56% determining the positive and negative cases more efficiently. With this proposed approach the classification of review data can be made more efficiently for various sort of recommendation systems which makes the user have good insight for a product review, movie review, and user rating analysis.

Keywords Data classification · Document classification · Recommendation system · Support vector machine · Movie reviews · Text analysis

A. Sheik Abdullah (✉) · K. Akash · J. ShaminThres
Thiagarajar College of Engineering, Madurai, Tamil Nadu 625015, India
e-mail: aa.sheikabdullah@gmail.com

K. Akash
e-mail: kaakash11c@gmail.com

J. ShaminThres
e-mail: shaminthres@gmail.com

S. Selvakumar
GKM College of Engineering and Technology, Chennai 600063, India
e-mail: sselvakumar@yahoo.com

1 Introduction

The process of sentiment analysis provides the mechanism of determining the state of attitude and subjective material for the given textual data. It also extracts the opinion or emotions from the textual data which can be further deployed in making decision/decision analysis. In recent days, sentiment analysis is used in social and health care applications for collecting and reviewing customer responses. Sentiment analysis basic task is to classify the given textual data to be as positive or negative. The advanced text analytic process is used to classify the emotions of a person as happy, sad, or angry. In simple terms, sentiment analysis predicts the psychological behavior of a person. In social media, sentiment analysis is used to find out reviews corresponding to products/movies/recommender systems. The aspect of sentiment analysis deals with the process of relative mining which in turn determines the specific information from the available data and it helps the vendor in identifying their product views from the customer point of view, so the reviews of social media are restricted to count. With the invent of deep learning algorithms text analysis is improved far better now. Artificial Intelligence techniques act as a tool for doing sentiment analysis in depth. In recent days sentiment analysis is used in Face book and Twitter data.

Social media are the main source for sentiment analysis. Sentiment Analysis is used mainly to classify text, based on the dataset sentiment analysis classify text as binary (positive, negative or neutral) or multiclass. Preprocessing is the initial step in sentiment analysis, several techniques are used for preprocessing, some of them remove numbers, remove punctuation, remove stop words, stemming, etc.,

Sentiment analysis process is segregated into two types such as machine learning-based and lexicon-based analysis. The mechanism behind the machine learning-based approach involves algorithms relative to supervised and un-supervised learning schemes. The scheme of supervised learning involves class labeled training data tuples for the given available dataset. In un-supervised scheme the training data tuples won't be provided with class labeled tuples, the tuples once trained then will be assigned for classification/prediction process. The mechanism of lexicon-based approach involves the process of determining the sentiments (positive/negative) from the given semantic context of the observed word/phrase from the given text.

2 Literature Study

The authors [1] deployed the process of sentiment analysis for data corresponding to Twitter (social media) with the analysis from the tweets collected from the users. The process is then segregated into positive/negative/neutral. The classification process has been carried out using Naïve Bayes, random forest algorithm, and logistic regression analysis. The authors proposed a new classifier approach called ensemble classifier which combines the base classifier into a single classifier, with the intention to improve the accuracy and the performance estimation of the sentiment analysis

process. The implementation is carried out using python programming (data in the form of multi-dimensional array). For data preprocessing, Scikit-learn and Natural Language Toolkit are used. Recall, Precision, and F-Call are used as performance metrics. Ensemble classifier predicts higher accuracy when compared to traditional classification algorithms.

Author [2] use sentiment lexica are used to check the polarity estimation by matching the words and sentiment polarities in the given text. Sentiment Classification model is used in the proposed methodology and semantic similarity metric is used to measure the performance. Proposed methodology includes two sub-modules: document embedding and semantic similarity. For similarity-based feature extraction SIMON algorithm is used and embedding text representation selection over a lexicon is used.

Authors [3] give an overview of sentiment analysis, Challenges in the evaluation phase of sentiment analysis. Forty-seven papers related to sentiment analysis are used for this survey and classify papers based on domain oriented, Challenge type, Sentiment Analysis Challenge, and Review Type. For Sentiment Analysis Challenge types BOW technique, POS technique, semantic technique, lexicon technique, maximum entropy, and n-gram techniques are used. To improve the accuracy of sentiment analysis various models are used from linear to neural network models. Now a day, deep learning methods are used in all fields of research including sentiment analysis. Moreover, some of the Arabic tweets seem to be complex with different dialects. To solve this, the proposed methodology is used for analysis. The work proposed by the authors [4] used two sorts of tweets: one such corresponding to the Saudi dialect and the other one corresponds to the sentiments of Arabic dictionary. Performance metrics such as F-measure, precision, and recall is used for measuring the efficacy of the proposed approach [5].

Sentiment analysis in Arabic tweets is emerging today. In this research work, the authors considered three forms of approaches to perform the task of sentiment analysis and its evaluation. With the considered dataset algorithms corresponding to support vector and Naïve Bayes performs well for binary classification scheme. For multiclass classification in sentiment analysis multi-nominal Naïve Bayes works well in prediction and classification [6]. Author [7] uses sentiment analysis to investigate the effect of demonetization policy introduced by the Indian Government on November 16, 2015. To conclude that authors find that only 30% of people are unhappy with the policy. Tweets are used to analyze the individual behavior of a common man regarding the demonetization policy.

Authors [8] use lexicon-based sentiment analysis for movie reviews dataset and finding accuracy for evaluation. Authors use deep learning for extracting features from raw data, movie review dataset is used and Google's algorithm Word2Vec is applied in the movie review dataset to find the semantic associations. As a result, authors provide a comparison among different clustering algorithms and types of classifiers [9].

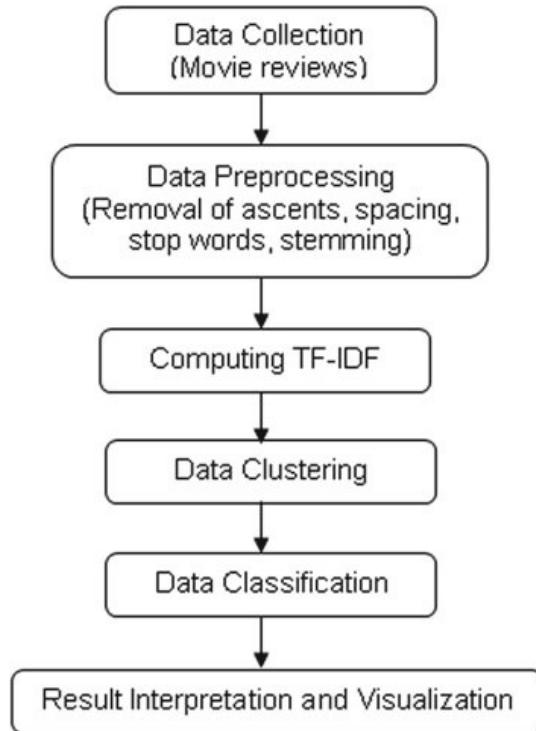
3 Methodological Workflow

3.1 Collection of Data Set

The movie dataset collection is done with a rapid miner using Twitter API. Creation of Twitter API is done by creating a user's Twitter account. The Twitter API involves the creation of an app called Movie Critic. Installation of rapid miner is done and a process is created. The linking of Twitter API is done by setting up connection type as Twitter connection followed by giving access token and authorizing it. Query field is given as rapid miner and result type is recent and limit is given. Now a set of tweets is obtained and converted to .csv files.

Using the rapid miner tool we have extracted tweets by the English language. Below is the graph showing the distribution of a sample movie by language. The most frequently appearing terms as related to the movie reviews collected are represented in the above figure for good classification and predictive results. Also, the positive and negative reviews are ascertained based on the observed conjunctions, terms, prepositions, and verbal contents. The colors in the Fig. 1 signifies the maximum level of appearance of the above terms with regard to the reviews collected. There are three different forms of sentiment classification such as document, sentence-level, and feature-based.

Fig. 1 Proposed methodological workflow



3.2 Data Preprocessing

Preprocessing is a first step needed to be done for efficient algorithm processing. Here, eliminating the noisy text from the retrieved data set or tweets. It has these steps: Removal of stop words like articles, prepositions are done mainly. In this context stop words corresponds to common words such as the, on, which, etc.

3.3 Computing TF-IDF

An IR model is a quadruple consisting of the following terms D, Q, F, and R (q, d). Here, R corresponds to a ranking function. In general, each retrieval document is provided by a set of keywords. Here index term is used to represent or summarize the document contents and its words. Therefore, the vocabulary is represented in Eq. 1 as:

$$V = \{k_1, k_2, k_3 \dots k_t\} \quad (1)$$

Therefore, the given set of documents and the queries can be represented in terms of patterns of term-cooccurrences in Eq. 2 as

$$V = \begin{bmatrix} k_1 & k_2 & k_3 & k_t \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (2)$$

The occurrence with regard to the given term k_i for the doc d_j provides a relation among k_i and d_j . The term relation between them is determined by the frequency of the document and is represented in Eq. 3 as:

$$\begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \\ f_{3,1} & f_{3,2} \end{bmatrix} \quad (3)$$

where the value of each $f_{i,j}$ corresponds to the frequency of the term in the document d_j .

The following are the five distinct variants used for IDF computation:

1. Unary—1
2. Inverse frequency— $\log \frac{N}{n_i}$
3. Inv. Freq. smooth— $\log \left(1 + \frac{N}{n_i}\right)$
4. Inverse freq. Max— $\log \left(1 + \frac{\max_i n_i}{n_i}\right)$
5. Probabilistic inv. freq— $\log \frac{N - n_i}{n_i}$

3.4 Clustering

The movie reviews considered for this research work involve the utilization of k-means clustering algorithm. The feature or aspect with the consideration of given text has to be analyzed in accordance with the number of levels in the selected cluster. This algorithm determines the levels of grouping with regard to the similarity among the mean distance measure and the cluster centroids. The proposed workflow is illustrated in Fig. 1.

3.5 SVM Based Classification

Support vector machine classifiers are widely used and most promising algorithms for handling data at all levels including textual data. The algorithm deploys the mechanism of non-linear mapping to transform the original observed training samples to a level of higher order dimensions. Thereby, with the available form of dimensional level it searches for the optimal linearly separable hyperplane which is also considered to be the decision level for segregating the tuples of records from one class to another (Boser et al. 1992; Vapnik 1998).

Consider the case for a two-class problem (linearly separable):

Let D —The dataset consisting of the training cases.

The representation of the dataset is given in Eq. 4 as:

$$(X_1, y_1), (X_2, y_2), \dots, (X_{|D|}, y_{|D|}) \quad (4)$$

where X_i —corresponds to the set of training data tuples associated with the class label y_i . The value of y_i signifies the value of the label which is either 0 or 1, in case of binary labels and it may range $[0-n]$ in case of the multiclass label.

After we group the datasets under different clusters as per the number of clusters provided by the user by the usage of k-means clustering, the obtained data is then used for classification sentimentally. By doing the clustering process, we get to divide the data in a better way and utilize the data in a more efficient manner. The final sentiment classification is done using the SVM classifier. For the two-class problems SVM is generally put to use. Using the space vector machine classification we feed the tweets and classify them as positive, negative, and neutral ones. After classifying the tweets into their nature we then feed it in terms of a graph. This brings out the accuracy of the process in an accurate manner.

4 Experimental Results and Discussion

In order to perform the mechanism of sentiment analysis for the given set of text corpus, the table corresponding to sentiment and the selected text has to be indicated. The row has the textual part and its segregated sentiments. The column signifies the sentiments accordingly with the classified labels. The data has been taken from movie reviews which signify whether the movie comments represent to positive or negative. The designed model works accordingly with regard to the opinion that has been classified with the corresponding sentiments. Therefore, the opinions considered with the free form of text review are identified from the library executed files. The stop words, noun phrases are identified with the sentiment analysis process and the word cloud is formed as a result. With the observed set of opinion pairs, the nearest observed opinion pairs are analyzed and recorded.

One of the major functionalities of SVM classifier is the estimation value for overfitting or under-fitting which can be observed by choosing the parameter C. The value of C determines the trade-off value between the complexity and the proportion of data samples that have been selected by the user. The range of the value C determines the output values of the SVM classifier with the effect of the determination of outliers in the observed training data. The value corresponding to the observed parameter determines the marginal size and the size of the variables.

The ϵ value of the Support Vector is considered to be 50% of the sample ratio. Hence we can easily compute the optimal generalized performance to be smaller than 50%. Also, the value of ϵ must be smaller for a larger sample size than that of smaller sample. When compared to other classification techniques SVM has found to be showing an improvement in accuracy than other classification techniques such as Naïve Bayes, Neural Network, Decision Trees, and Decision stump. Figures 2 and 3 provide the set of most positive and negatively classified words in the collected documents. The comparison of accuracy among the data classification algorithm is illustrated in Table 1.

The behavior of the model developed is observed through the analysis from the applicability of different kernel functions (linear/polynomial/Gaussian/string). Linear Classifier is the first new best hyperplane algorithm was proposed by Vapnik in the year 1963. After 30 years from the invention of linear classifier hyperplane algorithm, kernel trick has been brought into practice.

Linear Kernel is the efficient mechanism for handling textual data, and it is suggested for text classification due to the following reasons:

- It linearly separates the text data,
- It supports a lot of features in a text when compared to RBF Kernel,
- By using a dedicated library LibLinear, linear kernel trained SVM faster when compared to other kernels,

These are the reasons; Linear Kernel is best suited for text classification. Linear Kernel functions used in SVM can be evaluated using the following Eq. 5 as:

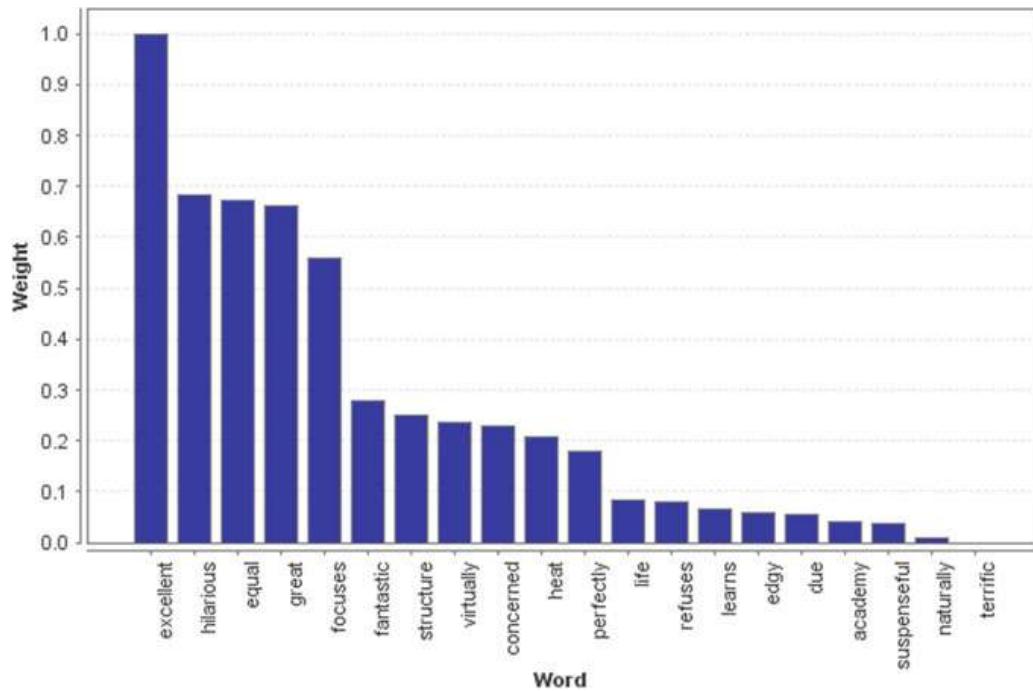


Fig. 2 Most important words indicating positive sentiment in the corpus

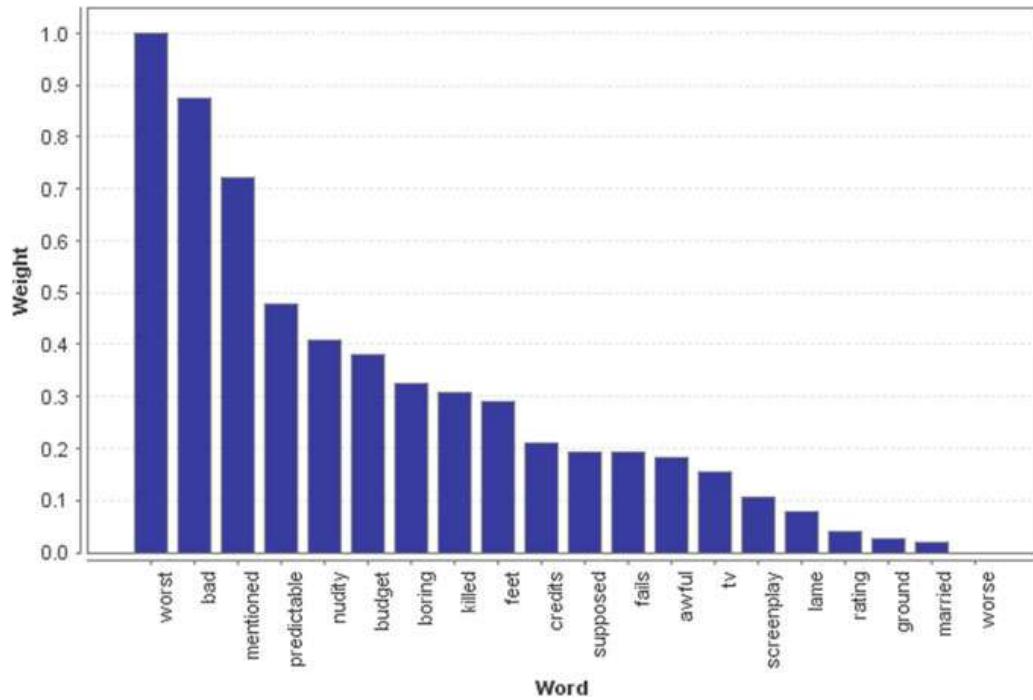


Fig. 3 Most important words indicating negative sentiment in the corpus

Table 1 Comparison of accuracy among data classification algorithms

S. No.	Algorithm	Accuracy (%)
1.	Decision trees	74.25
2.	Support vector machines	87.56
3.	Neural network	69.88
4.	Random trees	79.02
5.	Naïve Bayes	81.14

$$\text{Linear } k(x_1, x_2) = x_1 \cdot x_2 \quad (5)$$

Hence it has been observed that data classification using SVM classifier for movie review dataset. The kernel function used is the linear kernel for the text classification process. This kernel performs well when compared to other kernels that are suited for SVM classification. The value of C has been optimized with the processing nature of linear kernel [10]. With this configuration parameters linear kernel suits well for a larger set of files in the text classification process.

5 Conclusion

Sentiment analysis has become one of the most important and emerging fields in all disciplines. The realm of determining the positive and negative opinions makes the judgment in all concerns. This paper deals with the analysis of movie review data by using data clustering and data classification process. The classification algorithm used in SVM with a linear kernel process. The classification process produces 87.56% of accuracy level in segregating the positive and negative opinions for the movie review data. Hence this can be used for classifying other social media data in future findings. The future work will be the process of collecting real-time tweets for classifying a socially relevant activity with the determination of positive and negative tweets.

Acknowledgements We would like to acknowledge that we have collected reviews in publicly available repositories for the evaluation and the development of the model.

Declaration We have taken permission from competent authorities to use the images/data as given in the paper. In case of any dispute in the future, we shall be wholly responsible.

References

1. Saleena, N.: An ensemble classification system for twitter sentiment analysis. In: International Conference on Computational Intelligence and Data Science (ICCIDDS), Procedia Computer Science, vol. 132, pp. 937–946 (2018)

2. Araque, O., Zhu, G., Iglesias, C.A.: A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowl. Based Syst.* **165**, 346–359 (2019)
3. Hussein, D.M.E.D.M.: A survey on sentiment analysis challenges. *J. King Saud Univ. Eng. Sci.* **30**, 330–338 (2018)
4. AlThubaity, A., et al.: Sentiment lexicon for sentiment analysis of Saudi dialect tweets. In: The 4th International Conference on Arabic Computational Linguistics (ACLing 2018), Procedia Computer Science vol. 142, pp. 301–307 (2018)
5. Heikal, M., Torki, M., El-Makky, N.: Sentiment analysis of Arabic Tweets using deep learning. In: The 4th International Conference on Arabic Computational Linguistics (ACLing 2018), Procedia Computer Science vol. 142, pp. 114–122 (2018)
6. Boudad, N., et al.: Sentiment analysis in Arabic: a review of the literature. *Ain Shams Eng. J.* **9**, 2479–2490 (2018)
7. Singh, P., et al.: Sentiment analysis of demonetization of 500 & 1000 rupee banknotes by Indian government. *ICT Express* **4**, 124–129 (2018)
8. Anandarajan, M., Hill, C., Nolan, T.: Sentiment analysis of movie reviews using R. Practical Text Analytics as a part of the Advances in Analytics and Data Science book series, pp. 193–220 (2018)
9. Guha, R., et al.: Deluge based genetic algorithm for feature selection. *Evol. Intell.* 1–11 (2018)
10. Abirami, A.M., et al.: Sentiment analysis. In: *Handbook of Research on Advanced Data Mining Techniques and Applications for Business Intelligence*, IGI Global. (2017). ISBN 978-1-5225-2031-3

Blockchain-Based Sybil-Secure Data Transmission (SSDT) IoT Framework for Smart City Applications



Sonal Kumar, Ayan Kumar Das, and Ditipriya Sinha

Abstract The application of the Internet of Things (IoT) is resulting in smart and advanced technology. The smart cities involve smart utilities. The vision of a smart city ensures accessing billions of IoT devices from commonplace. The communication between IoT devices of smart cities demands privacy. Since the sensitive data generated from the smart devices are vulnerable to various attacks. One of the major attacks perform in the IoT network is the Sybil attack, in which an adversary creates multiple identities of a node to harm the network. Blockchain is an emerging technology that can be used to meet the demand for privacy preservation in IoT based smart cities. This paper proposes a Blockchain-based Sybil-secured data transmission (SSDT) IoT framework for smart city applications. The proposed frameworks effectively capture any node which is trying to initiate the Sybil attack as well as provides a secure platform for devices to exchange information with each other. The performance of the proposed Blockchain-based SSDT IoT framework is simulated by designing a Sybil-Secured Data Transmission (SSDT) simulator.

Keywords Blockchain · Internet of Things · Sybil attack

1 Introduction

The rising population density in metropolitan cities is growing very fast. So, the number of people using the internet and their necessities are increasing. It is expected by 2050; at least 70% of the world population (approximate 6 billion) will be part

S. Kumar (✉) · D. Sinha
National Institute of Technology, Patna, Bihar, India
e-mail: sonal.cs18@nitp.ac.in

D. Sinha
e-mail: ditipriya.cse@nitp.ac.in

A. K. Das
Birla Institute of Technology, Patna, Bihar, India
e-mail: das.ayan777@gmail.com

of the urban area [1]. It is a challenging task to provide quality of life to all the city residents and fulfilling their various necessities. Cities around the world are looking for optimum solutions to meet new challenges, which are constantly growing on space and time [2]. Thus, smart city has been considered to solve the problems of global urbanization. In a smart city, smart services are accessible regardless of time or location and smart cities are equipped with various smart devices to achieve such suitability.

Internet of Things (IoT) is the key to accomplish the smart city vision. The smart city and the Internet of Things (IoT) with different origins are stepping forward to each other in order to achieve a common goal. The act of Sybil attack can affect the data integrity, throughput and performance of the network as well as can bring down the whole IoT network over time. Prevention of IoT network from Sybil attack is one of the challenging tasks.

In this paper, we propose a Sybil-secured data transmission (SSDT) IoT framework, which is based on the Blockchain technology. This proposed SSDT framework is secured cryptographically with immutable distributed ledger technology. Also, data generated and broadcasted by IoT devices are stored into blocks of the Blockchain. In order to detect Sybil nodes, Blockchain provides hash-based security to verify the identity of IoT devices. The numerous key characteristics of Blockchain technology are persistency, decentralization, audit-ability, and anonymity [3]. These various characteristics of Blockchain improve the functionalities of the proposed SSDT IoT framework. We are also implementing the Blockchain-based Sybil-Secured Data Transmission (SSDT) simulator to realize the functionalities of the proposed SSDT framework.

2 Backgrounds

2.1 *Blockchain Technology*

A Blockchain can be defined as a distributed ledger; where data can be stored in a secure manner so that the alteration or modification of data is not possible. The sequence of blocks in a Blockchain maintains a complete list of transactional records between the nodes of a network, where the exchange of messages between two nodes can be represented as a transaction. Figure 1 shows the design structure of the Blockchain and components of a block.

2.2 *Sybil Attack in IoT Network*

Sybil attack is one of the major security threat performed over the peer to peer network. In a Sybil attack, an adversary node acquires multiple identities by either

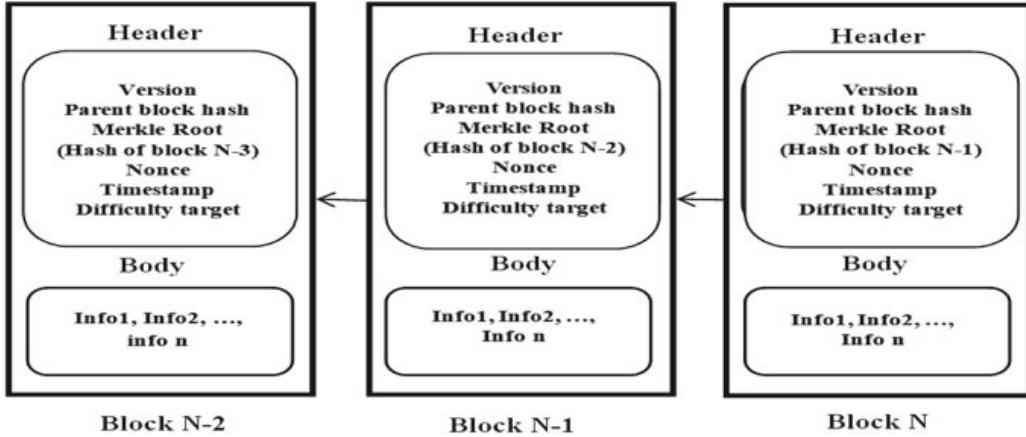


Fig. 1 Data structure of Blockchain

fabricating or stealing the identity of a legitimate device, which is called Sybil node [4, 5]. A Sybil node joins the network with hidden purposes; such as false root broadcast, selective forwarding of packet, packet dropping and compromise the sensitive data.

3 Literature Survey

In this section, we go through different literature survey based on “Blockchain-based solutions” for IoT enabled smart city applications. Many researchers have been done in the interest of benefiting IoT infrastructure. In [6], a new Blockchain-based security model and related protocol are developed, which ensure the integrity and validity of cryptographic authentication data and associate peer trust level, throughout the sensor network lifetime. In [7], Oscar Novo has proposed a distributed IoT management to address the challenges of already existing security management frameworks for IoT using Blockchain technology. Tosh et al. [8] proposed Blockchain-based data provenance architecture to securely record the data operations performed in the cloud environment. In [9], the authors proposed a Blockchain-based financial product management platform, which reduces the information update delay among participating institutions. In [10] authors used the transparency and no-tempering feature of Blockchain to implement Hyperledger framework-based education industry cooperative system. In [11], the authors introduced an end to end inter-bank payment system prototype built upon the Hyperledger fabric enterprises Blockchain platform. In [12], the authors proposed a novel Blockchain-based contractual routing (BCR) protocol to support a network of distrustful devices. Bahga et al. [13] proposed a Blockchain platform where machines are owned with respective Blockchain accounts and can directly transact with the machines to avail manufacturing services for Industrial IoT. However, Blockchain technology seems promising; it has to address various challenges such as smart contract vulnerabilities, awareness towards Blockchain

technology, privacy preservation and regulations to ensure legal enforceability and its widespread adoption. The security framework for a smart city is presented in [14].

4 Proposed Sybil-Secured Data Transmission (SSDT) IoT Framework

In this section, we propose a Sybil-secured data transmission (SSDT) framework for the IoT environment, which is based on the Blockchain technology. The framework has features like a secure mechanism to join the network, strong identity management, data integrity, and authentication to prevent the Sybil attack. The following subsection describes the functionalities of the SSDT IoT framework.

4.1 Functionalities of SSDT IoT Framework

As shown in Fig. 2, the device setup stage, Dual identity verification stage and secure data transmission stage assist to design proposed Blockchain-based SSDT IoT framework and enable the working procedure of Blockchain-based data sharing the architecture of IoT network. The dual identity verification stage establishes an authentication and identity verification to capture Sybil nodes.

Device setup stage: The device, which wants to join the Blockchain network, has to go through the device setup stage. In this stage, each device receives a unique identity set that cannot be duplicated by any device in the IoT network. It will be used in further stages to authenticate the devices. The steps of the device setup stage are:

- Step 1: The device will generate a wallet having a public/private key pair for the encryption process at the Blockchain server. The wallet will also hold some balance in it, which is sufficient for performing the transaction in the first stage.
- Step 2: The device will perform the transaction of some predefined amount to a pre-decided receiver wallet address in the Blockchain network. The transaction will be validated by the available miners of the Blockchain network.

Fig. 2 Stages of the proposed framework

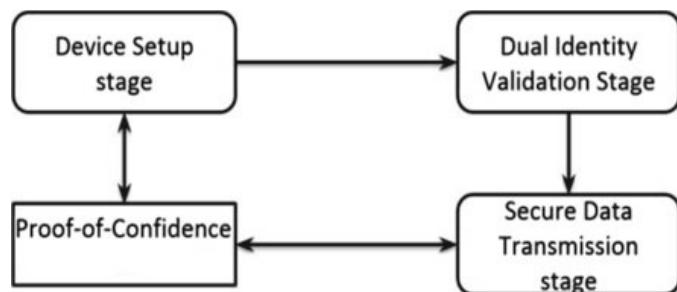


Fig. 3 Structure of identity set

Identity Set
Block number
Block hash
Transaction index/id
Merkle tree hash
Device ID
Public key

Step 3: The validated transaction will be mined with the public key of the sender device by the winner of the consensus process as a part of Block_K of Blockchain.
 Step 4: An **Identity set** will be generated for the device, which will contain some information from the Block_K like block number, block hash, transaction index, and the hash of transaction, the hash of Merkle tree, and the public key of the device and Device ID.

As shown in Fig. 3, the **Identity set**, stores a uniquely generated Id namely Device ID, which will be generated using the hash-based message authentication code (HMAC). The HMAC will use block hash as key with the transaction index to calculate the Device ID using Eq. 3. Since the Device ID will be calculated using the block hash and transaction index, it will be difficult for intruders to predict Device Id of devices in the Blockchain network.

$$\text{Key} = \text{hash of Block}_K \quad (1)$$

$$\text{Txn_Index} = \text{index of related transaction in Block}_K \quad (2)$$

$$\text{Device ID} = \text{HMAC}(\text{Key}, \text{Txn_Index}) \quad (3)$$

Dual identity verification stage: The dual identity validation stage basically ensures the sender/receiver side that the device on another side is a legitimate device of Blockchain network or not and detects Sybil attack. The steps of the dual identity validation are:

Step 1: The sender and receiver will validate the structure of the identity set of each other. A legitimate identity set structure will be comprised of block number, block hash, transaction hash, Merkle tree hash, and Device ID.

Step 2: The validating device will verify the block hash and use Merkle tree hash to check whether the transaction shown in the identity set is part of a block or not.

Step 3: The transaction details will be verified such as the transaction amount is equal to the predefined amount or not, the receiver address is the same as the

pre-decide receiver wallet or not, and whether the transaction is signed properly or not.

Step 4: The Device ID of Identity set will be verified by using Eq. 3.

Step 5: If both sender and receiver will be verified successfully and found legitimate in all the steps then devices can initiate the data transmission request in further process. Else, an alarm is generated to inform others about the presence of Sybil node in the network and no further communication will be done with the device.

Secure data transmission stage: The steps of the Secure Data Transmission stage are:

Step 1: At the sender end, the sender encrypts as Encrypted data (ED) using the RSA algorithm and public key of the receiver as key.

Step 2: The hash of the ED is computed by using SHA256 as Hashed data (HD).

Step 3: The digital signature of the Hashed data is obtained using the RSA algorithm and private key of the sender as key as Digitally signed hash data (DSHD).

Step 4: The DSED is appended with the ED and send to the receiver as Message text (MT).

Step 5: At the receiver end, the receiver will receive the message as MT_{New} and extract digital signature as $DSED_{New}$ and encrypted data as ED_{New} .

Step 6: Now, the receiver will compute the hash of ED_{new} using SHA256 as HD_{new} .

Step 7: The receiver will compute the digital signature of HD_{new} using RSA and the public key of the sender as key.

Step 8: If the computed signature and $DSED_{new}$ will be the same receiver will accept the ED_{new} and decrypt it using its own private key. Else, ED_{new} will be discarded.

The information generated while performing the steps of the device setup phase cannot be duplicated completely. Hence, even if Sybil node will join the network as a legitimate device, it will not be able to set multiple identities.

4.2 Architecture of SSDT Framework

In the proposed Blockchain-based IoT framework as shown in Fig. 4, the network comprises two types of node Blockchain node and client node. All the smart devices in the network are client nodes and the Blockchain nodes will be selected among client nodes. The Blockchain node will perform all the functions of the client node and in addition, it will also participate in the Blockchain process.

$$T(n) = \begin{cases} \frac{P}{1-P(r \bmod \frac{1}{P})}; & \text{for participating nodes} \\ 0; & \text{Otherwise} \end{cases} \quad (4)$$

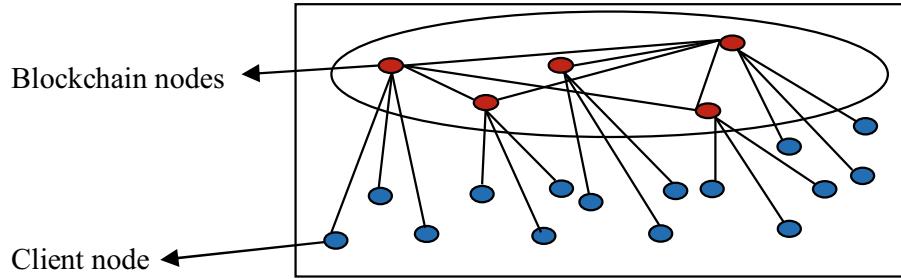


Fig. 4 Decentralized data-sharing architecture

$$T(n) = \begin{cases} \frac{P}{1-P(r \bmod \frac{1}{p})} \times \frac{E(\text{Residual})}{E(\text{Initial})}; & \text{for participating nodes} \\ 0; & \text{Otherwise} \end{cases} \quad (5)$$

Equations 4 and 5 form the cluster and cluster head for the Leach algorithm and Residual energy-based Leach algorithm, respectively [15]. Here, $T(n)$ is a Threshold value to select cluster head, N is Probability to opt for cluster head, P is Probability to opt for cluster head, $E(\text{Residual})$ is Residual energy of node and $E(\text{Initial})$ is Initial energy of node. The step-wise working procedure of the proposed system model is:

Step 1: All devices in the existing IoT network will perform the device setup stage (explained in sub-Sect. 4.1) to join the Blockchain network.

Step 2: Now, all IoT devices will form clusters and cluster heads for the selection of Blockchain nodes among them applying Eqs. 4 and 5. Cluster head will act as a Blockchain node of the cluster.

Step 3: Once the Blockchain node is selected, all devices will start exchanging the data using a dual identity verification stage followed by a secure data transmission stage (explained in sub-Sect. 4.1).

Step 4: Blockchain node will validate the transaction request of the client node of their cluster and broadcast it to the other Blockchain nodes for validation.

Step 5: Blockchain node will add all validated transactions in its local block and participate in the consensus process.

Step 6: Winner of the consensus process will add its local block into the Blockchain and broadcast it to other Blockchain nodes and also update at cloud/base station.

Step 7: Other Blockchain nodes will update their Blockchain chain and broadcast received block in their cluster.

The client devices of the network will be connected via a Blockchain network and all the transactions are done in the network will be stored in the Blockchain by each device. The device in the network can store the Blockchain formed in a given duration of time to avoid the memory overflow but cloud/base station will store complete Blockchain since the beginning.

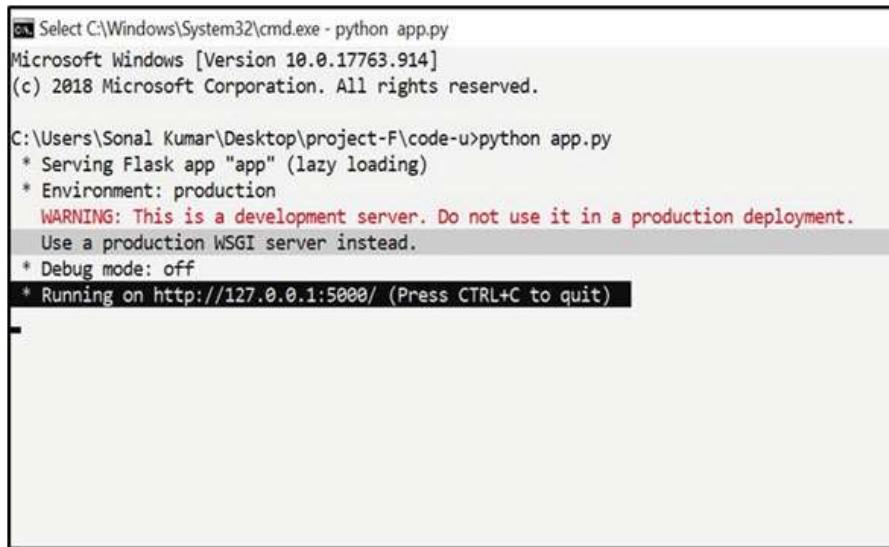
5 Experimental Setup and Results

In this section, we will implement a simulator to realize the functionality of the Blockchain-based SSDT IoT framework to capture the Sybil nodes. The name of the simulator is the SSDT simulator.

5.1 SSDT Simulator

In the SSDT simulator, we have used python to implement the backend part and HTML to implement the frontend part of the simulator. The Flask 1.1.1 micro framework is used to integrate the front end and backend part and it is using XAMPP as a local server. The simulator ensures some features such as setting up a required number of device in Blockchain network, generating wallet for every joining device, generate identity set for every legitimate device of Blockchain network, verification of identity set of device to allow data exchange, provide a secure data transmission portal for Blockchain device and mine the block of a valid transaction to maintain the Blockchain. As shown in Fig. 5, in terminal “python backend_code.py” command, will be executed to compile the backend code. Figure 6 shows the device setup web page of the frontend, it asks to input the number of Blockchain devices to be configured and then click the submit tab. The submit tab of device setup web page will generate wallet with key pair for requester number of the device and open the next page namely wallet Detail in the same window as shown in Fig. 7.

The wallet details web page contains a table storing wallet id with the public key for the requested number of the device and an identity set generation tab. The



```
Select C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Sonal Kumar\Desktop\project-F\code-u>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig. 5 Compilation of backend code

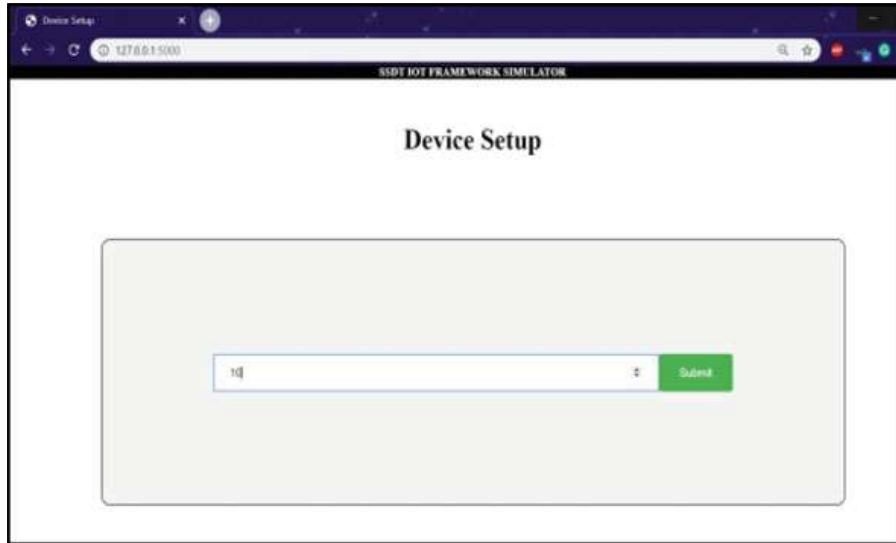


Fig. 6 Device setup web page view

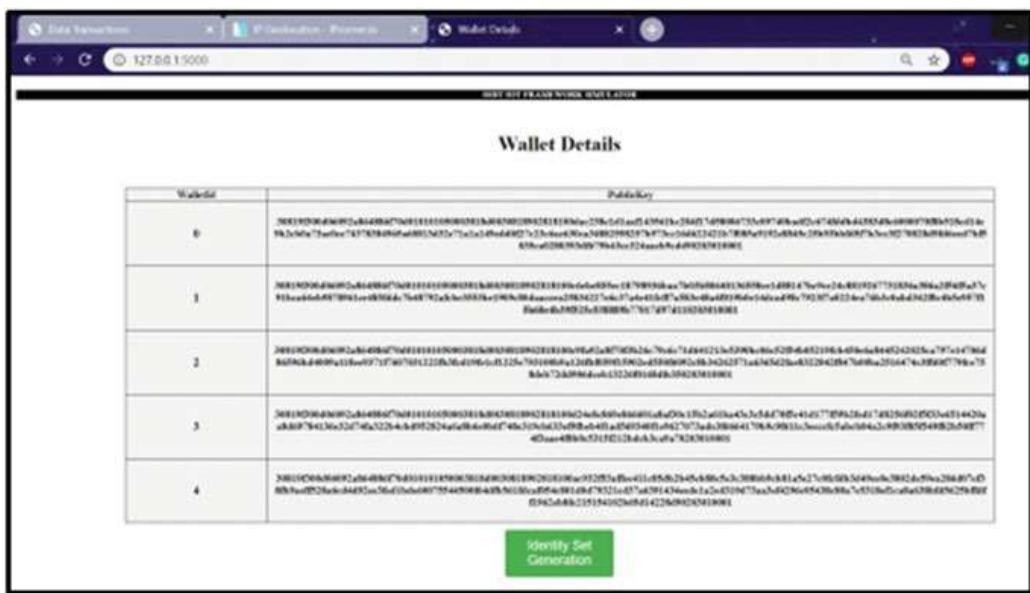


Fig. 7 Wallet details web page view

identity set generation tab will open a new page namely Device Setup Transaction as shown in Fig. 8, where the device can perform transaction of a predefined amount to pre-decided receiver wallet as explained in the device setup stage of subsection 4.1. For every valid transaction, a block will be created and added to the Blockchain. Using the information of added block and device wallet; an identity set of the device will be generated.

The submit tab of Device Setup Transaction web page will submit the transaction for all devices one by one and then it will open a new web page namely Block Detail

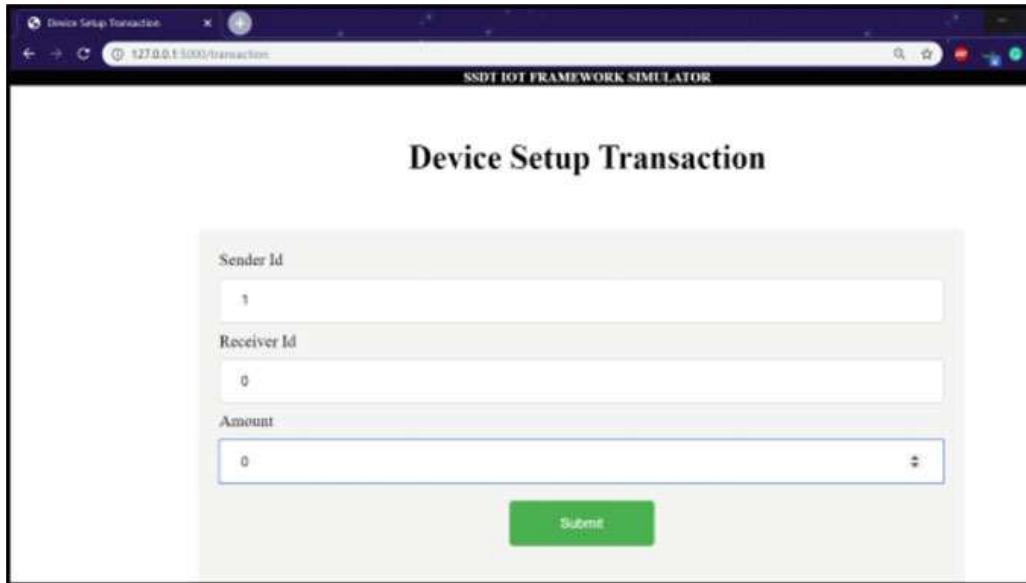


Fig. 8 Transaction web page view

as shown in Fig. 9. The Block Detail web page contains a table storing details of each block of Blockchain like Previous hash, Current hash, Transaction time, Nonce, Merkle root, and Transactions with a View Valid Transaction tab.

The View Valid Transaction tab of Block Detail web page will open a next web page namely Valid Transactions. As shown in Fig. 10, Valid Transaction web page list out details for all Valid Transactions performed in-network with a View identity

Fig. 9 Block detail web page view

Transaction Id	Sender Address	Receiver Address	Amount / Message
Txn-1	11f3d2830af8d005fb6732d5bf5fc32ea1a085c	35b7019695d502e9a516829ac128551c36cf3c67	0
Txn-2	ddd8a0e7284691069013ed249a6fc1b1ba74bf9	35b7019695d502e9a516829ac128551c36cf3c67	0
Txn-3	48fd1838ch963ce07bd8d77ddh38ec177eedf180	35b7019695d502e9a516829ac128551c36cf3c67	0
Txn-4	1860dbe92fba57998821cd1268da33f58f414a8c	35b7019695d502e9a516829ac128551c36cf3c67	0

View Identity Sets

Fig. 10 Valid transaction web page view

sets tab. The View Identity sets tab of the Valid Transaction web page will open the next web page namely identity set. As shown in Fig. 11, it contains a table storing the identity set of legitimate devices of the Blockchain network with a Perform Data Transmission tab.

Device No.	Public Key	Block No.	Block Hash	Difficulty	Transaction Index	Device ID
0	30819D00d...	0	988793b4ee1c82d553742b4b5967d55df83713	3	1	8363d7e749253f3a416c5260346fe5
1	30819D00d...	1	0944cc04ec53c53764db446691575765109c108	3	1	7779393792f1a0e87130fa4b4b7b6d
2	30819D00d...	2	002095582767110ec5cde5a7101d56b5a33bc	3	1	55d71367c1bb997ea14e05449vh13
3	30819D00d...	3	00e057240f24b133899f022cf8d64ff95C283a2c	3	1	5e39e7de8847063bcd717404d1b373e
4	30819D00d...	4	00ea4c3408d0e21065f8e068a46682944315771	3	1	2ab9a4b678316355a322a390ca4f0b3

Perform Data transactions

Fig. 11 Identity set web page view

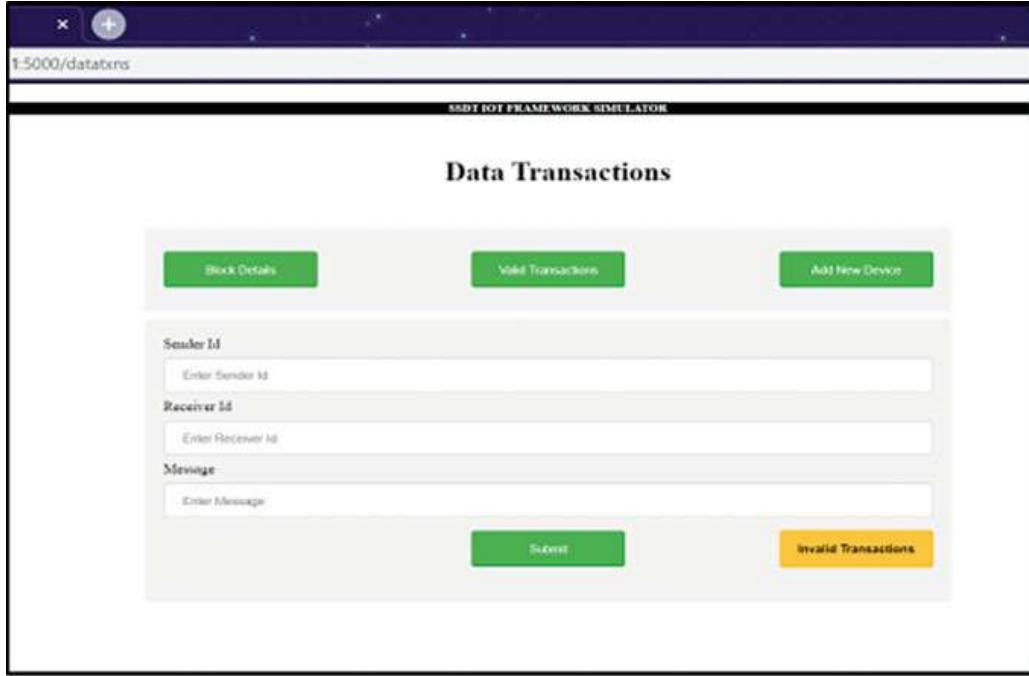


Fig. 12 Data transmission web page view

The Perform Data Transmission tab will open the next web page namely data transmission. As shown in Fig. 12, it contains View Block Detail, View Valid Transactions, and Add New device tab, which will open Block Detail web page, Valid Transaction web page, and device setup web page, respectively. It also contains three input lines with one submit tab to enable secure data transmission between devices. All successful transmission will be added in the block and the block will be appended to the Blockchain in the backend.

5.2 Results and Discussion

The malicious device can be detected in various stages of the SSDT framework. We simulated the Blockchain-based SSDT framework using the SSDT simulator implemented in Sect. 5.1 and initiated the Sybil attack in various phases. The malicious devices intended to initiate the Sybil attack is identified and disabled by the SSDT simulator for further operation.

As shown in Figs. 13 and 14, device number 9 tries to join the network by performing a transaction with some other device instead of receiver wallet (in our case device number 0 is selected as receiver wallet and every device supposed to send 0 unit to device number 0). The simulator identifies the malicious device and the transaction request is discarded. As a result, the transaction is not included in any block and the identity set of devices is not generated.

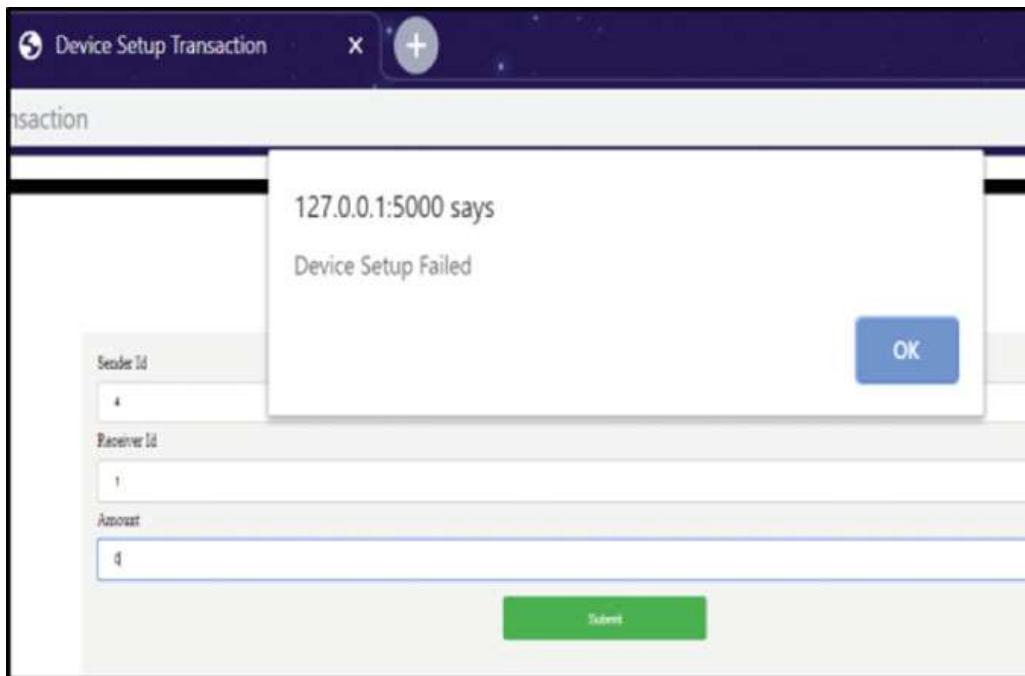


Fig. 13 Frontend view of Sybil detection in the device setup phase

```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Sonal Kumar\Desktop\latest>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [14/Nov/2019 16:30:01] "GET / HTTP/1.1" 200 -
app.py:289: DeprecationWarning: HMAC() without an explicit digestmod argument is deprecated since Python 3.4, and will be removed in 3.8
key=hmac.HMAC(pk)
app.py:298: DeprecationWarning: HMAC() without an explicit digestmod argument is deprecated since Python 3.4, and will be removed in 3.8
H.idProof['Node_ID'] = hmac.HMAC(key.digest(),txID.encode()).hexdigest()
127.0.0.1 - - [14/Nov/2019 16:30:10] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2019 16:30:13] "GET /transaction HTTP/1.1" 200 -
Transaction is verified
app.py:319: DeprecationWarning: HMAC() without an explicit digestmod argument is deprecated since Python 3.4, and will be removed in 3.8
key=hmac.HMAC(pk)
app.py:328: DeprecationWarning: HMAC() without an explicit digestmod argument is deprecated since Python 3.4, and will be removed in 3.8
S.idProof['Node_ID'] = hmac.HMAC(key.digest(),txID.encode()).hexdigest()
127.0.0.1 - - [14/Nov/2019 16:30:18] "POST /result HTTP/1.1" 302 -
127.0.0.1 - - [14/Nov/2019 16:30:18] "GET /transaction HTTP/1.1" 200 -
Transaction is verified
127.0.0.1 - - [14/Nov/2019 16:30:28] "POST /result HTTP/1.1" 302 -
127.0.0.1 - - [14/Nov/2019 16:30:28] "GET /transaction HTTP/1.1" 200 -
Transaction is verified
127.0.0.1 - - [14/Nov/2019 16:30:41] "POST /result HTTP/1.1" 302 -
127.0.0.1 - - [14/Nov/2019 16:30:41] "GET /transaction HTTP/1.1" 200 -
```

Fig. 14 Backend view of Sybil detection in the device setup stage

Figure 15 shows that in the later stage when device number 9 tries to Perform Data Transmission with any legitimate device in the Blockchain network using the data transmission web page, the identity set of 9 is not found by the receiver. Hence, the transmission request is discarded by the receiver and a warning message regarding the Sybil attack is generated. As shown in Fig. 16, when an outside device tries to send a message to a legitimate device of the network, the identity set of outside devices is not found by the receiver, and transmission request is discarded and warning regarding Sybil attack is generated. We automated the simulator for two hours and analyzed

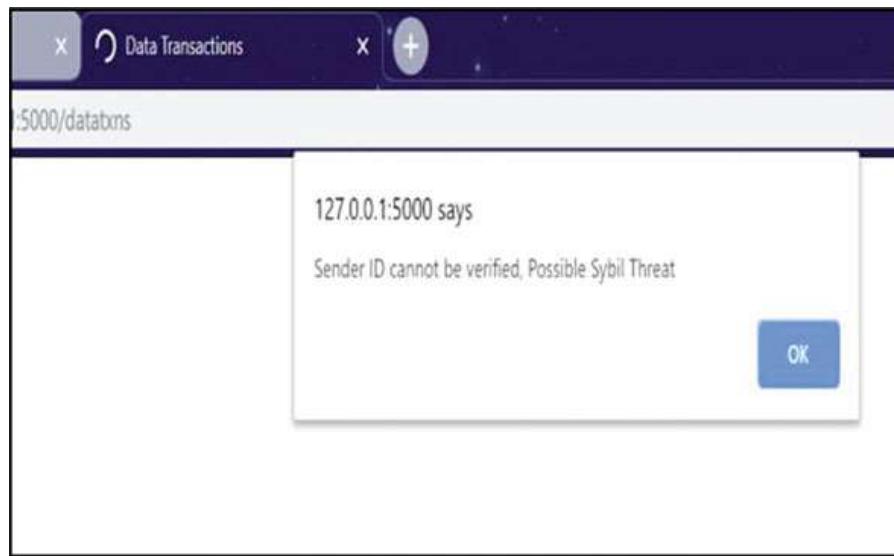


Fig. 15 Sybil node detection in because set not generated

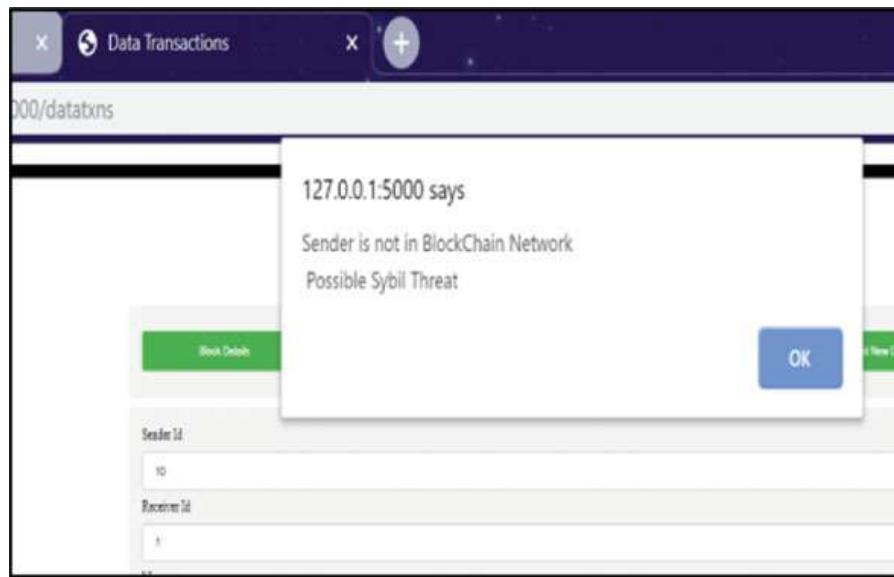


Fig. 16 Sybil device detection regarding identity device from outside network

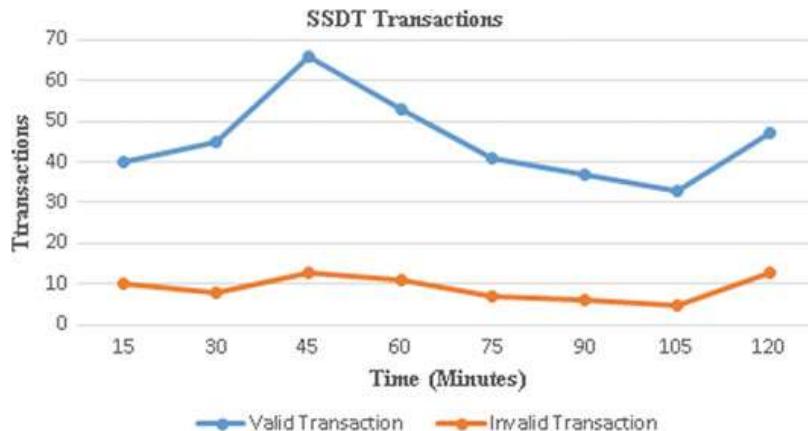


Fig. 17 Valid and invalid transaction performed by SSDT simulator

the generated data of every fifteen minutes. Figure 17 shows the number of valid and invalid transactions performed in SSDT simulator every 15 min.

6 Conclusion

As a critical component of the future world, high level of security is required in smart city. Internet of Things (IoT) is the key to accomplish the smart city vision. Blockchain is proved to be strong against a number of threats. The combination of IoT with Blockchain is an autonomous technology that provides smart and secure applications. Sybil attack is a challenging task in the IoT paradigm. This paper proposes a Blockchain-based IoT framework for smart city applications to prevent Sybil attack and also provides a secure data transmission platform for IoT devices. The performance of the proposed SSDT framework is evaluated by implementing an SSDT simulator. The simulator realizes the functionality of the proposed SSDT simulator to detect possible Sybil thread at various stages.

Acknowledgements This research is supported by Information Security Education and Awareness (ISEA) Project II funded by the Ministry of Electronics.

References

1. Oktaria, D., Kurniawan, N. B.: Smart city services: A systematic literature review. In: 2017 International Conference on Information Technology Systems and Innovation (ICITSI), pp. 206–213. IEEE (2017)
2. Arroub, A., Zahri, B., Sabir, E. and Sadik, M.: A literature review on smart cities: paradigms, opportunities and open problems. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 180–186. IEEE, (2016)

3. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. *Int. J. Web Grid Serv.* **14**(4), 352–375 (2018)
4. Mishra, A.K., Tripathy, A.K., Puthal, D. and Yang, L.T.: Analytical model for Sybil attack phases in internet of things. *IEEE Internet Things J.* **6**(1), 379–387 (2018)
5. Rajan, A., Jithish, J., Sankaran, S.: Sybil attack in IOT: modelling and defenses. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, pp. 2323–2327 (2017)
6. Moinet, A., Darties, B., Baril, J. L.: Blockchain based trust & authentication for decentralized sensor networks. arXiv preprint [arXiv:1706.01730](https://arxiv.org/abs/1706.01730) (2017)
7. Novo, O.: Scalable access management in IoT using Blockchain: a performance evaluation. *IEEE Internet Things J.* **6**(9), 4694–4701 (2019)
8. Tosh, D., Shetty, S., Foytik, P., Kamhoua, C., Njilla, L.: CloudPoS: a proof-of-stake consensus design for Blockchain integrated cloud. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), IEEE, pp. 302–309 (2018)
9. Chen, B., Tan, Z., Fang, W.: Blockchain-based implementation for financial product management. In: 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1–3. IEEE, (2018)
10. Liu, Q., Guan, Q., Yang, X., Zhu, H., Green, G., Yin, S.: Education-industry cooperative system based on Blockchain. In: 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), IEEE, pp. 207–211 (2018)
11. Wang, X., Xu, X., Feagan, L., Huang, S., Jiao, L., Zhao, W.: Inter-bank payment system on enterprise blockchain platform. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 614–621. IEEE, (2018)
12. Ramezan, G., Leung, C.: A blockchain-based contractual routing protocol for the internet of things using smart contracts. *Wirel. Commun. Mobile Comput.* 1–14 (2019)
13. Bahga, A., Madisetti, V.K.: Blockchain platform for industrial internet of things. *J. Softw. Eng. Appl.* **9**(10), 533 (2016)
14. Biswas, K., Muthukumarasamy, V.: Securing smart cities using Blockchain technology. In: 2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS,) pp. 1392–1393. IEEE (2016)
15. Behera, T.M., Mohapatra, S.K., Samal, U.C., Khan, M.S., Daneshmand, M., Gandomi, A.H.: Residual energy-based cluster-head selection in WSNs for IoT application. *IEEE Internet of Things J.* **6**(3), 5132–5139 (2019)

An Empirical Study of Neural Network Hyperparameters



Aditya Makwe and Abhishek Singh Rathore

Abstract The learning algorithms related to deep learning involves many attributes called hyperparameters, these variables help in determining the network structure. The performance of algorithms depends upon these hyper-parameter variables that are needed to be set prior to the actual implementation of the algorithm. This study involves an overview of some of the commonly used hyperparameters in the context of learning algorithms used for training neural networks along with the analysis of adaptive learning algorithms used for tuning learning rates.

Keywords Neural network · Hyper-parameter · Deep learning

1 Introduction

With the rise of deep learning the potential of human being for solving complex problems has increased. Solving these complex problems using deep learning neural network based approach is common in today's scenario. The deep learning technique uses statistical techniques for solving problems on the basis of available sample data in data set. Multiple deep learning architectures like Alex Net, ResNet, VGG GoogLeNet and MobileNet are designed for the purpose. These architectures have yield numerous state of art and result in domains of speech recognition, image recognition, language translation, object detection and many more.

A. Makwe (✉)
Institute of Engineering and Technology DAVV, Indore, India
e-mail: sumitmakwe13@gmail.com

A. S. Rathore
Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India
e-mail: abhishekutjjain@gmail.com

Regardless of availability of various models, one cannot solve all the problems optimally. The reason behind this includes multiple factors like how to select the model, how to train the model, and how to set different parameters of network. The selection of these parameters requires expertise and extensive trial and error approach. Many approaches like Manual Search, Grid Search and Adaptive algorithms are used to decide these parameters. Manual and Grid search [1] are computationally expensive and time-consuming, whereas an adaptive algorithm requires an extensive understanding and study.

This paper gives an overview of different hyperparameters associated with neural network and a study of learning rate hyperparameters related to neural network has been done. The goal of the study is to identify the relationship of learning rate with the other neural network parameters and to identify how adaptive learning approach shows the balance between over-fitting and under-fitting.

2 Literature Survey

Several papers have discussed the hyperparameters related to deep learning based neural networks. Bengio [2] discussed various parameters related to neural network algorithm and its structure; it gives practical recommendations for how to tune hyperparameters, particularly in the context of learning algorithms based on gradient-based optimization.

In the context of setting the values of hyperparameters Smith [3] introduced a method based on cyclic learning rates, which eliminates the need for manually setting the value of learning rate based on hit and trial method. In the Later work by author, the improved method determined the reasonable bounds in which learning rate varies and shows the improvement of classification accuracy on CIFAR-10 and CIFAR-100 data sets with different architectures like AlexNet, GoogLeNet, etc. [4].

Parameters like learning rate and batch size have a collective impact over accuracy of network it is common practice to decrease the learning rate. Smith et al. [5] stated that instead of decaying the learning rate increase the batch size. The approach is best suited for stochastic gradient descent (SGD), SGD with momentum. The designed approach shows equivalent test accuracy for the same number of epochs after updating fewer parameters which leads to shorted training time.

Lorraine and Duvenaud [6] give an approach to solve the problem of tuning network hyperparameters using gradient-based optimization. The proposed approach optimizes the hyperparameters and weights for large networks having multiple parameters.

An adaptive learning rate relies on local value as compare to the traditional cyclical learning rate method which relies on global value. Various adaptive learning methods

have been proposed in the literature, in the context of this Duchi et al. [7] proposed an Adagrad method which is one of the earliest adaptive methods that estimate the learning rate from gradients.

Zeiler [8] proposed a new approach which is an extension of Adagrad called Adadelta. The proposed approach decreases the learning rate monotonically. Instead of accumulating all past gradients, the algorithm restricts the accumulation to a fixed size window.

Tieleman and Hinton [9] describe the RMSProp method which is a base for all the fundamental adaptive learning rate method. The proposed method divides the learning rate for weight by a running average of magnitudes of recent magnitudes for that weight.

Adaptive Moment Estimation by Kingma and Ba [10] is another method for computing learning rate the algorithm uses first-order gradient-based optimization of objective functions based on the estimation of lower order moments. The algorithm is computationally efficient and requires little memory and performs well over problems having large data and parameters.

Nesterov accelerated Adaptive Momentum estimation [11] is a combination of Adam and Nesterov accelerated gradient (NAG) [12]. The proposed algorithm modifies Adam momentum components by taking advantage of NAG, the proposed strategy improves the speed of convergence and the quality of learning without increasing the complexity.

Tiwari et al. [13] address the problem of enhancing mammograms for increasing the visual quality and delectability of anomalies present in the breasts. A nonlinear logistic function is used for pre-processing of mammograms. The proposed method improves the pixel intensity in the only anomalous region and it reduces the necessity of segmentation of mammograms.

3 Neural Network Hyperparameters and Its Effect

Hyper-parameter is the variable that helps in determining the network structure. In deep learning, a learning algorithm can be defined as a function taking training set as input and produces an output as function or a set of functions. The performance of these algorithms depends on various attributes called hyperparameters. These hyperparameters are set prior to the actual implementation of the algorithm and can be controlled by another algorithm called Hyper-learner. In general, these hyperparameters are categories into two categories, i.e. hyper-parameter related to the training algorithm and hyper-parameter related to network structure defined in Table 1. Choosing the values of these parameters is as important as the selection of a model for the given dataset.

Table 1 List of hyperparameters

Name of hyper-parameter	Definition	Related to	
		Training algorithm	Network structure
Initial learning rate	$\epsilon_t = \epsilon_0 * \tau / \max(t, \tau)$ (1) where ϵ_0 represents the learning rate	✓	
Learning rate schedule	In Eq. 1 τ represents the value	✓	
Batch size	Number of training examples taken in one forward and backward	✓	
Number of training iterations (epochs)	Iteration is number of epoch over the entire data set	✓	
Momentum	This parameter helps accelerated gradients vectors in right direction which leads to faster convergence	✓	
Decay	This parameter reduces the learning rate after each step in an epoch	✓	
Number of steps per epoch	These are the number of steps taken in an epoch for training the network	✓	
Number of neurons in hidden layers	Neuron or node is a computational unit that has one or more weighed input connection with activation function and an output connection		✓
Dropout regularization	Used to ignore the neurons not needed for performing computation		✓
Network weight initialization	These are the weights of neurons that are set prior to the start of computation		✓
Activation function	Used as a computational function by a computational unit called as neurons		✓

In deep learning, the algorithm performs optimally when it approximates the target function that maps input to output variables. The most important thing that is needed to be considered while learning the target function from the training data is how well the model generalizes to new data. The terminology that is used to define how well the model learns and generalize to new data is named as over-fitting and under-fitting. The over-fitting and under-fitting are the two biggest causes for the poor

performance of the model. The amount that the weights are updated during training is referred to as the step size or learning rate. During training the backpropagation of error estimates the amount of error for which the weights of the nodes in the network are responsible, so instead of updating the weight with full amount, we scaled it by learning rate. A perfect configured learning rate leads the model to best approximate function at given available resources.

4 Experiment and Result Analysis

To optimally determine the performance of a model over the plant village data set [14] and the synthetic data set, the selection of a specific learning algorithm and the adjustment of learning rates need to be required. The aim of the study is to learn the answers to the following questions: How large learning rate results in unstable training and failure of network? How adaptive learning rates can accelerate the training and alleviate the problem of keeping learning rate constant and can help in the optimization process?

To answer the above questions the Convolutional Neural network [15] model is applied over the different data set. The typical range of learning rate used for training the model varies in domain $1E-0$ to $1E-7$ [1]. In the study we observed that optimizer gives good performance in range $0.1-0.001$ over the model and the optimum value that can be used to optimize the model exist in range $0.1-0.001$.

In the experiment, the optimizer uses the decay/schedule_decay function defined in Eq. 2. This decreases the value of learning rate after each epoch. In all the methods the initial learning rate with a decay function are used as a parameter. Accept Nadam optimizer all optimizers have the same decay rate defined as:

$$\text{Decay} = \text{INIT_LR}/\text{EPOCHS} \quad (2)$$

The Nadam optimizer uses schedule decay rate as:

$$\text{schedule_decay} = \text{INIT_LR}/\text{EPOCH} \quad (3)$$

The performance of SGD, Adagrad, Adadelta, RMS Prop, Adamax and Nadam is tested over plant village data set and a synthetic data set over the range of learning rate from 0.1 to 0.001.

In SGD as shown in Fig. 1 at LR 0.01 the model performs over-fitting since the ratio of training to validation loss is large and at LR 0.001 the model performs under-fitting the performance of model is good near to LR 0.1. In Adagrad at LR 0.01

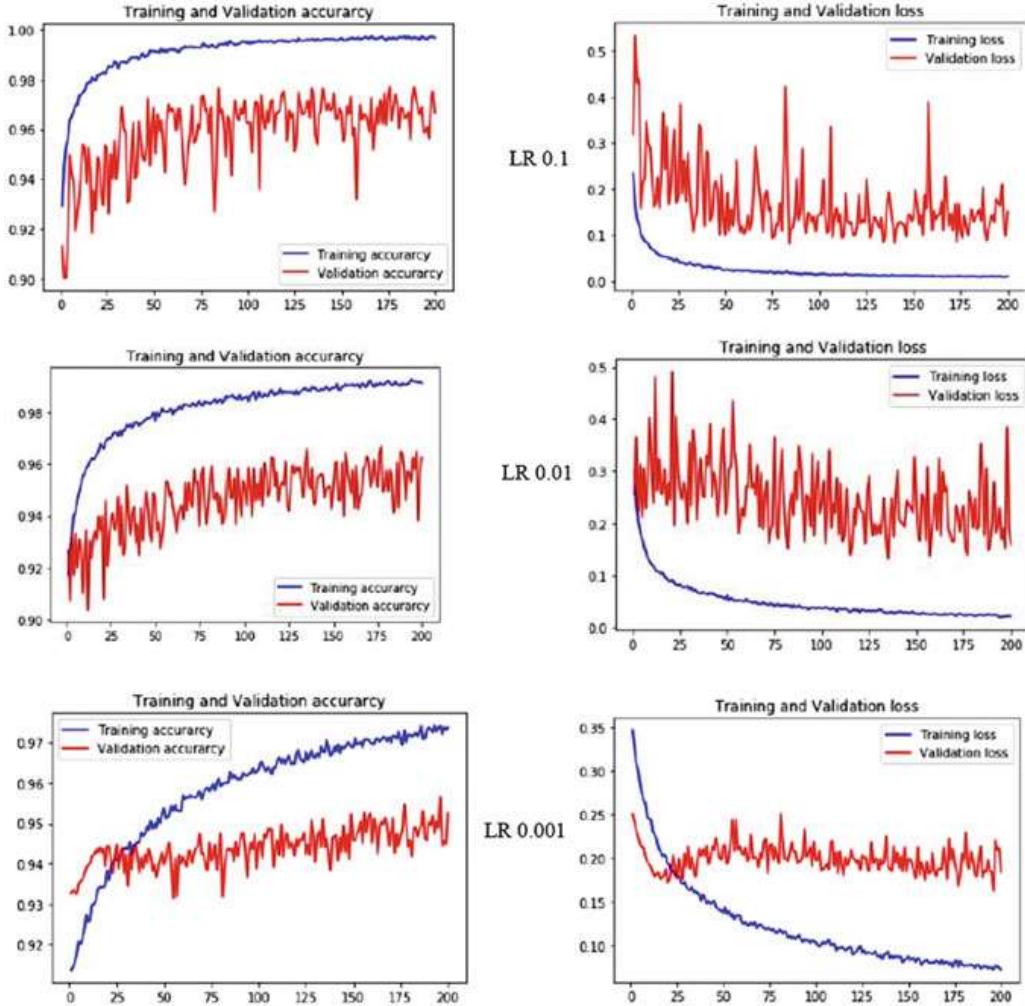


Fig. 1 Training—validation accuracy and loss of SGD at different LR

and 0.001 the model performs over-fitting at LR 0.1 the ratio between training and validation loss is almost negligible as shown in Fig. 2, here the model is perfectly fit. In Adadleta the model shows over-fitting as in Fig. 3 over the range of LR. In RMS Prop the performance of model is good at LR 0.1 to compare to 0.01 and 0.001. As shown in Fig. 4. Using Adam optimizer the model shows good behaviour, at LR 0.1 the behaviour of model is under-fitting since the training to validation loss is constant near to 2.0, at LR 0.01 and 0.001 the ratio to training to validation loss is almost negligible as shown in Fig. 5. Adamax shows a different behaviour at LR 0.1 as shown in Fig. 6 the model is under-fit with less noise without fluctuation, at LR 0.01 the model shows good behaviour but with noise whereas at LR 0.001 the model is over-fit, over the range of LR the accuracy is almost same. In Nadam as

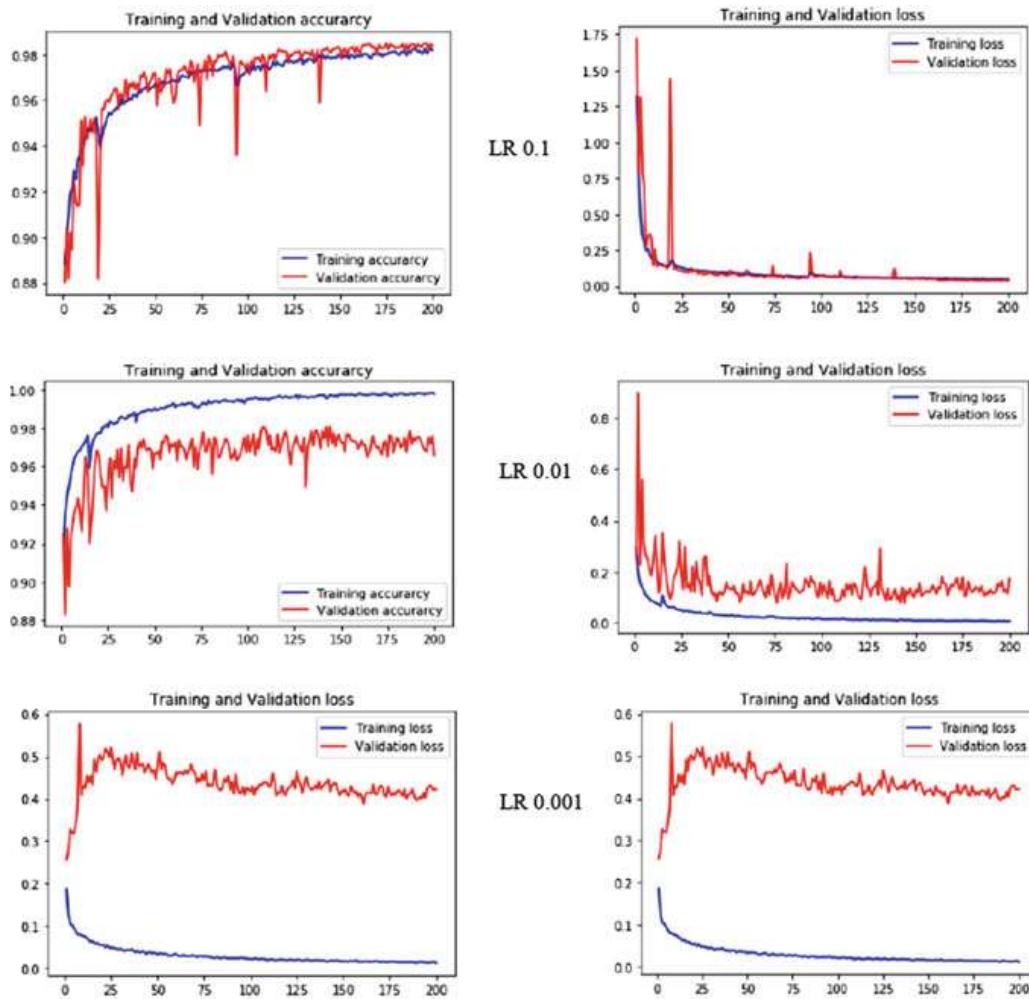


Fig. 2 Training—validation accuracy and loss of Adagrad at different LR

in Fig. 7 at LR 0.1 the model shows under-fitting behaviour since the training and validation loss is almost constant at all the epochs, at LR 0.01 the model shows best fit with accuracy near to 99% and at LR 0.001 the model is slightly over-fit with 99% accuracy. In another experiment over a synthetic data set the comparison of accuracy and loss calculation is shown in Fig. 8. The result shows that the Adagrad optimizer shows good performance with less loss.

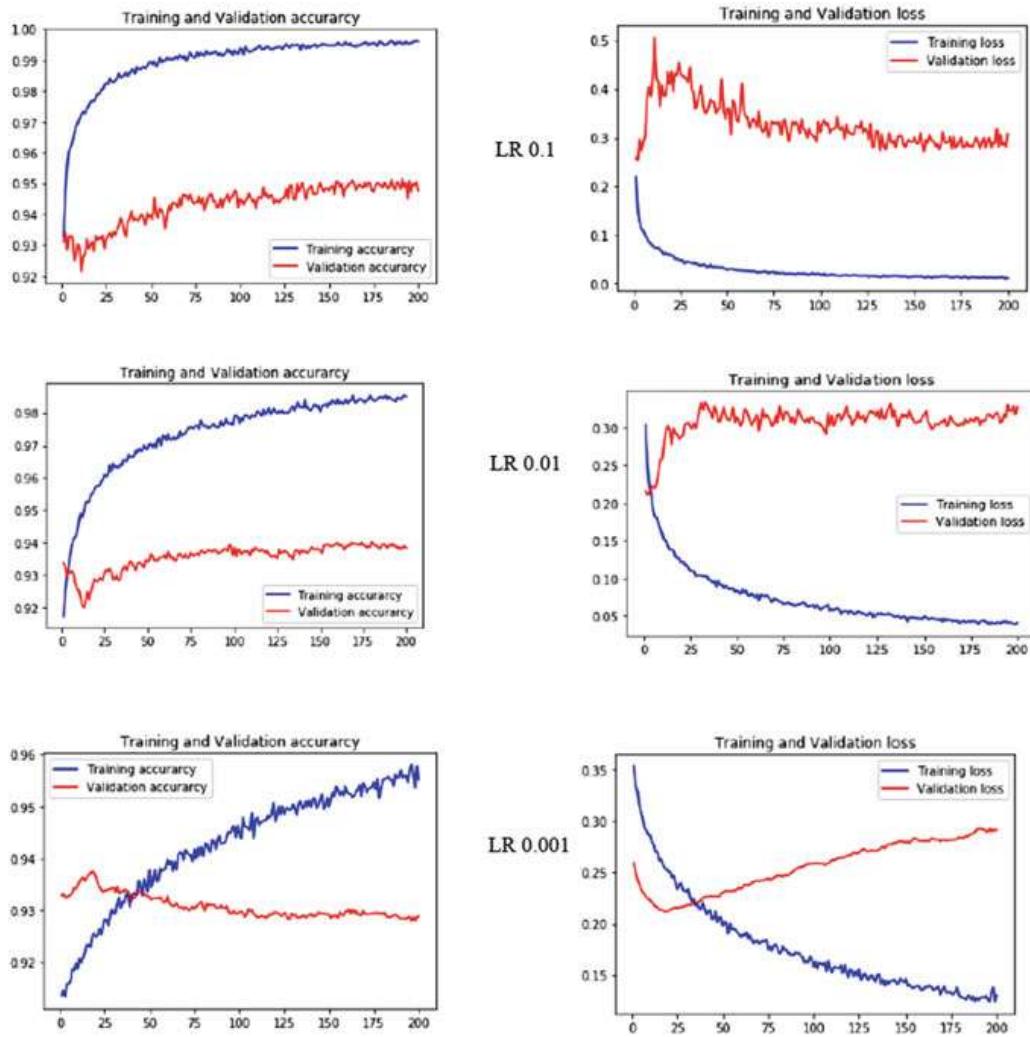


Fig. 3 Training—validation accuracy and loss of Adadelta at different LR

5 Conclusion

In this work, a comparative analysis of adaptive algorithms like SGD, Adagrad, Adadelta, RMSProp, Adam, Adamax and Nadam over a plant village data set has been done. The study discusses the effect of hyperparameters and their role in the performance of the model. How large learning rate results in unstable training and failure of network. How adaptive learning rates can accelerate the training and alleviate the

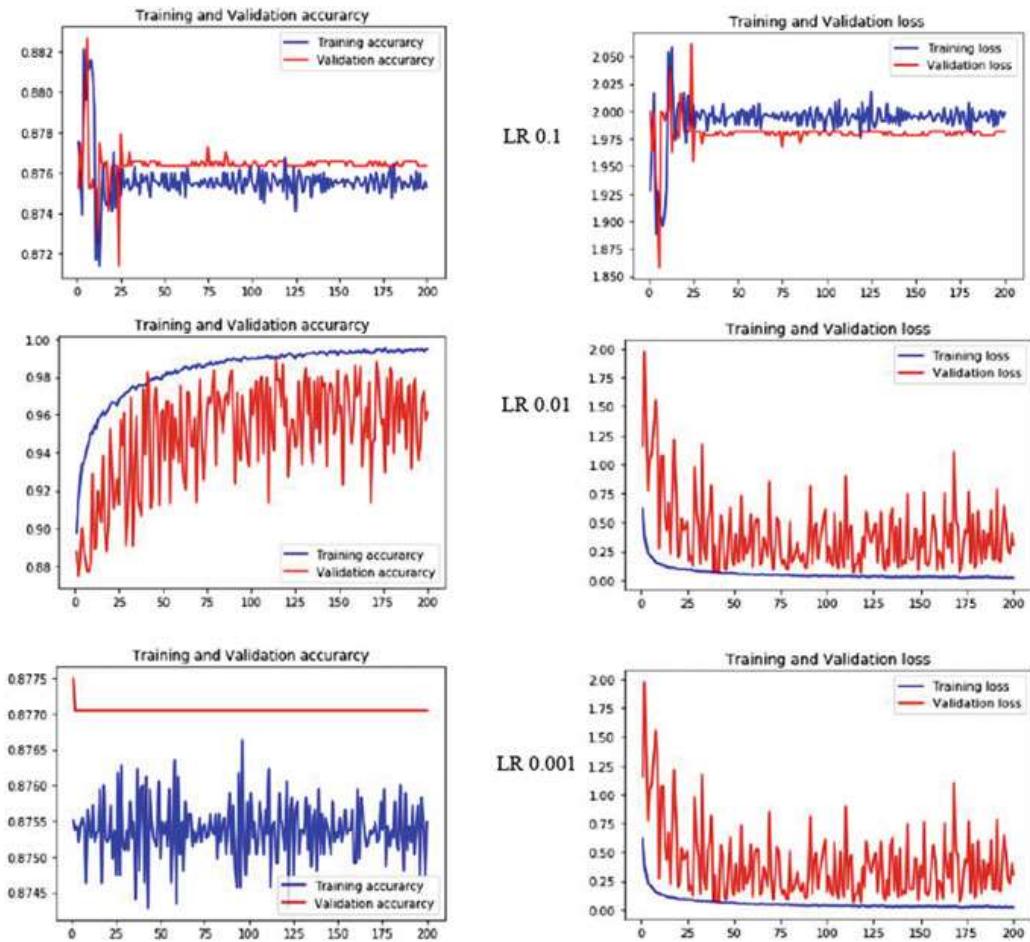


Fig. 4 Training—validation accuracy and loss of RMSProp at different LR

problem of keeping learning rate constant and can help in the optimization process. In the study, we observed that optimizers give good performance in range 0.1–0.001 over the model. Experimentally we observed that optimum value that can be used to optimize the model exist in range 0.1–0.001. The model performs well by using an Adam optimizer.

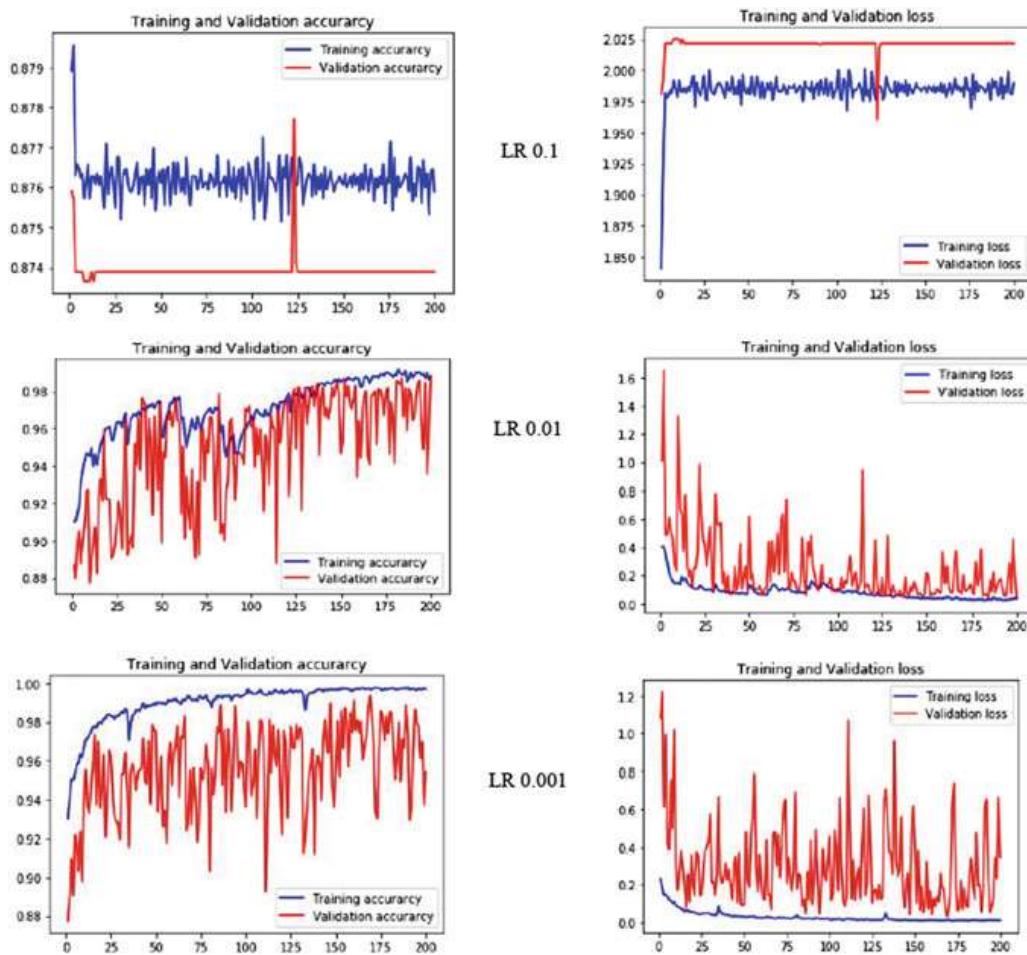


Fig. 5 Training—validation accuracy and loss of Adam at different LR

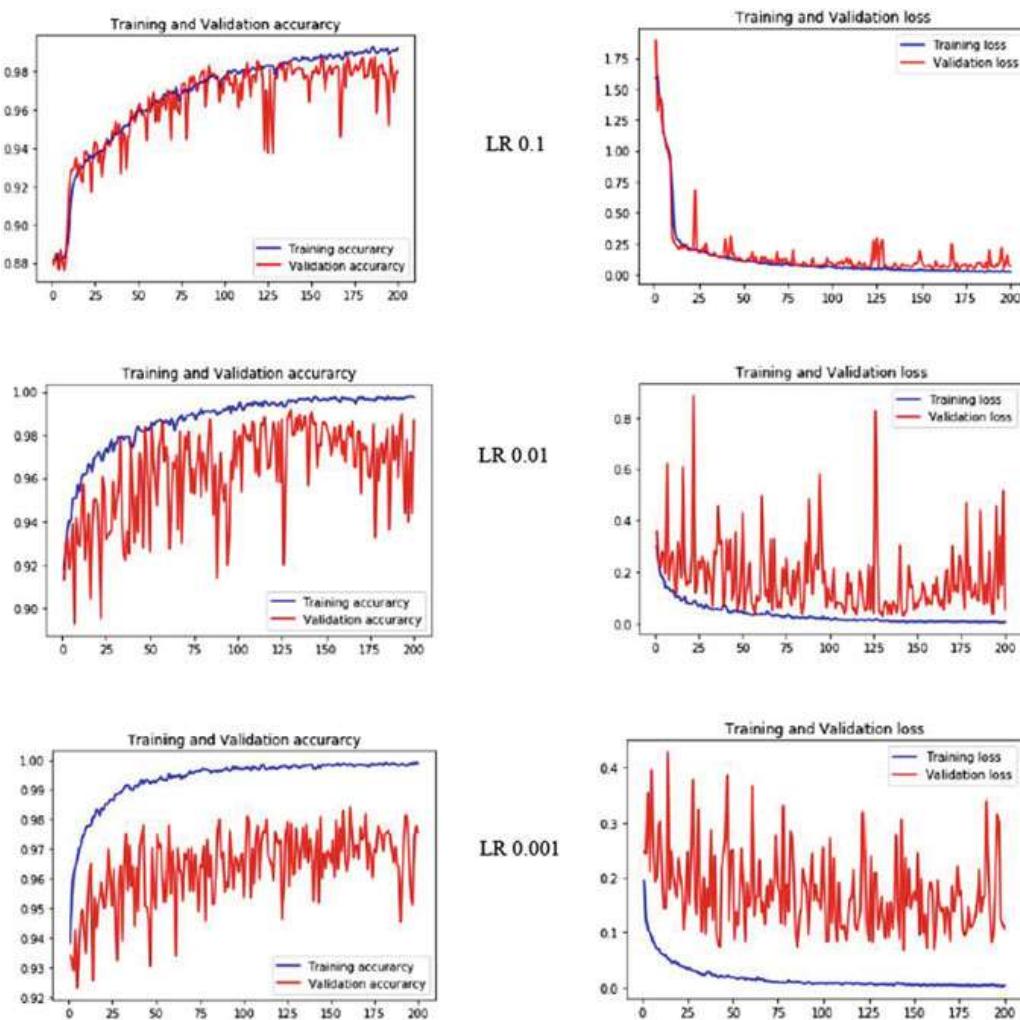


Fig. 6 Training—validation accuracy and loss of Adamax at different LR

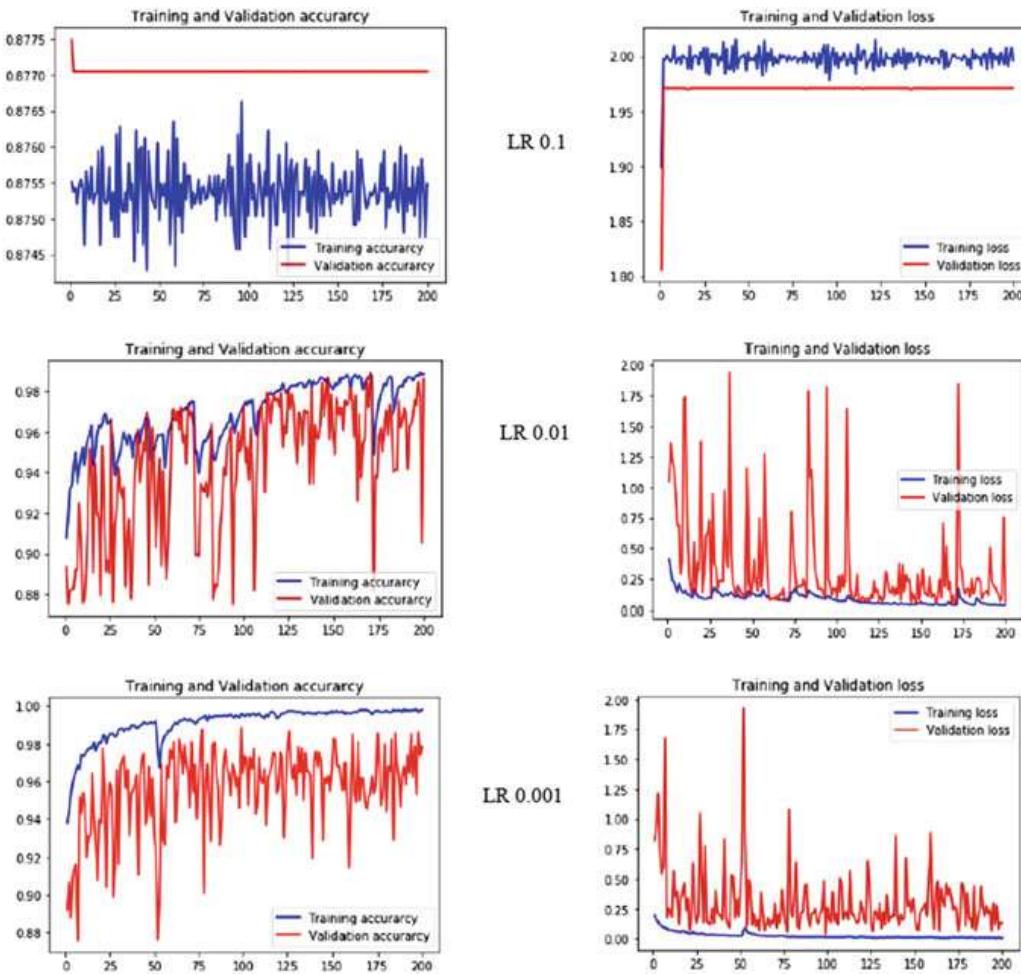


Fig. 7 Training—validation accuracy and loss of Nadam at different LR

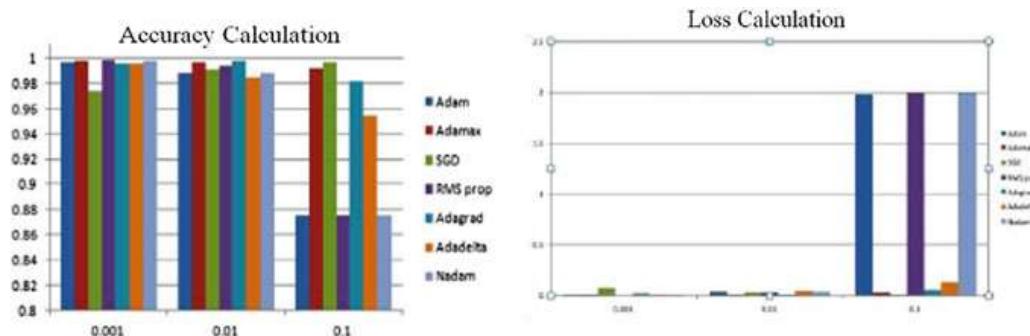


Fig. 8 Comparison of accuracy and loss calculation at different LR

References

1. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**, 281–305 (2012)
2. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: *Neural Networks: Tricks of the Trade*, pp. 437–478. Springer, Berlin, Heidelberg (2012)
3. Smith, L.N.: No more pesky learning rate guessing games. *CoRR*, abs/1506.01186 (2015)
4. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, pp. 464–472 (2017)
5. Smith, S.L., Kindermans, P.J., Ying, C., Le, Q.V.: Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489 (2017)
6. Lorraine, J., Duvenaud, D.: Stochastic hyperparameter optimization through hypernetworks. arXiv preprint arXiv:1802.09419 (2018)
7. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **12**, 2121–2159 (2011)
8. Zeiler, M. D.: ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
9. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **4**(2), 26–31 (2012)
10. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations, pp. 1–13 (2015)
11. Dozat, T.: Incorporating nesterov momentum into adam 2016
12. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
13. Tiwari, A., Bhateja, V., Gautam, A., Satapathy, S.C.: ANN-based classification of mammograms using nonlinear preprocessing. In: Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications, pp. 375–382. Springer, Singapore (2018)
14. <https://github.com/spMohanty/PlantVillage-Dataset>
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

Waste Management System: Approach with IoT, Prediction, and Dashboard



Viswanadhapalli Bhanuja, Ramai Varangaonkar, Yashveer Girdhar, and Kumar Kannan

Abstract After the mechanical transformation and development in innovation in recent decades, there has been a fast increment in the assembling ventures and its squanders as a result of which huge amounts of wastes are generated. These wastes contain harmful elements, gases, and toxic substances. The decomposition and degradation of certain wastes generate landfill harmful gases. The wastes and gases lead to soil, air, and water pollution. To manage these wastes in an effective way we propose an approach that can provide a way of monitoring the wastes and the gas levels and managing it by taking measures. The idea is to make use of certain sensors or cells that detect changes in the wastes and gas levels. Making use of concepts of IoT, machine learning, and graphical representations to provide information about the current and future level changes in the wastes and gases in the regions where sensors are located. In this paper, we are focusing on the levels of changes in gaseous wastes including landfill gases generated due to the wastes in the various regions. The prediction results of the gas levels can help in taking preventive and precautionary measures for proper management and disposal of these wastes.

Keywords Waste management · Prediction · IoT · Technical analyst · Dashboard

V. Bhanuja · R. Varangaonkar (✉) · Y. Girdhar · K. Kannan

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: ramai.varangaonkar@gmail.com

V. Bhanuja
e-mail: viswanadhapallibhanuja@gmail.com

Y. Girdhar
e-mail: veeryash05@gmail.com

K. Kannan
e-mail: kkumar@gmail.com

1 Introduction

After the industrial revolution and evolution in technology over the past few decades, there has been a rapid increase in the manufacturing industries, thus producing a huge number of various products. Along with useful and good quality products, a lot of solid, liquid, as well as gaseous wastes are generated. Emission and disposal of these directly into the environment adds up to the ever-rising greenhouse effects, global warming problems, and also other environmental problems. To keep a check on the statistics of all such harmful wastes, we propose a framework.

The framework consists of the design where we collect data about the wastes, perform some analysis that predicts the values in the percentages and levels of certain harmful substances and gases and displays it on a dashboard. We make use of certain electrochemical sensors [1], semiconductor metal oxide sensors [2] or cells that detect changes in the levels of the wastes, the data collected is stored on the cloud. A set of predictive analysis algorithms are applied to the collected datasets to give information about the current as well as future level changes that might take place in these toxic wastes. All the statistics and analysis [3] regarding the waste-related data will be displayed on the dashboard in the form of graphs and pie charts [4]. This data can be used by certain government organizations, environmental activists, and other waste management organizations to implement certain preventive and precautionary measures.

2 Survey of Existing Work

Waste management has become an important factor in our ecosystem. We have searched our papers by using certain keywords like IoT sensors, detecting gases, Dashboards, Visualization, Prediction analysis, and recommendations. For detecting the different gases and monitoring the temperature and levels of pH, different devices/sensors are used like Electronic Nose, IoT (UVI-01) and RFID device, semiconductor metal oxide sensors [2, 5–7]. For prediction analysis and data analytics and also for effective refurbishment, the EMARP algorithm is used. And also for efficient analysis, there are different algorithms like Neural Networks which are used to identify efficient dimensions, i.e., cost, time, quality [3, 4]. The definition of dashboard might be depending on performance and visual effect [8] and also it must be flexible and user-friendly [9]. For visual management, LP Steen Kamp, and the team used the Automation Pyramid tool [10]. Hence in our project, we're going to use the concepts of IoT sensors, predictive analytics algorithms, and dashboard techniques to display the analysis of data related to all sorts of wastes.

3 Proposed System Design

In the proposed idea, we have sensors located in N different regions. The sensors are classified into sets such as sensors detecting a particular gas/element that belongs to that particular sensor set. Every sensor set has its unique id and also every region has its reg_ID. As shown in Fig. 1. IoT-based approach various areas like Area 1, Area 2, and Area 3 where sensor sets are placed that detect various harmful gases.

The set of Arduino boards act as the middleware that gets the data from sensors and classifies and identifies the data based on the sensors. The data is sent on the cloud where it is stored based on region and sensor's respective ids. The dashboard is a graphical user interface that simplifies the complex data. In this dashboard phase, we are displaying the data in different patterns. For displaying data we are retrieving the data from the cloud. For that data, we are performing different display operations,

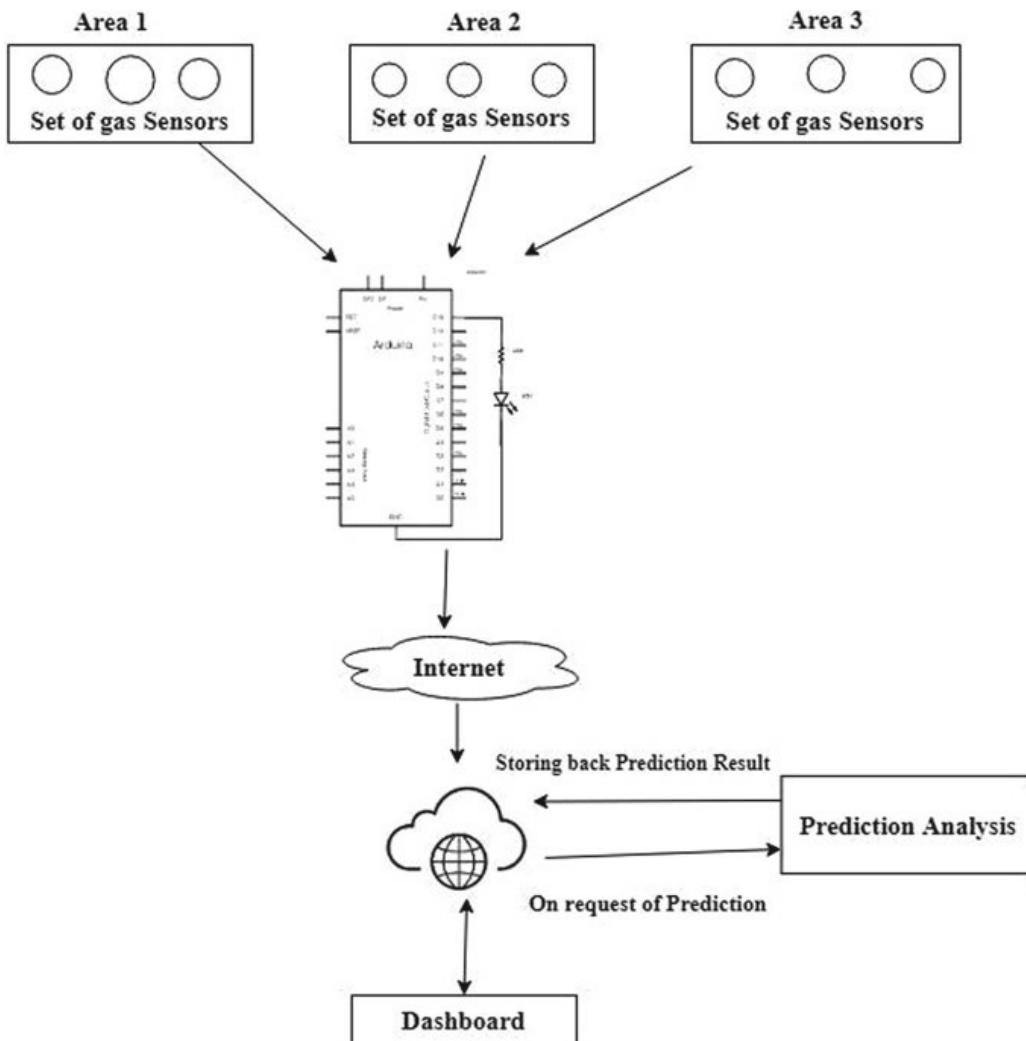


Fig. 1 IoT-based approach

i.e., line-chart, pie-chart, radial-chart, and so on. When a prediction request is made on the dashboard, the prediction analysis is applied to the respective datasets on the cloud and the data is stored back. The dashboard retrieves this data and displays it. This data analysis can be used to incorporate effective steps and manage the wastes-causing the gases in the respective regions.

3.1 Functional Flow of the IoT Approach

The steps to be followed in the functional flow diagram are (Fig. 2):

1. Sensors collect the data.
2. Arduino board acting as a middleware store the information on the cloud, data classification data based on sensor ids.
3. Data processing is done on the cloud. The cloud server continues checking whether the information is in the necessary format. At the point when the information is ready, it stores on the cloud.
4. The home screen is shown to the client as an interface. The client can make demands for information with respect to level changes of different gases and furthermore for forecasts about the gas levels.
5. On-demand from the client, the information will be shown. At the point when a prediction is made, in the event that it is for simply showing the level changes in the gases as for time, amount of wastes, and other such parameters, then the information is sent legitimately on the dashboard and showed in graphical portrayals.
6. In the event that the prediction is accomplished for forecast, at that point, the datasets are given to analysis model.
7. Then the predictive analysis applied on datasets.
8. Then the result will be displayed on the dashboard.

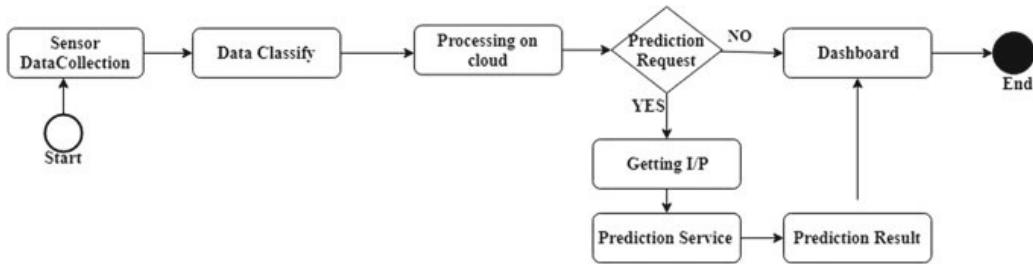


Fig. 2 Functional flow diagram

4 Implementation and Result

As discussed in the previous section, this work is addressing an IoT prediction and dashboard making the whole framework modular and compact to different environments.

The particular components used to execute the modules are given below.

4.1 Sensor Module as Input Phase

Various landfill gases are released in the environment on/after the waste disposal. The sensors sets in various regions detect values of gas levels. In this system, we can use various sensors like ionization detectors used for detecting hydrocarbons (HC) such as acetylene (C_2H_2), ethane (C_2H_6), Methane (CH_4), etc. Suppose there are 10 sensors in the sensor set of Methane detectors, each sensor gives a value for the same gas. That means sensors $id_1, id_2, \dots, id_{10}$ denoting ionization detectors detect values v_1, v_2, v_{10} . These are sent on the cloud, to store values of the gases. The values stored in the cloud in a schema with attributes as weights/amounts of wastes dumped, gas levels, time, date, etc.

In this state diagram, at first sensors are in idle state, on an hourly basis intervals the sensor data is collected, and the state of the sensor changes to Active state. Inactive state continuous detection and sensing of the gas levels in the environment are done. The data collected is sent to the cloud with the help of the Arduino board. Arduino board classifies the data in Type of sensors and stores it into the cloud. In some cases, if the gas levels cross the threshold level which is set in the sensor, the data is sent directly to the cloud (Fig. 3).

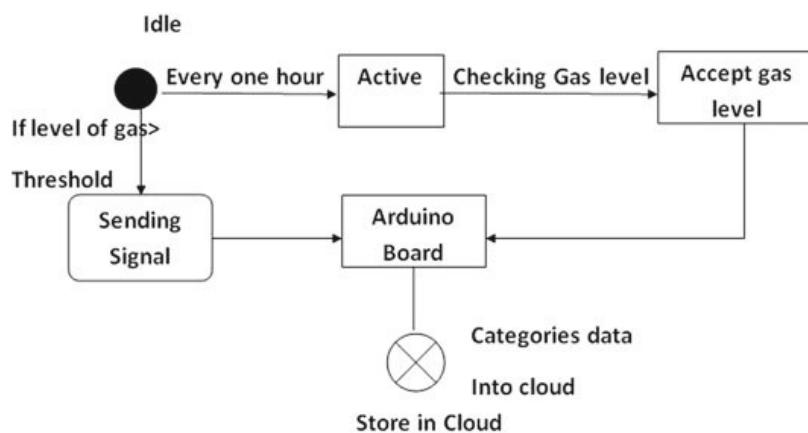


Fig. 3 Working model of sensors

4.2 Prediction Analysis Phase

In the whole process, a lot of data will be collected and stored on the cloud, which can be used to make predictions about levels of several gases. Predictive analysis uses statistical techniques from data mining, predictive modeling and machine learning that analyze the current and historical facts to make predictions let's consider one of the regions where our setup is established, for example, Area 1. Various sensor sets are placed that detect the changes in values of gases. To make predictions regarding the Methane gas levels, analysis is applied to the cloud data. The dependency on Methane gas and some fixed parameters like weight. Preprocessing is done on this data. The data obtained is analyzed for patterns and trends. In our Methane gas example, the dataset shows a dependency on the weights of wastes with the Methane gas. The prediction algorithm that best suits this case is regression. In our pre-processed dataset available, the modeling can be done to identify what sort of dependencies are present. We get that there is a linear dependency between the weight of wastes dumped (kgs) and gas released. In the linear regression model, a line is found that most closely fits the data according to the specific mathematical criterion.

In our Methane gas case from the graph we realize that a linear line can be fitted the equation can be shown as:

$$MG = a + b * (W) \quad (1)$$

MG—Methane Gas levels, W—Weight of wastes dumped, a —y intercept, b —slope of line. A generalized equation for any gas can be written as:

$$G = a + b * (W) \quad (2)$$

The weight (W) is an independent variable while the Methane gas levels (MG) depend on it. Thus depending on the trends and patterns in our dataset, the analysis is done and the regression technique is decided. The modeling contains data training thus finding the values of unknowns a and b of the equation. The values of a and b can be found by:

$$a = \frac{(\sum G) * (\sum (W)^2) - ((\sum W)^* (\sum (W * G)))}{n * (\sum (W^2)) - (\sum W)^2} \quad (3)$$

$$b = \frac{n * (\sum (G * W)) - ((\sum W) * (\sum G))}{n * (\sum (W^2)) - (\sum W)^2} \quad (4)$$

where G is generalized for all gas levels, W is weight of wastes dumped, n is the number of datasets. Thus calculating a and b we get the line equation which is used to predict a gas level value given the weight of wastes dumped in the region. The predicted values are sent to the dashboard for display.