

Amit Joshi
Mahdi Khosravy
Neeraj Gupta *Editors*

Machine Learning for Predictive Analysis

Proceedings of ICTIS 2020

Lecture Notes in Networks and Systems

Volume 141

Series Editor

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

Advisory Editors

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

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

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

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

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

Imre J. Rudas, Óbuda University, Budapest, Hungary

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

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

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

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

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

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

More information about this series at <http://www.springer.com/series/15179>

Amit Joshi · Mahdi Khosravy · Neeraj Gupta
Editors

Machine Learning for Predictive Analysis

Proceedings of ICTIS 2020



Springer

Editors

Amit Joshi
Global Knowledge Research Foundation
Gujarat, India

Mahdi Khosravy
University of Osaka
Suita, Japan

Neeraj Gupta
School of Engineering
and Computer Science
Oakland University
Rochester Hills, MI, USA

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-981-15-7105-3

ISBN 978-981-15-7106-0 (eBook)

<https://doi.org/10.1007/978-981-15-7106-0>

© 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

Committees

Technical Program Committee Chairs

Tomonobu Senju, Editorial Advisory Board, Renewable Energy Focus, University of the Ryukyus, Okinawa, Japan

Thinagaran Perumal, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang, Malaysia

Dr. Parikshit Mahalle, Sinhgad Group, Pune, India

Dr. Nilanjan Dey, Techno India Institute of Technology, Kolkata, India

Dr. Nilesh Modi, Chairman, Professor, Dr. Babasaheb Ambedkar University, Ahmedabad, India

Technical Program Committee Members

Dr. Aynur Unal, Stanford University, USA

Prof. Brent Waters, University of Texas, Austin, TX, USA

Er. kalpana Jain, CTAE, Udaipur, India

Prof. (Dr.) Avdesh Sharma, Jodhpur, India

Er. Nilay Mathur, Director, NIIT Udaipur, India

Prof. Philip Yang, Price water house Coopers, Beijing, China

Mr. Jeril Kuriakose, Manipal University, Jaipur, India

Prof. R. K. Bayal, Rajasthan Technical University, Kota, Rajasthan, India

Prof. Martin Everett, University of Manchester, England

Prof. Feng Jiang, Harbin Institute of Technology, China

Dr. Savita Gandhi, Professor, Gujarat University, Ahmedabad, India

Prof. Xiaoyi Yu, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Prof. Gengshen Zhong, Jinan, Shandong, China

- Prof. Abdul Rajak A. R., Department of Electronics and Communication Engineering Birla Institute of Dr. Nitika Vats Doohan, Indore, India
Dr. Harshal Arolkar, Immd. Past Chairman, CSI Ahmedabad Chapter, India
Mr. Bhavesh Joshi, Advent College, Udaipur, India
Prof. K. C. Roy, Principal, Kautilya Institute of Technology and Engineering, Jaipur, India
Dr. Mukesh Shrimali, Pacific University, Udaipur, India
Mrs. Meenakshi Tripathi, MNIT, Jaipur, India
Prof. S. N. Tazi, Government Engineering College, Ajmer, Rajasthan, India
Shuhong Gao, Mathematical Sciences, Clemson University, Clemson, SC, USA
Sanjam Garg, University of California, Los Angeles, CA, USA
Garani Georgia, University of North London, UK
Chiang Hung-Lung, China Medical University, Taichung, Taiwan
Kyeong Hur, Department of Computer Education, Gyeongin National University of Education, Incheon, Korea
Sudath Indrasinghe, School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool, England
Ushio Inoue, Department of Information and Communication Engineering, Engineering Tokyo Denki University, Tokyo, Japan
Dr. Stephen Intille, Associate Professor College of Computer and Information Science and Department of Health Sciences, Northeastern University, Boston, MA, USA
Dr. M. T. Islam, Institute of Space Science, Universiti Kebangsaan Malaysia, Selangor, Malaysia
Lillykutty Jacob, Professor, Department of Electronics and Communication Engineering, NIT, Calicut, Kerala, India
Dagmar Janacova, Tomas Bata University in Zlín, Faculty of Applied Informatics nám. T. G, Czech Republic, Europe
Jin-Woo Kim, Department of Electronics and Electrical Engineering, Korea University, Seoul, Korea
Muzafer Khan, Computer Sciences Department, COMSATS University, Pakistan
Jamal Akhtar Khan, Department of Computer Science College of Computer Engineering and Sciences, Salman Bin Abdulaziz University, Kingdom of Saudi Arabia
Kholaddi Kheir Eddine, University of Constantine, Algeria
Ajay Kshemkalyani, Department of Computer Science, University of Illinois, Chicago, IL, USA
Madhu Kumar, Associate Professor, Computer Engineering Department, Nanyang Technological University, Singapore
Rajendra Kumar Bharti, Assistant Professor, Kumaon Engineering College, Dwarahat, Uttarakhand, India
Prof. Murali Bhaskaran, Dhirajlal Gandhi College of Technology, Salem, Tamil Nadu, India
Prof. Komal Bhatia, YMCA University, Faridabad, Haryana, India

Prof. S. R. Biradar, Department of Information Science and Engineering, SDM College of Engineering and Technology, Dharwad, Karnataka, India
A. K. Chaturvedi, Department of Electrical Engineering, IIT Kanpur, India
Jitender Kumar Chhabra, NIT, Kurukshetra, Haryana, India
Pradeep Chouksey, Principal, TIT College, Bhopal, Madhya Pradesh, India
Chhaya Dalela, Associate Professor, JSSATE, Noida, Uttar Pradesh, India
Jayanti Dansana, KIIT University, Bhubaneswar, Odisha, India
Soura Dasgupta, Department of TCE, SRM University, Chennai, India
Dr. Apurva A. Desai, Veer Narmad South Gujarat University, Surat, India
Dr. Sushil Kumar, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India
Amioy Kumar, Biometrics Research Lab, Department of Electrical Engineering, IIT Delhi, India
Prof. L. C. Bishnoi, GPC, Kota, India
Dr. Vikrant Bhateja, Lucknow, India
Dr. Satyen Parikh, Dean, Ganpat University, Ahmedabad, India
Dr. Puspendra Singh, JKLU, Jaipur, India
Dr. Aditya Patel, Ahmedabad University, Gujarat, India
Mr. Ajay Choudhary, IIT Roorkee, India
Prashant Panse, Associate Professor, Medi-Caps University, India
Roshani Raut, Associate Professor, Vishwakarma Institute of Information Technology, Pune, India
Rachit Adhvaryu, Assistant Professor, Marwadi University, India
Purnima Shah, Assistant professor, Adani Institute of Infrastructure and Engineering, India
Mohd. Saifuzzaman, Lecturer, Daffodil International University, Bangladesh
Nandakumar Iyengar, Professor, City Engineering College, India
Nilu Singh, Assistant Professor, SoCA, BBD University, Lucknow, India
Habib Hadj-Mabrouk, Researcher in Artificial Intelligence and Railway Safety, University Gustave Eiffel, France
Ankita Kanajiya, Assistant Professor, GLS University, India
Ripal Ranpara, Assistant Professor and DBA, Atmiya University, India
Santhosh John, Institute Director, Middle East Institute for Advanced Training, Oman
Nageswara Rao Moparthi, Professor, K L University, Andhra Pradesh, India
Akhilesh Sharma, Associate Professor, Manipal University, Jaipur, India
Sylwia Werbińska-Wojciechowska, Associate Professor, Wroclaw University of Science and Technology, Poland
Divya Srivastava, Ph.D. Scholar, Indian Institute of Technology Jodhpur, India
Dr. Debajyoti Mukhopadhyay, Professor, WIDiCoReL Research Lab, India
Dr. Neelam Chaplot, Associate Professor, Poornima College of Engineering, India
Shruti Suman, Associate Professor, K L University, Andhra Pradesh, India
M. V. V. Prasad Kantipudi, Associate Professor, Sreyas Institute of Engineering and Technology, Hyderabad, India
Urmila Shravankar, Professor, G H Raisoni College of Engineering, Nagpur, India

Dr. Pradeep Laxkar, Associate Professor, Mandsaur University, India
Muhammad Asif Khan, Researcher, Qatar University, Qatar
Gayatri Doctor, Associate Professor, CEPT University, India
Jagadeesha Bhat, Associate Professor, St. Joseph Engineering College, Mangalore, India
Dr. Madhan Kumar Srinivasan, Associate Vice President, Accenture Technology, India
Ajay Vyas, Assistant Professor, Adani Institute of Infrastructure Engineering, India
Mounika Neelam, Assistant Professor, PSCMRCE, India
Prakash Samana, Assistant Professor, Gokula Krishna College of Engineering, India
Dr. Monika Jain, Professor and Head-ECE, I.T.S Engineering College, Greater Noida, India
Dr. K. Srujan Raju, Professor and Dean Student Welfare, CMR Technical Campus, Telangana, India
Dr. Narendrakumar Dasre, Associate Professor, Ramrao Adik Institute of Technology, India
Anand Nayyar, Professor, Researcher and Scientist, Duy Tan University, Vietnam
Sakshi Arora, Assistant Professor, SMVD University, India
Dr. Ajay Roy, Associate Professor, Lovely Professional University, India
Prasanth Vaidya, Associate Professor, Sanivarapu, India
Dr. Jeevanandam Jotheeswaran, Director, Amity University Online, India
John Moses C., Associate Professor, JNTUH, India
Vaithyasubramanian S., Assistant Professor, Sathyabama University, India
Ashish Revar, Assistant Professor, Symbiosis University of Applied Sciences, Indore, India

Organizing Chairs

Mr. Bharat Patel, Chairman, CEDB
Dr. Priyanka Sharma, Raksha Shakti University, Ahmedabad, India
Amit Joshi, Director—Global Knowledge Research Foundation

Organizing Secretary

Mr. Mihir Chauhan, Organizing Secretary, ICTIS 2020

Conference Secretary

Mr. Aman Barot, Conference Secretary, ICTIS 2020

Supporting Chairs

Dr. Vijay Singh Rathore, Professor and Head, JECRC, Jaipur, India

Dr. Nilanjan Dey, Techno India Institute of Technology, Kolkata, India

Dr. Nisarg Pathak, Swarnim Gujarat Sports University, Gandhinagar, India

Preface

This LNNS volume contains the papers presented specifically at the theme Machine Learning at the ICTIS 2020: Fourth International Conference on Information and Communication Technology for Intelligent Systems. The conference was held during May 15–16, 2020, organized on a digital platform ZOOM due to the pandemic COVID-19. The supporting partners were InterYIT IFIP and Knowledge Chamber of Commerce and Industry (KCCI).

This conference aimed at targeting the state-of-the-art as well as emerging topics pertaining to ICT and effective strategies for its implementation in engineering and intelligent applications. The objective of this international conference is to provide opportunities for the researchers, academicians, industry persons, and students to interact and exchange ideas, experience, and expertise in the current trend and strategies for information and communication technologies. Besides this, participants will also be enlightened about the vast avenues and current and emerging technological developments in the field of ICT in this era, and its applications will be thoroughly explored and discussed. The conference is anticipated to attract a large number of high-quality submissions and stimulate the cutting-edge research discussions among many academic pioneering researchers, scientists, industrial engineers, and students from all around the world and provide a forum to researchers; propose new technologies, share their experiences, and discuss future solutions for design infrastructure for ICT; provide a common platform for academic pioneering researchers, scientists, engineers, and students to share their views and achievements; enrich technocrats and academicians by presenting their innovative and constructive ideas; and focus on innovative issues at the international level by bringing together the experts from different countries. Research submissions in various advanced technology areas were received, and after a rigorous peer-review process with the help of the program committee members and external reviewers, 60 papers were accepted with an acceptance rate of 0.23 for this volume.

The conference featured many distinguished personalities like Mike Hinckey, Ph.D., University of Limerick, Ireland, President, International Federation of Information Processing; Bharat Patel, Honorary Secretary-General, Knowledge Chamber of Commerce and Industry, India; Aninda Bose, Senior Editor, Springer,

India; Mufti Mahmud, Ph.D., Nottingham Trent University, UK; Suresh Chandra Satapathy, Ph.D., Kalinga Institute of Industrial Technology, Bhubaneswar, India; Neeraj Gupta, Ph.D., School of Engineering and Computer Science, Oakland University, USA; and Nilanjan Dey, Ph.D., Techno India College of Technology, Kolkata, India. We are indebted to all our organizing partners for their immense support to make this virtual conference successfully possible. A total of 23 sessions were organized as a part of ICTIS 2020 including 22 technical and 1 inaugural session. Approximately 154 papers were presented in 22 technical sessions with high discussion insights. The total number of accepted submissions was 112 with a focal point on ICT and intelligent systems. Our sincere thanks to our Organizing Secretary, ICTIS 2020 Mihir Chauhan and Conference Secretary, ICTIS 2020 Aman Barot and the entire team of Global Knowledge Research Foundation and Conference committee for their hard work and support for the entire shift of ICTIS 2020 from physical to digital modes in these new normal times.

Gujarat, India
Suita, Japan
Rochester Hills, USA

Amit Joshi
Mahdi Khosravy
Neeraj Gupta

Contents

| | |
|--|----|
| A Hybrid Deep Learning Approach for Stock Price Prediction | 1 |
| Abhishek Dutta, Gopu Pooja, Neeraj Jain, Rama Ranjan Panda, and Naresh Kumar Nagwani | |
| Detection of Alphanumeric Characters by Connectionist Temporal Classification with Vanilla Beam Search Algorithm and NLP Using MATLAB and Keras | 11 |
| Aseem Patil | |
| Multilabel Toxic Comment Classification Using Supervised Machine Learning Algorithms | 23 |
| Darshin Kalpesh Shah, Meet Ashok Sanghvi, Raj Paresh Mehta, Prasham Sanjay Shah, and Artika Singh | |
| Model of Speed Spheroidization of Metals and Alloys Based on Multiprocessor Computing Complexes | 33 |
| Gennady Shvachych, Boris Moroz, Andrii Martynenko, Iryna Hulina, Volodymyr Busygin, and Dmytro Moroz | |
| Prediction of Sales Using Stacking Classifier | 43 |
| Rajni Jindal, Isha Jain, Isha Saxena, and Manish Kumar Chaurasia | |
| How to Use LDA Model to Analyze Patent Information? Taking Ships Integrated Power System as an Example | 51 |
| Danyang Li and Xnlai Li | |
| A Comparison Study on Various Continuous Integration Tools in Software Development | 65 |
| Sheeba, Ganeshayya Shidaganti, and Ankur P. Gosar | |
| A Holistic Study on Approaches to Prevent Sexual Harassment on Twitter | 77 |
| Aishwariya Rao Nagar, Meghana R. Bhat, K. Sneha Priya, and K. Rajeshwari | |

| | |
|--|-----|
| Cyber Bullying Detection Based on Twitter Dataset | 87 |
| Debajyoti Mukhopadhyay, Kirti Mishra, Kriti Mishra, and Laxmi Tiwari | |
| Soil pH Prediction Using Machine Learning Classifiers and Color Spaces | 95 |
| Tejas Wani, Neha Dhas, Sanskruti Sasane, Kalpesh Nikam, and Deepa Abin | |
| A Food Recommendation System Based on BMI, BMR, k-NN Algorithm, and a BPNN | 107 |
| Anilkumar Kothalil Gopalakrishnan | |
| Complexity Reduced Bi-channel CNN for Image Classification | 119 |
| Nivea Kesav and M. G. Jibukumar | |
| An Approach to Mitigate the Risk of Customer Churn Using Machine Learning Algorithms | 133 |
| Debajyoti Mukhopadhyay, Aarati Malusare, Anagha Nandanwar, and Shriya Sakshi | |
| Frequency Detection and Variation with Smart-Sensor Data Analysis Using Artificial Neural Network and Cloud Computing | 143 |
| Arabinda Rath, Dillip K. Mishra, S. Q. Baig, and Gayatri Devi | |
| Comprehensive Study of Fetal Monitoring Methods for Detection of Fetal Compromise | 153 |
| Vidya Sujit Kurtadikar and Himangi Milind Pande | |
| Enhanced Flower Pollination Algorithm for Task Scheduling in Cloud Computing Environment | 163 |
| Timea Bezdan, Miodrag Zivkovic, Milos Antonijevic, Tamara Zivkovic, and Nebojsa Bacanin | |
| Counterfeit Currency Detection Using Supervised Machine Learning Algorithms | 173 |
| R. K. Yadav, Pulkit Valecha, and Shaivya Paliwal | |
| Spam Mail Classification Using Ensemble and Non-Ensemble Machine Learning Algorithms | 179 |
| Khyati Agarwal, Prakhar Uniyal, Suryavanshi Virendrasingh, Sai Krishna, and Varun Dutt | |
| On the Desired Properties of Linear Feedback Shift Register (LFSR) Based High-Speed PN-Sequence-Generator | 191 |
| Le Cuong Nguyen, Vu Kien Tran, and Chi Quynh Le | |
| Syngas Assessment from Plastic Waste Using Artificial Neural Network—A Review | 203 |
| Maulik A. Modi and Tushar M. Patel | |

| | |
|---|-----|
| Applications of Data Mining in Predicting Stock Values | 209 |
| Aparna Raghunath and A. R. Abdul Rajak | |
| Smart Artificial Intelligent-Based Controller for Hydroponic: New Technique for Soilless Plantation | 217 |
| Anurag S. D. Rai, Reeta Pawar, Alpana Pandey, C. S. Rajeshwari, and Ashok Kumar Gwal | |
| Human Action Detection Using Deep Learning | 229 |
| S. Gowri, Syed Aarif Suhaib Qadri, Suvam Bhowal, and J. Jabez | |
| Automatic Detection of Leaf Disease Using CNN Algorithm | 237 |
| S. Nandhini, R. Suganya, K. Nandhana, S. Varsha, S. Deivalakshmi, and Senthil Kumar Thangavel | |
| Prediction of Emotional Condition Through Dialog Narratives Using Deep Learning Approach | 245 |
| SaiTeja Segu, Yaswanth Reddy Poreddy, and Kiran L. N. Eranki | |
| Software Requirements Classification and Prioritisation Using Machine Learning | 257 |
| Pratvina Talele and Rashmi Phalnikar | |
| Comparison of Hidden Markov Models and the FAST Algorithm for Feature-Aware Knowledge Tracing | 269 |
| Georg Gutjahr, Pantina Chandrashekar, M. Gowri Nair, Mithun Haridas, and Prema Nedungadi | |
| ABCADF: Deploy Artificially Bee Colony Algorithm for Model Transformation Cohesive with Fitness Function of Adaptive Dragonfly Algorithm | 277 |
| Pramod P. Jadhav and Shashank D. Joshi | |
| Performance Comparison of Markov Chain and LSTM Models for Spectrum Prediction in GSM Bands | 289 |
| Sandeep Bidwai, Shilpa Mayannavar, and Uday V. Wali | |
| Prediction of Network Attacks Using Connection Behavior | 299 |
| N. Aakaash, K. Akshaya Bala, Veerabrahmam Pranathi, and Meenakshi S. Arya | |
| Multi-Face Recognition Using CNN for Attendance System | 313 |
| Prasanth Vaidya Sanivarapu | |
| Simulating the Concept of Self-Driving Cars Using Deep-Q Learning | 321 |
| Akhilesh P. Patil, Pramod Sunagar, Karthik Ganesan, Biswajit Kumar, and Kartik Sethi | |

| | |
|--|-----|
| Dynamic Cloud Access Security Broker Using Artificial Intelligence | 335 |
| Debayan Bhattacharya, Adeep Biswas, S. Rajkumar, and Ramani Selvanambi | |
| A Comparative VHDL Implementation of Advanced Encryption Standard Algorithm on FPGA | 343 |
| Darshit Suratwala and Ganesh Rahate | |
| Document Recommendation for Medical Training Using Learning to Rank | 353 |
| Raghvendra Rao, Suyash Choubey, Georg Gutjahr, and Prema Nedungadi | |
| An Algorithmic Game Theory Approach for the Stable Coalition and Optimum Transmission Cost in D2D Communication | 363 |
| Mahima Chaudhary, Anjana Jain, and Shekhar Sharma | |
| A Study of Hybrid Approach for Face Recognition Using Student Database | 375 |
| Sarika Ashok Sovitkar and Seema S. Kawathekar | |
| Multi-objective Consensus Clustering Framework for Flight Search Recommendation | 385 |
| Sujoy Chatterjee, Nicolas Pasquier, Simon Nanty, and Maria A. Zuluaga | |
| Profiling JVM for AI Applications Using Deep Learning Libraries | 395 |
| Neha Kumari and Rajeev Kumar | |
| Offline Signature Recognition Using Deep Features | 405 |
| Kamlesh Kumari and Sanjeev Rana | |
| An Analysis and Comparative Study of Data Deduplication Scheme in Cloud Storage | 423 |
| Pronika and S. S. Tyagi | |
| Prediction of the Most Productive Crop in a Geographical Area Using Machine Learning | 433 |
| Atharva Karwande, Medha Wyawahare, Tejas Kolhe, Soham Kamble, Rushikesh Magar, and Laksh Maheshwari | |
| The Smart Set: A Study on the Factors that Affect the Adoption of Smart Home Technology | 443 |
| S. Shanthana Lakshmi and Deepak Gupta | |
| New Approach for Multimodal Biometric Recognition | 451 |
| S. Preetha and S. V. Sheela | |
| Survey on Object Detection, Distance Estimation and Navigation Systems for Blind People | 463 |
| Bela Shah, Smeet Shah, Purvesh Shah, and Aneri Shah | |

| | |
|---|------|
| Contents | xvii |
| Classroom to Industry: A Pathway of a Student to Be an Engineer | 473 |
| K. Rajeshwari, S. Preetha, and H. M. Anitha | |
| ML Suite: An Auto Machine Learning Tool | 483 |
| Nilesh M. Patil, Tanmay P. Rane, and Anmol A. Panjwani | |
| Survey and Gap Analysis on Event Prediction of English Unstructured Texts | 491 |
| Krishnanjan Bhattacharjee, S. ShivaKarthik, Swati Mehta, Ajai Kumar, Rushikesh Kothawade, Paritosh Katre, Piyush Dharkar, Neil Pillai, and Devika Verma | |
| Human Action Detection Using Deep Learning Techniques | 503 |
| Vedantham Ramachandran, Peddireddy Janaki Rani, and Kalavathi Alla | |
| Deep Learning Methods and Applications for Precision Agriculture | 515 |
| Nilay Ganatra and Atul Patel | |
| Object Detection with Convolutional Neural Networks | 529 |
| Sanskruti Patel and Atul Patel | |
| Autonomous Vehicle Simulation Using Deep Reinforcement Learning | 541 |
| Rishikesh Kadam, Vishakha Vidhani, Bhavika Valecha, Anushree Bane, and Nupur Giri | |
| Bitcoin Price Prediction Using Time Series Analysis and Machine Learning Techniques | 551 |
| Aman Gupta and Himanshu Nain | |
| The Detection of Diabetic Retinopathy in Human Eyes Using Convolution Neural Network (CNN) | 561 |
| Saloni Dhuru and Avinash Shrivastava | |
| Breast Cancer Classification Using Machine Learning Algorithms | 571 |
| Simran Sharma and Sachin Deshpande | |
| A Strategic Approach to Enrich Brand Through Artificial Intelligence | 579 |
| Pranav Desai | |
| Real Estate Price's Forecasting Through Predictive Modelling | 589 |
| Nitin Sharma, Yojna Arora, Priyanka Makkar, Vikas Sharma, and Hardik Gupta | |
| Bitcoin Cost Prediction Using Neural Networks | 599 |
| Nitin Sharma, Yojna Arora, and Priyanka Makkar | |

| | |
|---|------------|
| A Comprehensive Analysis for Validation of AVISAR Object-Oriented Testing Tool | 613 |
| Prashant Vats and Manju Mandot | |
| Author Index | 625 |

Editors and Contributors

About the Editors

Amit Joshi is currently the Director of Global Knowledge Research Foundation and also an Entrepreneur Researcher who has completed his Masters and research in the areas of cloud computing and cryptography in medical imaging. Dr. Joshi has an experience of around 10 years in academic and industry in prestigious organizations. Dr. Joshi is an active member of ACM, IEEE, CSI, AMIE, IACSIT, Singapore, IDES, ACEEE, NPA and many other professional societies. Currently, Dr. Joshi is the International Chair of InterYIT at International Federation of Information Processing (IFIP, Austria). He has presented and published more than 50 papers in national and international journals/conferences of IEEE and ACM. Dr. Joshi has also edited more than 40 books which are published by Springer, ACM and other reputed publishers. Dr. Joshi has also organized more than 50 national and international conferences and programs in association with ACM, Springer and IEEE to name a few across different countries including India, UK, Europe, USA, Canada, Thailand, Egypt and many more.

Mahdi Khosravy holds a B.Sc. in Electrical Engineering (bio-electric) from Sahand University of Technology, Tabriz, Iran; an M.Sc. in Biomedical Engineering (bio-electric) from Beheshti University of Medical Studies, Tehran, Iran; and a Ph.D. in the field of Information Technology from the University of the Ryukyus, Okinawa, Japan. He received an award from the Head of the University for his research excellence. In 2010, he joined the University of Information Science and Technology (UIST), Ohrid, Macedonia as an Assistant Professor, and since 2018, he has been a Visiting Associate Professor at the Electrical Engineering Department, Federal University of Juiz de Fora in Brazil, and the Electrical Department at the University of the Ryukyus, Okinawa, Japan. Since November 2019, Dr. Khosravy has been a researcher at Media-integrated Laboratories, University of Osaka, Japan. He is a member of IEEE.

Neeraj Gupta received his Diploma in Environmental and Pollution Control, Civil Engineering in 1999, Bachelor of Engineering (B.E.) in Electrical and Electronics Engineering in 2003, Master of Technology (M.Tech.) in Engineering Systems in 2006, and his Ph.D. in Electrical Engineering from the Indian Institute of Technology (IIT), Kanpur, India, in 2013. He was a Postdoctoral Fellow (Sr. Project Engineer) at the Indian Institute of Technology (IIT) Jodhpur, India, for one year (June 2012–May 2013), and was a member of the faculty at the System Engineering Department of the same institute from 2013 to 2014. He then became an Assistant Professor at the Department of Applied IT Machine Intelligence and Robotics at the University for Information Science and Technology, Ohrid, Macedonia, from 2014 to 2017. Currently, he is an Assistant Professor at the School of Engineering and Computer Science, Oakland University, USA.

Contributors

N. Aakaash SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India

A. R. Abdul Rajak Birla Institute of Technology Science Pilani, Dubai Campus, Dubai, United Arab Emirates

Deepa Abin Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

Khyati Agarwal Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi, Mandi, HP, India

K. Akshaya Bala SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India

Kalavathi Alla Information Technology Department, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India

H. M. Anitha B.M.S. College of Engineering, Bengaluru, India

Milos Antonijevic Singidunum University, Belgrade, Serbia

Yojna Arora Department of Computer Science & Engineering, Amity University, Gurugram, India

Meenakshi S. Arya SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India

Nebojsa Bacanin Singidunum University, Belgrade, Serbia

S. Q. Baig Department of Computer Science and Engineering, ABIT, Cuttack, Odisha, India

Anushree Bane Vivekanand Education Society's Institute of Technology, Mumbai, India

Timea Bezdan Singidunum University, Belgrade, Serbia

Meghana R. Bhat B.M.S. College of Engineering, Bengaluru, India

Krishnanjan Bhattacharjee Centre for Development of Advanced Computing (C-DAC), Pune, India

Debayan Bhattacharya Vellore Institute of Technology, Vellore, India

Suvam Bhowal Sathyabama Institute of Science and Technology, Chennai, India

Sandeep Bidwai KLE Dr M S Sheshgiri College of Engineering and Technology, Belagavi, Karnataka, India

Adeep Biswas Vellore Institute of Technology, Vellore, India

Volodymyr Busygin Oles Honchar Dnipro National University, Dnipro, Ukraine

Pantina Chandrashekhar Center for Research in Analytics and Technologies for Education (CREATE), Amritapuri, India

Sujoy Chatterjee Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France; Amadeus S.A.S, Sophia Antipolis, France

Mahima Chaudhary Shri Govindram Seksaria Institute of Technology and Science, Indore, India

Manish Kumar Chaurasia Delhi Technological University, Delhi, India

Suyash Choubey Department of Computer Science, Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India

Deepak Gupta Amrita School of Business, Amrita Vishwa Vidyapeetham, Coimbatore, India

S. Deivalakshmi National Institute of Technology, Tiruchirappalli, Tiruchirappalli, India

Pranav Desai Indukaka Ipcowala Institute of Management, Faculty of Management Studies, Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat, India

Sachin Deshpande Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India

Gayatri Devi Department of Computer Science and Engineering, ABIT, Cuttack, Odisha, India

Piyush Dharkar Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

Neha Dhas Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

Saloni Dhuru Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India

Varun Dutt Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi, Mandi, India

Abhishek Dutta Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

Kiran L. N. Eranki School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

Nilay Ganatra Faculty of Computer Science and Applications, Charotar University of Science & Technology, Changa, India

Karthik Ganesan Ramaiah Institute of Technology, Bangalore, Karnataka, India

Nupur Giri Vivekanand Education Society's Institute of Technology, Mumbai, India

Anilkumar Kothalil Gopalakrishnan Department of Computer Science, Vincent Mary School of Science and Technology, Assumption University of Thailand, Bangkok, Thailand

Ankur P. Gosar Intel Technologies, Bangalore, India

S. Gowri Sathyabama Institute of Science and Technology, Chennai, India

Aman Gupta Delhi Technological University, Delhi, India

Hardik Gupta Department of Computer Science & Engineering, Chandigarh University, Mohali, India

Georg Gutjahr Center for Research in Analytics and Technologies for Education (CREATE), Amritapuri, India;

AmritaCREATE, Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India

Ashok Kumar Gwal Rabindranath Tagore University, Raisen, MP, India

Mithun Haridas Center for Research in Analytics and Technologies for Education (CREATE), Amritapuri, India

Iryna Hulina University of Technology, Dnipro, Ukraine

J. Jabez Sathyabama Institute of Science and Technology, Chennai, India

Pramod P. Jadhav G H Raisoni Institute of Engineering and Technology, Wagholi, Pune, Maharashtra, India

Anjana Jain Shri Govindram Seksaria Institute of Technology and Science, Indore, India

Isha Jain Delhi Technological University, Delhi, India

Neeraj Jain Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

M. G. Jibukumar Cochin University of Science and Technology, Cochin, Kerala, India

Rajni Jindal Delhi Technological University, Delhi, India

Shashank D. Joshi Bharati Vidyapeeth Deemed to be University, Pune, Maharashtra, India

Rishikesh Kadam Vivekanand Education Society's Institute of Technology, Mumbai, India

Soham Kamble Vishwakarma Institute of Technology, Pune, Maharashtra, India

Atharva Karwande Vishwakarma Institute of Technology, Pune, Maharashtra, India

Paritosh Katre Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

Seema S. Kawathekar CSIT Dept, Dr. BAM University, Aurangabad, India

Nivea Kesav Cochin University of Science and Technology, Cochin, Kerala, India

Tejas Kolhe Vishwakarma Institute of Technology, Pune, Maharashtra, India

Rushikesh Kothawade Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

Sai Krishna Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi, Mandi, HP, India

Ajai Kumar Centre for Development of Advanced Computing (C-DAC), Pune, India

Biswajit Kumar Ramaiah Institute of Technology, Bangalore, Karnataka, India

Rajeev Kumar School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

Kamlesh Kumari Department of Computer Science & Engineering, M. M (Deemed to Be University), Mullana-Ambala, India

Neha Kumari School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

Vidya Sujit Kurtadikar School of Computer Engineering and Technology,
MITWPU, Pune, India

Chi Quynh Le Hanoi Open University, Hanoi, Vietnam

Danyang Li School of Information Management, Wuhan University, Wuhan,
China

Xinlai Li Research Center for Chinese Science Evaluation, Wuhan University,
Wuhan, China

Rushikesh Magar Vishwakarma Institute of Technology, Pune, Maharashtra,
India

Laksh Maheshwari Vishwakarma Institute of Technology, Pune, Maharashtra,
India

Priyanka Makkar Department of Computer Science & Engineering, Amity
University, Gurugram, India

Aarati Malusare Computer Science Department, Mumbai University, Mumbai,
Maharashtra, India

Manju Mandot J.R.N. Rajasthan Vidyapith, Udaipur, Rajasthan, India

Andrii Martynenko University of Technology, Dnipro, Ukraine

Shilpa Mayannavar S G Balekundri Institute of Technology, Belagavi,
Karnataka, India

Raj Paresh Mehta Mukesh Patel School of Technology Management &
Engineering, NMIMS, Mumbai, India

Swati Mehta Centre for Development of Advanced Computing (C-DAC), Pune,
India

Dillip K. Mishra Department of Computer Science and Engineering, ABIT,
Cuttack, Odisha, India

Kirti Mishra Computer Science Department, Mumbai University, Mumbai,
Maharashtra, India

Kriti Mishra Computer Science Department, Mumbai University, Mumbai,
Maharashtra, India

Maulik A. Modi Mechanical Engineering Department, KSV, Gandhinagar, India

Boris Moroz University of Technology, Dnipro, Ukraine

Dmytro Moroz Oles Honchar Dnipro National University, Dnipro, Ukraine

Debjayoti Mukhopadhyay Computer Science Department, Mumbai University, Mumbai, Maharashtra, India;
WIDiCoReL Research Lab, Mumbai, Maharashtra, India

Aishwariya Rao Nagar B.M.S. College of Engineering, Bengaluru, India

Naresh Kumar Nagwani Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

Himanshu Nain Delhi Technological University, Delhi, India

M. Gowri Nair Department of Mathematics, Amrita VishwaVidyapeetham, Amritapuri, India

Anagha Nandanwar Computer Science Department, Mumbai University, Mumbai, Maharashtra, India

K. Nandhana Thiagarajar College of Engineering, Madurai, Tamilnadu, India

S. Nandhini Thiagarajar College of Engineering, Madurai, Tamilnadu, India

Simon Nanty Amadeus S.A.S, Sophia Antipolis, France

Prema Nedungadi Center for Research in Analytics and Technologies for Education (CREATE), Amritapuri, India;

Department of Computer Science, Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India

Le Cuong Nguyen Electric Power University, Hanoi, Vietnam

Kalpesh Nikam Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

Shaivya Paliwal Delhi Technological University, New Delhi, India

Rama Ranjan Panda Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

Himangi Milind Pande School of Computer Engineering and Technology, MITWPU, Pune, India

Alpana Pandey Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, MP, India

Anmol A. Panjwani Department of Information Technology, Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai, India

Nicolas Pasquier Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France

Atul Patel Faculty of Computer Science and Applications, Charotar University of Science and Technology, Changa, India

Sanskriti Patel Faculty of Computer Science and Applications, Charotar University of Science and Technology, Changa, India

Tushar M. Patel Mechanical Engineering Department, KSV, Gandhinagar, India

Akhilesh P. Patil Ramaiah Institute of Technology, Bangalore, Karnataka, India

Aseem Patil In2things Automation Pvt. Ltd., Pune, India

Nilesh M. Patil Department of Information Technology, Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai, India

Reeta Pawar Department of Electrical and Electronics Engineering, Rabindranath Tagore University, Raisen, MP, India

Rashmi Phalnikar School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India

Neil Pillai Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

Gopu Pooja Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

Yaswanth Reddy Poreddy School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

Veerabrahmam Pranathi SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India

S. Preetha B.M.S. College of Engineering, Affiliated to VTU, Bengaluru, India

Pronika Manav Rachna International Institute of Research & Studies, Faridabad, Haryana, India

Syed Aarif Suhaib Qadri Sathyabama Institute of Science and Technology, Chennai, India

Aparna Raghunath Birla Institute of Technology Science Pilani, Dubai Campus, Dubai, United Arab Emirates

Ganesh Rahate Electronics and Telecommunication Engineering Department, Pimpri Chinchwad College of Engineering, Pune, India

Anurag S. D. Rai Department of Electrical and Electronics Engineering, Rabindranath Tagore University, Raisen, MP, India

C. S. Rajeshwari Department of Electrical and Electronics Engineering, National Institute of Technical Teachers Training and Research, Shyamla Hills, Bhopal, MP, India

K. Rajeshwari B.M.S. College of Engineering, Bengaluru, India

S. Rajkumar Vellore Institute of Technology, Vellore, India

Vedantham Ramachandran Information Technology Department, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India

Sanjeev Rana Department of Computer Science & Engineering, M. M (Deemed to Be University), Mullana-Ambala, India

Tanmay P. Rane Department of Information Technology, Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai, India

Peddireddy Janaki Rani Information Technology Department, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India

Raghvendra Rao Department of Computer Science, Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India

Arabinda Rath Department of Computer Science and Engineering, ABIT, Cuttack, Odisha, India

Shriya Sakshi Computer Science Department, Mumbai University, Mumbai, Maharashtra, India

Meet Ashok Sanghvi Mukesh Patel School of Technology Management & Engineering, NMIMS, Mumbai, India

Prasanth Vaidya Sanivarapu Department of Computer Science and Engineering, MLR Institute of Technology, Hyderabad, Telangana, India

Sanskriti Sasane Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

Isha Saxena Delhi Technological University, Delhi, India

SaiTeja Segu School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

Ramani Selvanambi Vellore Institute of Technology, Vellore, India

Kartik Sethi Ramaiah Institute of Technology, Bangalore, Karnataka, India

Aneri Shah The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India

Bela Shah The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India

Darshin Kalpesh Shah Mukesh Patel School of Technology Management & Engineering, NMIMS, Mumbai, India

Prasham Sanjay Shah Mukesh Patel School of Technology Management & Engineering, NMIMS, Mumbai, India

Purvesh Shah The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India

Smeet Shah The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India

S. Shanthana Lakshmi Amrita School of Business, Amrita Vishwa Vidyapeetham, Coimbatore, India

Nitin Sharma Department of Computer Science & Engineering, Amity University, Gurugram, India

Shekhar Sharma Shri Govindram Seksaria Institute of Technology and Science, Indore, India

Simran Sharma Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India

Vikas Sharma Department of Computer Science & Engineering, Chandigarh University, Mohali, India

Sheeba Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, Karnataka, India

S. V. Sheela B.M.S. College of Engineering, Affiliated to VTU, Bengaluru, India

Ganeshayya Shidaganti Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, Karnataka, India

S. ShivaKarthik Centre for Development of Advanced Computing (C-DAC), Pune, India

Avinash Shrivats Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India

Gennady Shvachych National Metallurgical Academy of Ukraine, Dnipro, Ukraine

Artika Singh Mukesh Patel School of Technology Management & Engineering, NMIMS, Mumbai, India

K. Sneha Priya B.M.S. College of Engineering, Bengaluru, India

Sarika Ashok Sovitkar CSIT Dept, Dr. BAM University, Aurangabad, India

R. Suganya Thiagarajar College of Engineering, Madurai, Tamilnadu, India

Pramod Sunagar Ramaiah Institute of Technology, Bangalore, Karnataka, India

Darshit Suratwala Electronics and Telecommunication Engineering Department, Pimpri Chinchwad College of Engineering, Pune, India

Pratvina Talele School of Computer Engineering and Technology,
Dr. Vishwanath Karad MIT World Peace University, Pune, India

Senthil Kumar Thangavel Amrita School of Engineering Coimbatore, Amrita
Vishwa Vidyapeetham, Coimbatore, India

Laxmi Tiwari Computer Science Department, Mumbai University, Mumbai,
Maharashtra, India

Vu Kien Tran Electric Power University, Hanoi, Vietnam

S. S. Tyagi Manav Rachna International Institute of Research & Studies,
Faridabad, Haryana, India

Prakhar Uniyal Applied Cognitive Science Laboratory, Indian Institute of
Technology Mandi, Mandi, HP, India

Bhavika Valecha Vivekanand Education Society's Institute of Technology,
Mumbai, India

Pulkit Valecha Delhi Technological University, New Delhi, India

S. Varsha Thiagarajar College of Engineering, Madurai, Tamilnadu, India

Prashant Vats AIMACT, Banasthali University, Vanasthali, Rajasthan, India

Devika Verma Vishwakarma Institute of Information Technology, Pune,
Maharashtra, India

Vishakha Vidhani Vivekanand Education Society's Institute of Technology,
Mumbai, India

Suryavanshi Virendrasingh Applied Cognitive Science Laboratory, Indian
Institute of Technology Mandi, Mandi, HP, India

Uday V. Wali S G Balekundri Institute of Technology, Belagavi, Karnataka, India

Tejas Wani Department of Computer Engineering, Pimpri Chinchwad College of
Engineering, Pune, India

Medha Wyawahare Vishwakarma Institute of Technology, Pune, Maharashtra,
India

R. K. Yadav Delhi Technological University, New Delhi, India

Miodrag Zivkovic Singidunum University, Belgrade, Serbia

Tamara Zivkovic School of Electrical Engineering, Belgrade, Serbia

Maria A. Zuluaga Amadeus S.A.S, Sophia Antipolis, France;
Data Science Department, EURECOM, Biot, France

A Hybrid Deep Learning Approach for Stock Price Prediction



**Abhishek Dutta, Gopu Pooja, Neeraj Jain, Rama Ranjan Panda,
and Naresh Kumar Nagwani**

Abstract Prediction of stock prices has been the primary objective of an investor. Any future decision taken by the investor directly depends on the stock prices associated with a company. This work presents a hybrid approach for the prediction of intra-day stock prices by considering both time-series and sentiment analysis. Furthermore, it focuses on long short-term memory (LSTM) architecture for the time-series analysis of stock prices and Valence Aware Dictionary and sEntiment Reasoner (VADER) for sentiment analysis. LSTM is a modified recurrent neural network (RNN) architecture. It is efficient at extracting patterns over sequential time-series data, where the data spans over long sequences and also overcomes the gradient vanishing problem of RNN. VADER is a lexicon and rule-based sentiment analysis tool attuned to sentiments expressed in social media and news articles. The results of both techniques are combined to forecast the intra-day stock movement and hence the model named as LSTM-VDR. The model is first of its kind, a combination of LSTM and VADER to predict stock prices. The dataset contains closing prices of the stock and recent news articles combined from various online sources. This approach, when applied on the stock prices of Bombay Stock Exchange (BSE) listed companies, has shown improvements in comparison to prior studies.

Keywords Time-series analysis · Sentiment analysis · Deep learning · LSTM · CNN · RNN · VADER · Web scraping · NLP

A. Dutta (✉) · G. Pooja · N. Jain · R. R. Panda · N. K. Nagwani
Department of Computer Science and Engineering,
National Institute of Technology, Raipur, India
e-mail: abhi97.dutta@gmail.com

G. Pooja
e-mail: poojagopu98@gmail.com

N. Jain
e-mail: neerajjain311@gmail.com

R. R. Panda
e-mail: rtpanda.phd2018.cs@nitrr.ac.in

N. K. Nagwani
e-mail: nknagwani.cs@nitrr.ac

1 Introduction

The stock market is volatile and dependent on various factors, such as socio-economic scenarios, political situations, and technological advancements. Also, the uncertainty involved in the prediction of intra-day stock prices can take random path and are unpredictable. It leads to every attempt of predicting intra-day stock prices futile over a larger period as discussed in random walk theory [1]. However, improvements in artificial intelligence methods and availability of powerful processors along with the growth of available data have significantly improved the accuracy of prediction.

Stock price is one such parameter, which can be either closing price or opening price. The reason being, its the most crucial stock market indicator on which investors rely. This paper considers closing price as stock prices and is used alternatively on various sources.

The majority of the earlier work in this domain involved prediction based on the time-series analysis over various technical indexes such as closing price, opening price, and volume of the Standard & Poor's (S & P) 500 indexes [2, 3]. However, such statistical analysis or artificial intelligence models solely depend upon dataset involving only the previously collected stock prices known as financial records [4, 5].

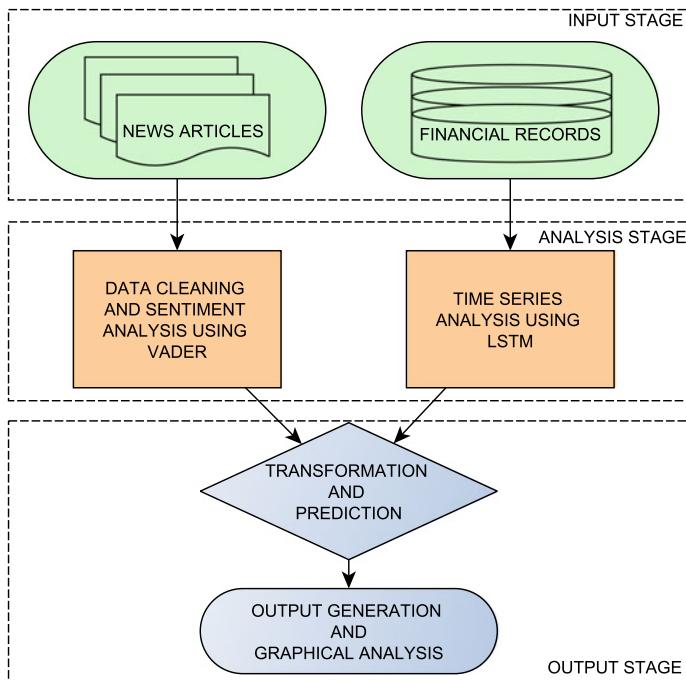


Fig. 1 LSTM-VDR model design

Growth of online newspapers, social media, and blogs has provided financial articles related to stocks of companies that were either scattered or unavailable earlier. The Internet has brought scattered data and information accessible easily and faster than ever before. It has geared new stock prediction techniques over those financial articles such as news impact on stock prices using sentiment analysis or event-based stock movement prediction [6, 7]. The studies show that recent news can have a substantial effect on the market trend analysis and should be considered to capture subtle changes [8, 10].

Prior research includes the use of techniques like, noun phrases, bags-of-words, or custom corpus that captures and performs better for certain stocks only [14, 15]. Advancement in NLP has helped in the determination of stock movement with better accuracy for sentiment analysis as discussed in [11–13].

This work considers the best of both the analysis using the hybrid LSTM-VDR model as shown in Fig. 1. The model considers LSTM architecture for the time-series analysis of intra-day stock prices [9]. VADER, on the other hand, applies sentiment analysis over the news articles for generating sentiment scores. It considers a combination of lexicons, which are collections of features such as phrases. It is usually labeled according to its orientation of semantics as either negative or positive [16]. The predicted time-series values, together with sentiment scores of recent news articles collected from various online sources, gives the final stock price predictions for certain number of days.

The discussion of paper follows Sect. 2 which describes the hybrid LSTM-VDR model. Section 3 explains the working through an example. Following it, Sect. 4 evaluates the result and perform comparison between different models.

2 Model Design and Problem Evaluation

2.1 Input Stage

The architecture considers two categories of input, the first one is financial records, and the second one is the sequence of recent news articles. In order to differentiate them, the input layer is restated as a quantitative layer and qualitative layer respectively in following paragraphs.

The quantitative layer takes chronologically ordered seven technical indicators as input from Yahoo Finance for S & P 500 index [17, 18]. For this work closing prices is selected as the technical indicator and is arranged chronologically according to its corresponding date. The qualitative layer takes news articles from Web scrapping of some online newspapers such as economics times et al. [19, 20]. Web scrapped data from various sources is combined and prepared in a paragraph as the master article. Then, sentiment scores are evaluated and arranged in chronological order according to date.

2.2 Analysis Stage

Time-series analysis of financial records performed using LSTM has internal mechanisms known as gates that regulate the flow of information [21]. The architecture as in Fig. 2 and Eqs. (1)–(6) explains the work in detail. Algorithm 1 explains the cell state calculations that selects the next cell state which in turn predicts the final value.

Cell state is the most important part of LSTM architecture, denoted as c_t , where t represents the timestamp. Three different gates, namely, the forget-gate, the input-gate, and the output-gate, evaluates the cell state c_t and output h_t . The forget-gate, f_t , decides which value from previous data to forget or remember. The input-gate, i_t selects the input signal that updates the values of current cell state. The output-gate, o_t allows the cell state to determine whether it has effect on other neurons or not. It generates the output considering the dependencies through activation function at gates. It also prevents the vanishing gradient problem of RNN.

$$f_t = \sigma(W_{vf} * v_t + W_{hf} * h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(b_i + W - vi * v_t + W_{hi} * h_{t-1}) \quad (2)$$

$$\tilde{c}_t = \tanh(b_c + W_{vc} * v_t + w_{hc} * h_{t-1}) \quad (3)$$

$$c_t = i_t * \tilde{c}_t + f_t * c_{t-1} \quad (4)$$

$$o_t = \sigma(b_o + W_{vo} * v_t + W_{ho} * h_{t-1}) \quad (5)$$

$$h_t = \tanh(c_t) * o_t \quad (6)$$

Equations (1)–(6) represent the LSTM equations, where v_t is the recurrent layer input, h_t is recurrent unit output, and W is the weight matrices as in [22].

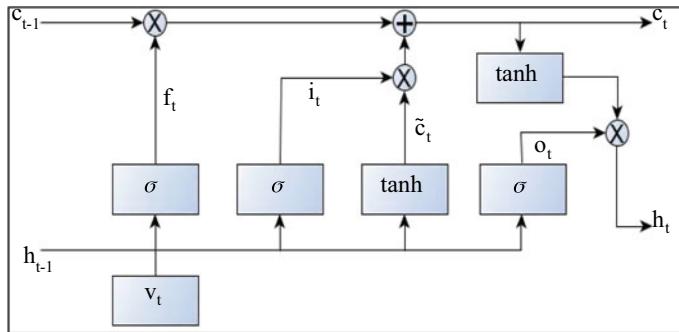


Fig. 2 LSTM architecture [22]

Algorithm 1: Calculation of cell state values.

Output: Determination of cell state c_t and value h_t , the next state should have from previous values.

Input: Previous c_t and h_t values along with current input V

$c_t = [0, 0, \dots, 0];$
 $h_t = [0, 0, \dots, 0];$

while v_t in V **do**

```

combine =  $h_t + v_t;$ 
 $f_t = \text{forget\_layer}(combine);$ 
 $\tilde{c}_t = \text{candidate\_layer}(combine);$ 
 $i_t = \text{input\_layer}(combine);$ 
 $c_t = c_t * f_t + \tilde{c}_t * i_t;$ 
 $o_t = \text{output\_layer}(combine);$ 
 $h_t = o_t + \tanh(c_t);$ 
if  $h_t > t$  then
    | select that hidden state as the next state;
else
    | Save  $h_t$  and  $c_t$  values for next iteration;
    | continue to loop;
end
end

```

¹**Note:** i_t is input-gate, f_t is forget-gate, and o_t is output-gate respectively and $*$ determines element wise multiplication.

The values v_t and h_t concatenated as *combine* and fed into the *forget_layer* which in turn removes any unnecessary data. A *candidate_layer* is created using *combine*. The *candidate_layer* holds possible values for combining with the cell state. Moreover, *combine* is further supplied to the *input_layer*. This layer selects the data from the *candidate_layer* that should be added to the next cell state. After computing the *forget_layer*, *candidate_layer*, and the *input_layer*, calculations using these newly generated values and the previous cell evaluate the next cell value.

On the other hand, VADER used for text sentiment analysis of news articles provides three different categories of polarity scores, namely positive, negative, and compound. These scores quantifies the emotion intensity of a statement. It combines quantitative analysis and validation of empirical results using human raters and wisdom of crowd technique.

Algorithm 2: Determination of new stock values through the processing of news articles and financial data

Output: Comparison graphs for actual and predicted stock prices.

Input: News articles and Financial Records.

while *true* **do**

- Web scrapping of news articles;
- Applying Natural Language Processing;
- Collecting Financial records and generating sentiment scores;
- Linear transformation of sentiment scores and closing prices;
- Generation of values considering date as index;
- Predicting news values using LSTM;
- Generating graphs and accuracy reports;

end

²**Note:** Results are generated for companies listed in BSE.

2.3 Output Stage

The predicted output value from the time-series analysis for day n validated by combining with the sentiment score of the recent news articles of $((n - 1)\text{th})$ days produce the final stock price. The predictions are combined with sentiment scores of news articles since market trends are susceptible to recent changes as explained by Vargas et al. [22]. The proposed hybrid model evaluates over time-series data using deep learning as well as handle recent market changes using sentiment analysis. Moreover, it is robust in predicting some stock prices for S & P 500 index for the Indian market over similar hybrid models as shown Table 2. It also shows advancements with VADER, not considered in previous works. Algorithm 2 demonstrates the workflow.

3 Experimental Setup

This section illustrates a working example of the proposed model and comparison of various models.

This illustration takes sample data for stock prices of six days as the input X shown in Table 1 and predicts the stock price for the seventh day. Average of input values is $\bar{X} = 1582.85$ and normalization of data between -1 and 1 calculated as $\frac{X - \bar{X}}{\bar{X}}$. The average of normalized value turns out to be $V = 0.003$.

LSTM works in similar fashion where the average of first n days calculate price for the $((n + 1)\text{th})$ day and feed again along with next n values to calculate the $((n + 2)\text{th})$ day closing price and so on. This example considers c_t and h_t as 1 such that the cell in focus has a higher probability of selection to calculate the next value.

Table 1 Sample stock closing prices of for six days with associated dates

| Date | Closing price | Date | Closing price |
|------------|---------------|------------|---------------|
| 12-12-2019 | 1599.10 | 12-18-2019 | 1566.60 |
| 12-13-2019 | 1609.95 | 12-19-2019 | 1582.90 |
| 12-16-2019 | 1575.85 | 12-20-2019 | 1568.20 |
| 12-17-2019 | 1562.70 | | |

De-normalized value for the seventh day turns to be 1577.45 which results to a difference of 5.39 from the actual price and an accuracy of $100 - 59.04 = 40.96\%$. The sentiment analysis through VADER generates the polarity scores between $[-1, 1]$, where values closer to 1 signifies greater possibilities of increased stock prices for the following day and vice-versa. The difference of predicted and actual value is 5.39 and considering sentiment score of -0.67 decreases the value as $1577.45 - 5.39 * 0.67 = 1573.84$ that increases the accuracy to 64.05%.

The accuracy of different models is compared based on stock prices. The comparison is with respect to combination of methods that include architectures such as linear regression (LRGS), moving average (MAVG), k-nearest neighbour (KNBR), and auto ARIMA (ARM) to evaluate the technical values and NLP methods like Naive Bayes and SVM for sentiment analysis.

4 Experimental Results

This section shows a comparative study between different models for stock prices over the same period through accuracy scores as shown in Table 2. It is for the stock prices of PC Jewellers Ltd (PCJ).

Stand-alone models, as well as hybrid models, are constructed using a combination of the above models, namely, LRGS-VDR, MAVG-VDR, KNBR-VDR, ARM-VDR, LSTM-NBY, LSTM-SVM, and LSTM-VDR to predict as described in the paragraph above.

Table 2 Comparison of architectures for PCJ stocks [18]

| Model | Accuracy | Model | Accuracy | Model | Accuracy |
|-------|----------|----------|----------|----------|----------|
| LRGS | 51.231 | LSTM | 70.896 | ARM-VDR | 63.376 |
| MAVG | 53.746 | LRGS-VDR | 52.443 | LSTM-NBY | 47.865 |
| KNBR | 53.984 | MAVG-VDR | 55.418 | LSTM-SVM | 66.795 |
| ARM | 61.412 | KNBR-VDR | 59.785 | LSTM-VDR | 77.496 |

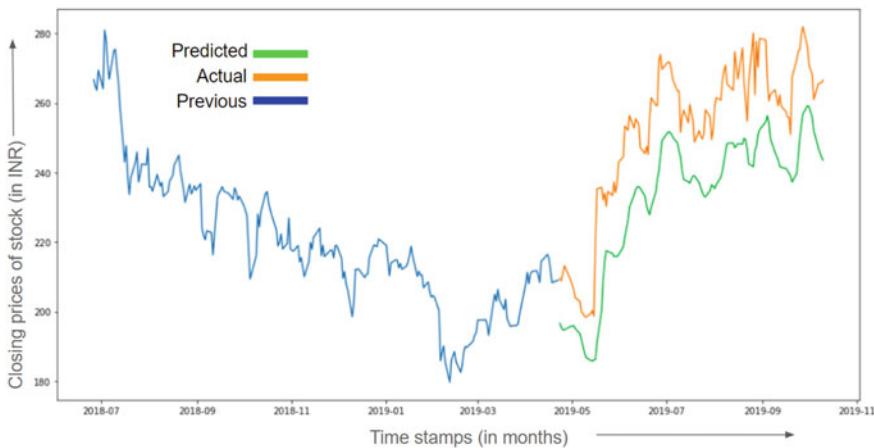


Fig. 3 PCJ

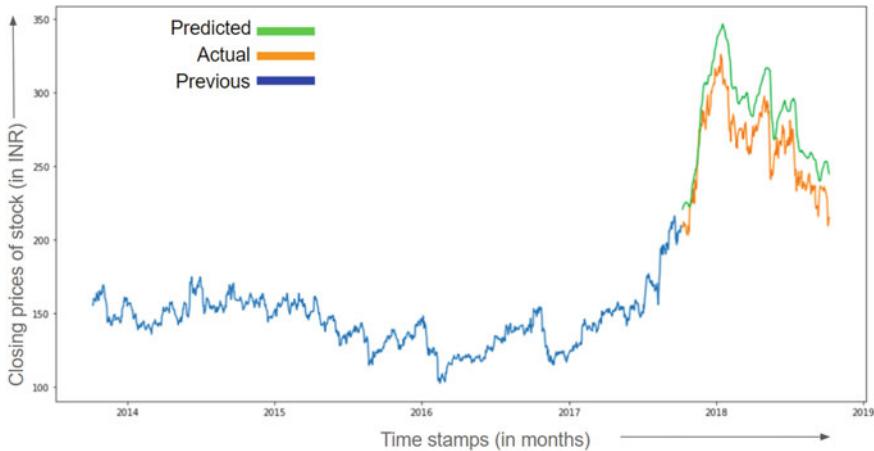


Fig. 4 RIL

Graphical comparison of the predicted and actual values using the proposed model LSTM-VDR is shown in Figs. 3 and 4 for the stocks of PCJ and Reliance Industries Ltd (RIL), respectively. It reinstates the relevance of customized input and the novel approach with better results.

5 Conclusion

The work demonstrates that the hybrid model that combines financial records and news articles can have better performance for certain market conditions. It also captures the temporal features satisfactorily even though the model considers news only from recent days. This results reinforce the fact that the information of news articles has a short temporal effect for analysis of the stock market.

References

1. B.G. Malkiel, *A Random Walk Down Wall Street: Including a Life-Cycle Guide to Personal Investing* (WW Norton & Company, New York, 1999)
2. H. Mizuno, M. Kosaka, H. Yajima, N. Komoda, Application of neural network to technical analysis of stock market prediction. *Stud. Inf. Control* **7**(3), 111–120 (1998)
3. W. Leigh, R. Purvis, J.M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decis. Support Syst.* **32**(4), 361–377 (2002)
4. K.J. Kim, Financial time series forecasting using support vector machines. *Neurocomputing* **55**(1–2), 307–319 (2003)
5. K.J. Kim, W.B. Lee, Stock market prediction using artificial neural networks with optimal feature transformation. *Neural Comput. Appl.* **13**(3), 255–260 (2004)
6. H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil, K. Muhammad, A local and global event sentiment based efficient stock exchange forecasting using deep learning. *Int. J. Inf. Manage.* **50**, 432–451 (2020)
7. S.L.O. Lim, H.M. Lim, E.K. Tan, T.P. Tan, Examining machine learning techniques in business news headline sentiment analysis, *Computational Science and Technology* (Springer, Singapore, 2020), pp. 363–372
8. R.P. Schumaker, H. Chen, A quantitative stock prediction system based on financial news. *Inf. Process. Manage.* **45**(5), 571–583 (2009)
9. S. Selvin, R. Vinayakumar, E.A. Gopalakrishnan, V.K. Menon, K.P. Soman, in Stock price prediction using LSTM, RNN and CNN-sliding window model, in *2017 International Conference on Advances in Computing, Communications and Informatics (icacci)* (IEEE, 2017), pp. 1643–1647
10. J.R. Nofsinger, The impact of public information on investors. *J. Banking Finan.* **25**(7), 1339–1366 (2001)
11. W. Wang, W. Li, N. Zhang, K. Liu, Portfolio formation with preselection using deep learning from long-term financial data. *Expert Syst. Appl.* **143**, 113042 (2020)
12. S. Hamori, *Empirical Finance* (2020)
13. R.P. Schumaker, H. Chen, Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Trans. Inf. Syst. (TOIS)* **27**(2), 1–19 (2009)
14. P.D. Yoo, M.H. Kim, T. Jan, Machine learning techniques and use of event information for stock market prediction: A survey and evaluation, in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent*

- Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 2 (IEEE, 2005), pp. 835–841
- 15. B. Wüthrich, D. Permunetilleke, S. Leung, W. Lam, V. Cho, J. Zhang, Daily prediction of major stock indices from textual www data. *Hkic Trans.* **5**(3), 151–156 (1998)
 - 16. C.J. Hutto, E. Gilbert, Vader: a parsimonious rule-based model for sentiment analysis of social media text, in *Eighth International AAAI Conference on Weblogs and Social Media* (2014, May)
 - 17. Yahoo Finance data for Reliance Industries Limited. <https://in.finance.yahoo.com/quote/RELIANCE.NS>. Last accessed on 23 Dec 2019
 - 18. Yahoo Finance data for PC Jeweller Limited. [www.in.finance.yahoo.com/quote/PCJEWELLER.NS](https://in.finance.yahoo.com/quote/PCJEWELLER.NS). Last accessed on 23 Dec 2019
 - 19. Economic Times articles for PC Jeweller Limited. www.economictimes.indiatimes.com/pc-jeweller-ltd/stocks/companyid-42269.cms. Last accessed on 23 Dec 2019
 - 20. Economic Times articles for Reliance Industries Limited. www.economictimes.indiatimes.com/reliance-industries-ltd/stocks/companyid-13215.cms. Last accessed on 23 Dec 2019
 - 21. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
 - 22. M.R. Vargas, B.S. De Lima, A.G. Evsukoff, Deep learning for stock market prediction from financial news articles, in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (IEEE, 2017, June), pp. 60–65

Detection of Alphanumeric Characters by Connectionist Temporal Classification with Vanilla Beam Search Algorithm and NLP Using MATLAB and Keras



Aseem Patil

Abstract An offline alphanumeric character recognition system using natural language processing and the Vanilla Beam Search Algorithm is described in the paper. A new method, called, heuristic search algorithm, usually used in database management applications and natural language preprocessing is introduced for extracting the features of the alphanumeric text imprinted on a rough and noisy surface. Fifty epochs, each containing 115 batches written by various people, are used for training the neural network and 650 different handwritten alphabetical and numeric characters are used for testing. The proposed recognition system performs quite well yielding higher levels of recognition accuracy compared to the systems employing the conventional horizontal and vertical methods of feature extraction. This system will be suitable for scanning alphanumeric characters into structural text for various deep learning based applications and recognizing handwritten or printed alphanumeric text using the vanilla beam search algorithm that has been proposed in this research paper.

Keywords Heuristic algorithm · Vanilla beam search · Character recognition · Alphanumeric characters · Word embedding · Feature extraction

1 Introduction

Alphanumeric recognition includes the identification of image text which is encoded inside that image and conversion of those pictures into embedded texts that can be readily understood by the machine. To recognize the symbols therein, a picture comprising of a message is analyzed and reviewed. The image will be transformed into machine-encoded script after authentication. Models created for these problems often work by analyzing prediction distributions throughout the input word range and it's up to computers to decode these frequency distributions to produce the most

A. Patil (✉)

In2things Automation Pvt. Ltd., Pune, India

e-mail: patilaseem98@gmail.com

probable colloquial expressions. This is where model construction comes into the picture. The image is first processed and the features in writing and images are transformed to a bit map, basically a black and white reference matrix. The image is then pre-processed in order to adjust brightness and hue to improve the precision of the system. The picture has now been divided into areas where the images or text are recognizing regions of concern, which assists in the extraction phase. Text regions can now be further broken down into sections, phrases, and symbols and the scheme can now match the images with comparisons and various classification algorithms. The ultimate outcome is the text of the image provided to us. Here, we shall use a different approach to tackle this problem. This paper proposes a new method of recognition of alphanumeric characters either handwritten or printed, which has an accuracy of more than 92% and uses an NLP decoder as a guide to help the Keras and MATLAB made system determine the words authenticity.

2 Implementation for the System

To understand the system more effectively we shall have a look at the block diagram we have made for the design of the system (Fig. 1).

2.1 *Input Image Capturing from the Live Stream of Raw Camera*

Image capturing also known as Image Acquisition is used to capture the image and save it in the data file. A scanned image as an input image is taken by the detection system. The picture must have a particular format, like JPG, PNG, etc. [1]. A scan, taken by a raw camera of 5-megapixel intensity value is used for capturing the image and saving the image to a required extension. A digital camera or any other suitable digital input system can be used to obtain this image.

2.2 *Increasing the Enhancement, Brightness and Contrast of the Captured Image*

Using the MATLAB functions, we can enhance the image, increase or decrease the brightness of the image or we can change the contrast of the image so as to make the image clearer than before. To use such functions we shall use the setup command used to transport the MATLAB functions to the Python application system (python3

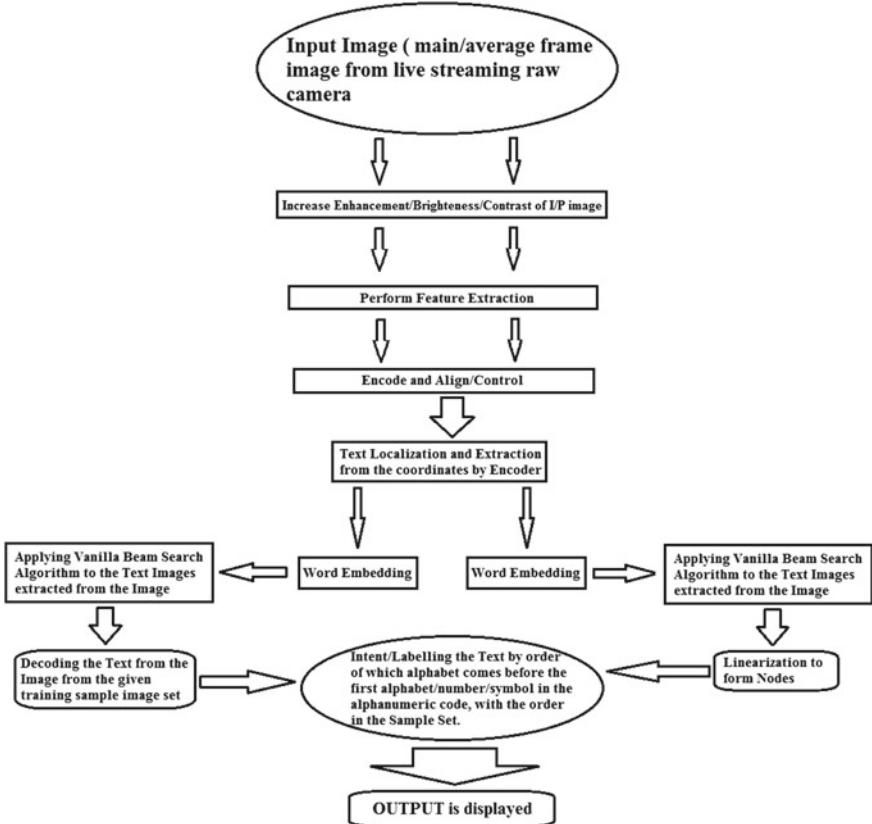


Fig. 1 Block diagram of the proposed system

`setup.py install`) from the root folder located in the bin of the python application system. For the image to be seen clearly, we shall use the enhancement, brightness and contrast image processing features [2].

We shall use feature extraction as the first step and then come to the next step which is image preprocessing of the image. After the image is preprocessed we shall again implement feature extraction so as to identify the borderlines of each alphabet and symbol typed on the piece of cardboard.

A. Feature Extraction:

Before measuring the images pixel value and finding their feature extraction points, we shall perform histogram equalization using the OpenCV function (`cv2.equalizeHist()`). Histogram equalization refers to the process in image processing used for contrast adjustment and representing the close contrast values in the form of a matrix.



Fig. 2 **a** Image captured after localization of strong feature points in the image from a live stream of the raw camera. **b** Image captured from a Basler's camera after localization of strong feature points

With this modification, the intensities of the histogram can be more efficiently distributed. It allows a higher contrast in those areas with a weaker localized contrast [3]. This is achieved by effectively extending the most frequent intensity values. The technique is effective in pictures with both bright and dark outlines and oversaturated highlights. In Fig. 2, the image is shown is the cropped part of the image after localization of feature points from the input image.

After the image is localized and the required text that needs to be identified is highlighted, we shall use enhancement and contrast so as to reduce the pollution, increase the brightness and maintain the contrast level in the image.

B. *Enhancement, Brightness and Contrast of Image:*

The enhancement of an image refers to the way in which certain images' details are clarified, enhanced or extracted according to specific requirements [4]. For example, noise removal, blurred information disclosure and level adjustment to illuminate image details. Here, we shall be using the enhancement technique used in the spatial domain. The equation for enhancing an image is mentioned below in Eq. 1.

$$g(x, y) = T(f(x, y)) \quad (1)$$

In the following equation, $g(x, y)$ represents the output image and $f(x, y)$ represents the input image. The idea is to map each pixel with a preset transition function onto a new image. After enhancing the image we shall perform contrast enhancement to the resultant image shown in Fig. 2. Contrast is the difference in light or color that distinguishes an object (or its view in an illustration or display) [5]. The difference in color and fade of the object and other objects within the same field of view determines the visual perception of the real world. Compared to normalized images, a simple technique is used to increase the contrast in a picture by “stretching the image”,

this process is referred to as contrast stretching. The intensity value range is found in the image to cover a specified value range, e.g., nearly the entire range of pixel intensities the type of image in consideration permits. We shall use contrast stretching to normalize the image without contorting the gray level intensities too much. To calculate the contrast in an image and find its external level to which it can withstand such moderations without deforming the gray level intensities can be given by Eq. 2.

$$P_{\text{out}} = (P_{\text{in}} - c) \times \left(\frac{b - a}{d - c} \right) + a \quad (2)$$

In the following equation, the highest pixel value intensity and the lowest pixel value intensity is denoted by c and d , respectively [6]. For each pixel value P , there is always a higher and lower limit for such grayscale intensity images and they are denoted by a and b , respectively. But if the output pixel intensity comes out to be 0 or 255, the value of P is reset to its original value. Hence, we used a stronger approach, which was to first select c and d on the 5th and 95th percentiles of the histogram that was made from histogram equalization, i.e. 5% of the pixel in the histogram is good enough to justify less than c , and 5% of the pixels are higher than d . This eliminates the effect of anomalies on the scaling. We can ignore all the pixel values before 1% and beyond 99% (Fig. 3).

C. Word Embedding Using NLP:

A normal practice in Natural language processing is the use in all kinds of indirect tasks of pre-trained vector representations of terms, also known as embedding. Analytically, these words are implicitly related to words useful when training data which can take advantage of contextual information. The goal is to modify pre-trained

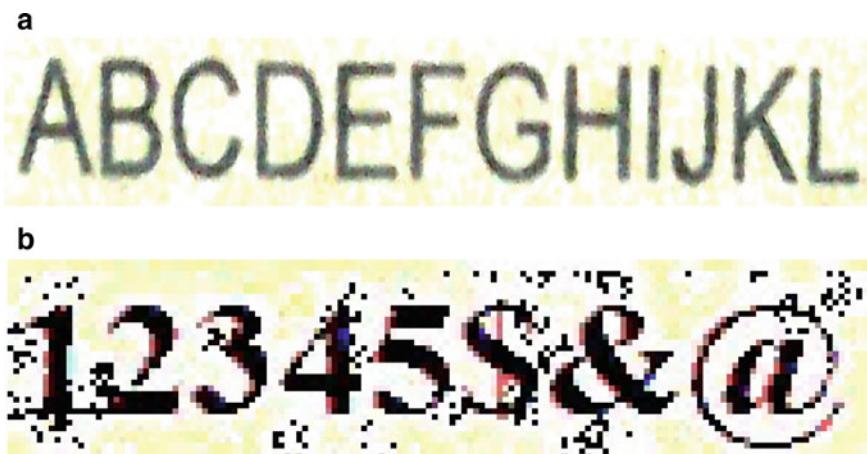


Fig. 3 **a** Output image after using image processing techniques to reduce pollution and remove noise from the image (Fig. 2a). **b** Output image after using image processing techniques to reduce pollution and remove noise from the image (Fig. 2b)

word vectors easily accessible and adjust them with your new domain content. The resulting word representations are perhaps more context-conscious than pre-trained words. This can be done by a process called retrofitting.

The Word Beam Search also known as the Vanilla Beam Search Algorithm is a heuristic search algorithm exploring a graph in a specific set seeking the most promising node [7]. Beam search is a type of best-first search algorithm that reduces its requirements for extra memory. The best first search is a graphical data model based search that, by certain Heuristic, orders all partial approaches (states). Only a certain number of the best partial solutions can be identified as candidates for the beam search. It is therefore referred to as a greedy algorithm. We need word embedding using NLP in the system for the following three reasons:

- (1) Represent words as dense real-viewed vectors that have a significant meaning. This solves most of the problems of single-hot encoding. Especially when you don't have much training data. Embedding boosts generalizes and adds performance for almost any NLP problem.
- (2) The word vector maps the semantic relation between words. It's a very important and necessary word embedding process because the NLP main issue is beneficial. If they have similar meanings, word vectors will be closed. For example, take and grasp will almost be considered as one term.
- (3) The translation of words into corresponding dense vectors has proved extremely efficient. The vector size is small and none of the vector indexes are empty. This helps the machine in understanding the words easier and does not take much time for preprocessing (Fig. 4).

In this word-to-integer mapping, we will then represent a word like the following vector of numbers:

- (1) Each word is defined as an n-dimensional vector, in which n is the vocab size.
- (2) The majority is "0" for each word, with the exception of the position that corresponds to the word's index in the vocabulary, which contains a single entry "1."

Considering the following measures taken for word embedding using natural language processing we shall have a glance at the first two words:

- (1) For our first phrase "aardvark," the vector representation will be $[1, 0, 0, 0 \dots 0]$, which in first place will be "1" followed by 5999 zeroes.
- (2) For our second word "able," $[0, 1, 0, 0, \dots, 0]$ is represented as vectors, which is "0" in the first, "1" in the second, and 5998 in the following places.
- (3) And it follows until we reach the 6000th index position.

Each word that has its own index number is converted to its own matrix form that is compared with the homogenous coordinates found on the resultant image after scanned. If the homogenous coordinates and the index numbering match up to 90% the system terminates on spot showing that the alphabet has been found at the specific position. Using the first letter of the word, the system checks for the next alphabet by comparing the coordinates with the next alphabet at hand. If they match, the system

Fig. 4 Vocabulary of words used to train the system with its respective index number

Words Used for Training with word embedding

| index | Word |
|-------|----------|
| 0 | aardvark |
| 1 | able |
| ... | ... |
| ... | ... |
| 1203 | duck |
| 1204 | dusk |
| ... | ... |
| ... | ... |
| 3309 | left |
| 3310 | lift |
| ... | ... |
| ... | ... |
| 5410 | pain |
| 5411 | pair |
| ... | ... |
| ... | ... |
| 6000 | zebra |

6000
 words
 with their
 indices



again terminates and the data set shortens iteration after iteration resulting in the word, which has been mentioned in the image. We can visualize the effect of word embedding on the alphanumeric characters in the image as shown in Figs. 5 and 6.

We want the algorithm to act differently by recognizing a word and considering a number of points that would help in identifying the next number/symbol/alphabet in that word. Hence, we add a fixed constant to each beam state. A beam can be either in word or non-word. If the beam text is simply “Cal” we can just add words until we have a full word such as “Call” or “Calendar”. Only characters, which eventually form words can only be introduced when in words-state. At most, we add one character to each and every beam in every iteration. In Fig. 4, the block diagram shows the implementation of the vanilla beam search algorithm. In the diagram, the letter “a” substitutes itself as the parent node for the three children nodes [8]. If the first letter matches the letter mentioned in the resultant image, the state process is continued until a flaw is discovered. If the first letter is not discovered, the process goes to the next parent node to check whether they are the same or not. If they are the same, the iteration terminates. But if the alphabet does not match it is taken to the second root node for further classification. In this way, each and every alphanumeric character is recognized from the image that has been trained and tested for classification.

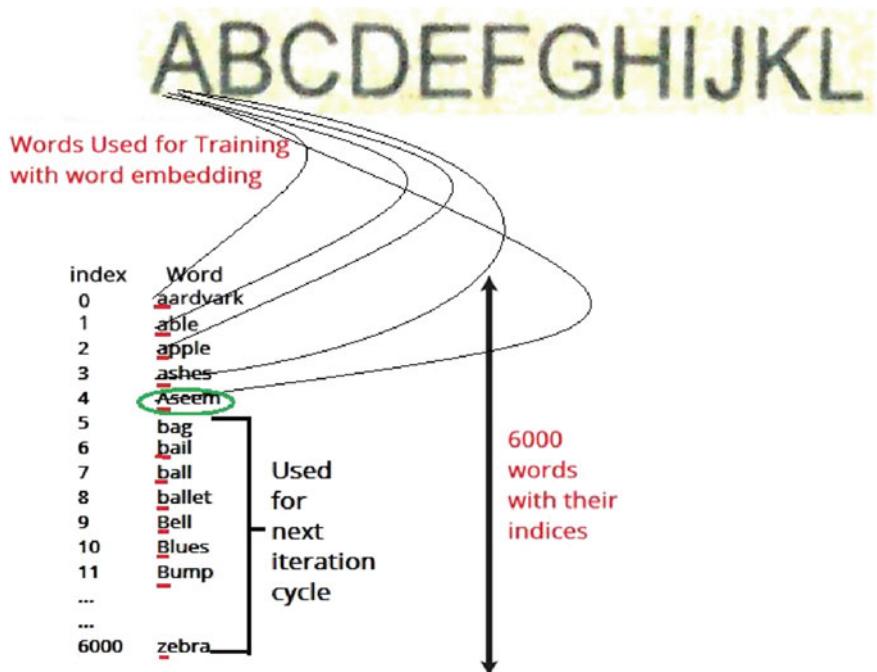


Fig. 5 Visualizing the effect of word embedding on the resultant image in NLP

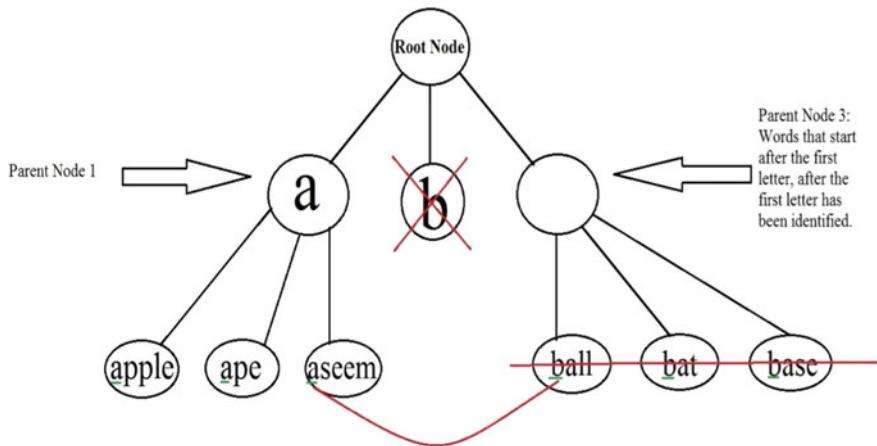


Fig. 6 Data structure nodal chain illustrating the CTC decoding after Linearization in Vanilla Beam Search

3 Results

From Table 1 we can infer that the Vanilla Beam Search Algorithm with CTC (connectionist temporal classification) has higher accuracy than any other method used (Figs. 7, 8 and 9).

Table 1 Testing of different approaches used to recognize the alphanumeric characters from the given image. Using the Optical Character Recognition (OCR), the built-in API key, Tesseract, Graph matching and the Vanilla beam search algorithm

| | Test set size (alphanumeric characters) | No. of alphanumeric characters trained | Accuracy (%) | Error (%) | Time taken (s) |
|--|---|---|--------------|-----------|----------------|
| Method 1: (OCR) | 6500 | 32,500 | 94.366 | 5.634 | 4.036 |
| Method 2: (Tesseract) | 6500 | 32,500 | 94.688 | 5.312 | 4.588 |
| Method 3: Graph Matching | 6500 | 32,500 | 91.363 | 8.637 | 8.322 |
| Method 4: Vanilla beam search algorithm | 6500 | 32,500 | 96.325 | 3.675 | 8.691 |

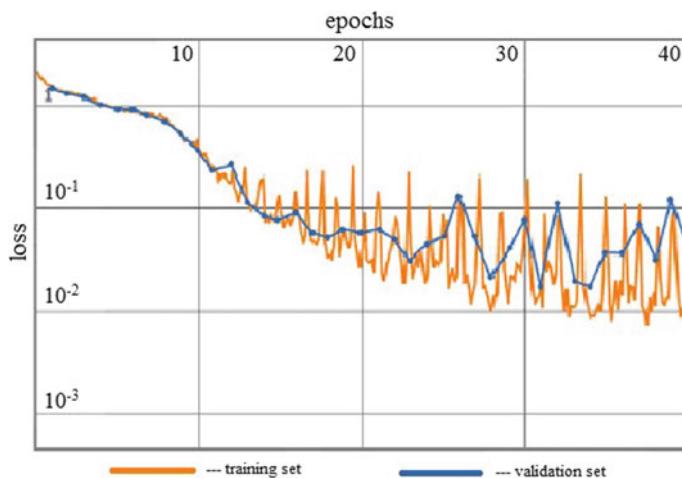


Fig. 7 Output results obtained by testing with the 50 epochs used for the Vanilla Beam Search Algorithm. The result we got was 96.325% accurate and at a much higher level than the other methods used. The orange linings indicate the training samples and the blue linings indicate the validation set of the epochs



Fig. 8 Output of the system after the recognition of every character along with their recognition percentage from the resultant image of Fig. 3a



Fig. 9 Output of the system after the recognition of every numeric and symbolic character from the resultant image of Fig. 3b

4 Conclusions

From the last handful of years, the techniques of character recognition have grown significantly. There has been a number of techniques inspired by advances in related areas such as scanning and facial recognition. We visualized the result of the proposed system in MATLAB and used Keras to build the model. We also proposed to structure these approaches according to one basic technique in this paper. This wide-ranging discussion is intended to provide insight and perhaps contribute to further development in this area. The difficulty of accurate recognition depends on the nature and quality of the text to be read. Table 1 provides an analysis of the different approaches we used for identifying alphanumeric characters. The paper focuses on the analysis of principles and procedures of the Vanilla Beam Search Algorithm and compares the result obtained to the other methods used. Despite receiving a valuable accuracy of 96.325%, the system took a bit more time than the other methods and we shall work in reducing the time complexity and compiling time needed for the preprocessing of the samples in the future.

References

1. L. Neumann, J. Matas, Text localization and recognition in real world images, in *Asian Conference on Computing Vision* (2010)
2. Y.L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level recognition features, in *Computer Vision and Pattern Recognition* (2010)
3. S. Lucas, A. Sosa, A. Tang, A. Panaretos, S. Wong, R. Young, ICDAR 2003 robust reading competitions, in *International Conference on Document Recognition and Analysis* (2003)
4. S.A. Mahmoud, B. Al-Badr, Arabic optical text recognition survey and bibliography. Process. Sig. **41**(1), 49–77 (1995)
5. Y. Pan, X. Hou, C. Liu, Localization of texts in natural scene images based on condition-random field. in *International Document Analysis and Recognition Conference* (2009)
6. M. Bhansali, P. Kumar, An alternate approach for easing the use of smart phones through test clearance. Eng. Manage. (IJAEEM) **2**(1), 211–217 (2013). (International Journal of Technology or Technology)
7. J. Yang, K. Yu, Y. Gong, T.S. Huang, Linear spatial pyramid matching by means of sparse object classification codes, in *Computer Vision and Pattern Recognition* (2009)
8. D.A. Satti, Offline Urdu Nastaliq OCR, analytical approach for printed text. Quaid-i-Azam University MS thesis report: Islamabad, Pakistan, p. 141 (2013)

Multilabel Toxic Comment Classification Using Supervised Machine Learning Algorithms



Darshin Kalpesh Shah, Meet Ashok Sanghvi, Raj Paresh Mehta,
Prasham Sanjay Shah, and Artika Singh

Abstract Web-based life has turned into an integral part of the regular day-to-day existence of a large number of individuals around the globe. Online commenting spaces generate a plethora of expressive content in the public domain, which contributes to a healthy environment for humans. However, it also has threats and dangers of cyberbullying, personal attacks, and the use of abusive language. This motivates industry researchers to model an automated process to curb this phenomenon. The aim of this paper is to perform multi-label text categorization, where each comment could belong to multiple toxic labels at the same time. We tested two models: RNN and LSTM. Their performance is significantly better than that of Logistic Regression and ExtraTrees, which are baseline models.

Keywords Long Short-Term memory (LSTM) · Logistic Regression (LR) · Extra Trees (ET) · Recurrent Neural Network (RNN) · Toxic Comment Classification (TCC) · Machine Learning (ML) · Multilabel classification

D. K. Shah (✉) · M. A. Sanghvi · R. P. Mehta · P. S. Shah · A. Singh
Mukesh Patel School of Technology Management & Engineering, NMIMS, Mumbai, India
e-mail: dkjgraphics@gmail.com

M. A. Sanghvi
e-mail: meetsanghvi98@gmail.com

R. P. Mehta
e-mail: rajpareshmehta@gmail.com

P. S. Shah
e-mail: prashamshah88@gmail.com

A. Singh
e-mail: artika.singh@nmims.edu

1 Introduction

According to a survey in 2017 [1], 41% of internet users have personally experienced online harassment, 45% of those harassed have experienced severe harassment (physical threats, sustained harassment, sexual harassment). The same survey also found that 66% of adult internet users have seen someone being harassed online. Moreover, according to a report by Statista [2], as of October 2018, the number of active users on Facebook, YouTube, WhatsApp, Facebook Messenger was more than 2.2, 1.9, 1.5, and 1.3 billion, respectively. The survey conducted by McAfee, 87% of teenagers have observed cyberbullying [3]. Another study found that 27% of American internet users self-censor online posts out of fright of online harassment [4]. The Futures Company found that 54% of the teenagers observed online bullying on social media platforms [5]. The Wikimedia Foundation found that 54% of the people who witnessed online harassment expressed lesser participation in the project that appeared in [6].

The harm that individual assaults cause to online discourse inspires numerous platforms to try to curb the phenomenon. Social media networks have been blamed for their enabling of cyberbullying because their users perform unethical practices like doxing.

The focus here is to build a multi-label text classification model that is capable of detecting different types of toxicity levels like toxic, severe toxic, threat, obscene, insult, and identity-hate. The model has been trained on a dataset of comments provided by Kaggle combined with the Facebook Shared Task Dataset. This identification could be used to help deter users from posting potentially harmful messages online.

Online comments are usually in non-standard English and contain a large number of spelling mistakes, one of the main reasons being typos (resulting from small screens of mobile devices), but more importantly because of the intentional goal to write abusive comments in innovative ways to swerve automatic filters.

The first step is preprocessing the data, after which we implemented Logistic Regression and Extra Tree Classifier as baselines. Two Deep Learning models—LSTM and RNN are implemented and give better performance than baselines. Figure 1 shows a concept map in order to depict the steps that have been taken during the research, while Fig. 2 shows the system architecture. We also built a web interface that takes comments as input and shows the toxicity levels associated with it as predicted by each model mentioned above. It provides a visual insight to the results obtained.

The rest of the paper is organized as follows. Section 2 covers related work. Section 3 talks about the datasets we used, while Sect. 4 details the models and its implementation. Section 5 covers the analysis of results. Sections 6 and 7 are the conclusion and future work, respectively.

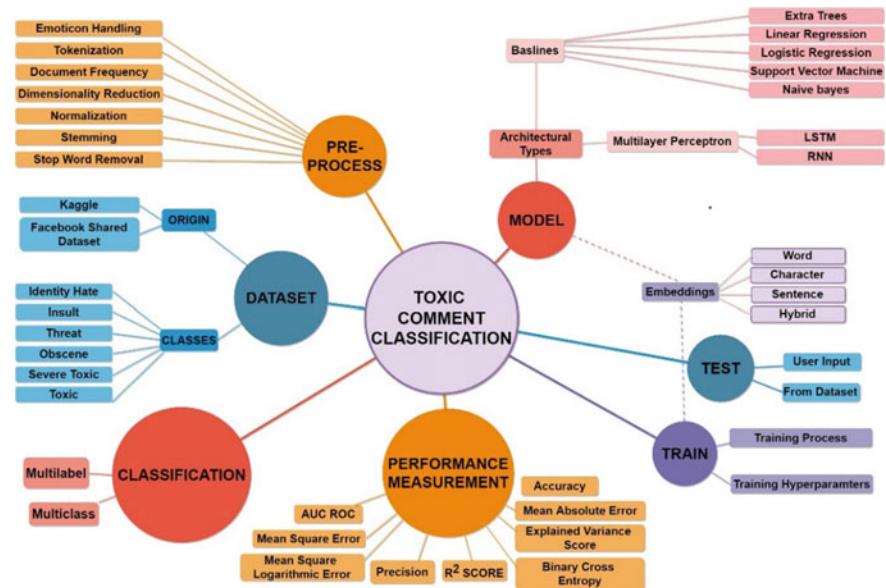
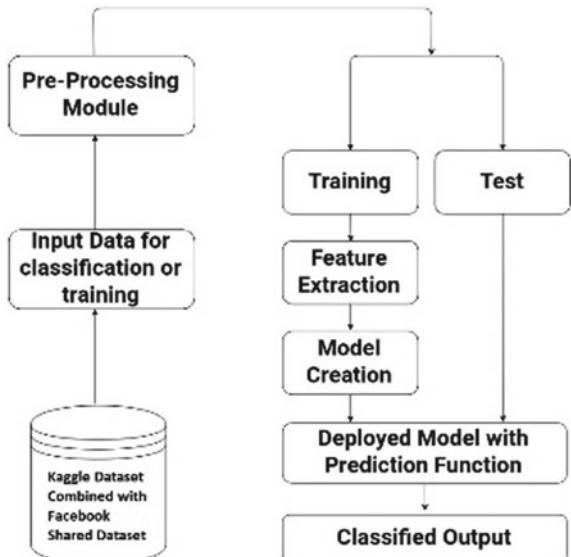


Fig. 1 Concept map for toxic comment classification

Fig. 2 Proposed system architecture



2 Related Work

The authors in [7] justify why CNN is better and motivates the research on the neural networks for future projects based on text mining. They compare CNN with the traditional bag-of-words approach with four standard machine learning algorithms for toxic comment classification, of which CNN provides a better model overall. The bag-of-words approach is used with kNN, LDA, Naive Bayes, SVM.

Online hate speech and cyberbullying are the root cause of people suppressing others, harassment, suicide, and physical violence. Paper [6] employs crowdsourcing technique to label the dataset. The technique is analyzed by many researchers. Each of the comments in the dataset is labeled by at least 10 different people, and the aggregate is computed for the final label. Model building is considered on the basis of:

1. Model Architecture (logistic regression, neural networks)
2. Embedding used (character or word level)
3. Label Type (OH, ED).

The author in [8] first discussed the application of text mining, i.e., Info Retrieval and Extraction, NLP, Categorization, and methods of Stop word removal techniques like term based random sampling, Zipf's Law and Mutual Information. The author also compared 9 stemming algorithms that are either truncating, statistical, or of mixed type.

In [9], the authors have proposed to use a two-step approach. In the first step, the comment is classified as either non-abusive or abusive. If the comment is classified to be abusive, in the second step it is classified as being either racist or sexist. Use of character-level embedding, word-level embedding, and a hybrid approach has been chosen.

3 Dataset

The Dataset employed is mainly from the competition on Kaggle under Toxic Comment Classification Challenge. Figure 3 shows the distribution of records in each label in the dataset. It can be visualized that the data is highly imbalanced—each label has irregular distributions leading to less effective classification. This can cause the model to discover incorrect relationships among the data, hence producing wrong results. To overcome this problem of highly imbalanced dataset various techniques like sub-sampling of the dataset, a mixture of two or more labels, PCA et cetera could be applied. We have combined our dataset with the Facebook Shared Task Dataset mentioned in [10], shown in Fig. 4. This is done to increase the number of samples for the labels that had less counts, i.e., Obscene, Threat, Insult, and Identity Hate.

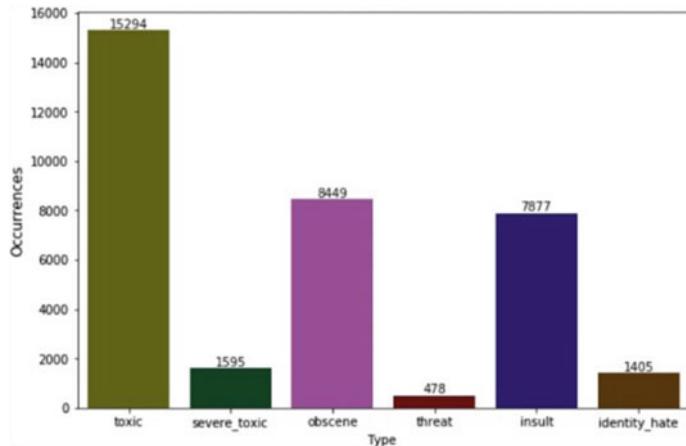


Fig. 3 Distribution in Wikipedia dataset

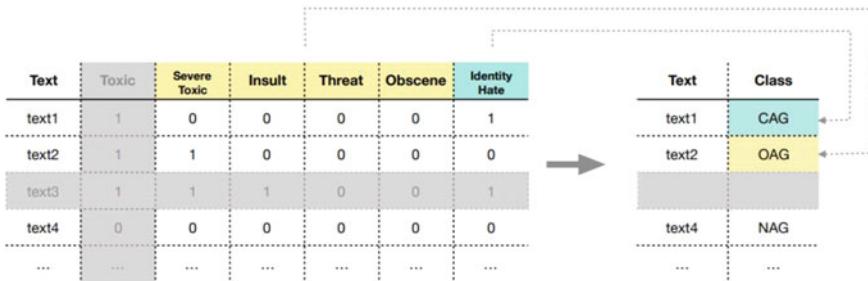


Fig. 4 Augmenting dataset to reduce the imbalance present [10]

In [11], the authors propose an ensemble utilization of recurrent and convolutional neural systems to catch both the local and global semantics. Also, to model a high-order label connection while having a tractable computational multifaceted nature. Broad trials demonstrate that the proposed methodology accomplishes state-of-the-art execution when the CNN-RNN model is prepared to utilize a substantially large dataset.

4 Implementation

4.1 Preprocessing

Data Preprocessing is a crucial step that has to be performed during machine learning. It helps in improving the overall quality of the data, hence leading to a better model. The preprocessing steps implemented in our research have been listed below:

1. To lower case
2. Replacing URLs with “URL”
3. Replacing @ with “USER MENTION”
4. Replacing #hashtag with the hashtag
5. Replace 2+ dots with space
6. Strip space, ” and ‘
7. Replacing emoji’s with either EMO POS or EMO NEG
8. Replacing multiple spaces with a single space
9. Removing stop words
10. Stemming
11. Tokenization.

The above-listed steps have been implemented using regular expressions and python libraries like nltk, os, argparse, and pandas, to name a few.

4.2 Models

Baselines are implemented to understand the problem of General machine learning and deep learning perspectives.

Baseline Models

Logistic Regression: LR is a classification technique. It is used to analyze a dataset that consists of 1 or more independent variables that determine the outcome. Its outcome is measured with a dichotomous variable (only 2 possible outcomes).

ExtraTrees: Also known as extremely randomized trees, its main objective is to further randomize trees building in the setting of numerical input features. There exists an optimal cut-point, which is responsible for most of the variance which is induced in the tree. In order to solve the multi-label problem, we need to create six models (since we have six classes) for predicting the probability that a sentence belongs to each class or not. We find out the probability of a comment belonging to each of the six classes and do not select the one with maximum probability since ours is a multi-label classification problem.

Deep Learning Models

Recurrent Neural Network: RNN is a type of ANN. It forms a directed graph between two nodes in a sequence. Using the knowledge learned from the past experience it displays a dynamic behavior for the current input in a time sequence. They can process input in a sequential manner, making them suitable for speech and handwriting recognition tasks.

We created our RNN model using keras, pickle, numpy, scipy, and pandas library provided by python. RNNs, as the definition suggests, provide good results for data that is given as input in a temporal sequence.

Long Short-Term Memory: It is a special type of RNN, which can learn long-term dependencies. They can add or remove information to or from the cell state with the help of memory gates. In simpler terms, it can selectively remember patterns for longer durations of time and this is very helpful in solving real-life problems like speech recognition, rhythm learning, human action recognition, etc.

LSTM helps in solving the vanishing and exploding gradient problem that is encountered in RNNs. We have developed our LSTM model by using keras, pickle, numpy, scipy and pandas library provided in python. Throughout the code, we have used pickle library in order to reduce computation time by storing intermediate results.

Glove 6B with 300 Dimensions was used for both the deep learning models. It is an unsupervised learning technique, that helps creating vector representation for the underlying textual data, and the values of this vector can be used as features.

Website

We have created a web interface that enables a user to access our models functionality. It is created with the help of technologies like HTML, CSS, and JS. Forms were used to send and retrieve data from the user. Flask Server was deployed using python to act as a web server for our web page so that the website can be accessed from any browser-enabled device. CanvasJS is used that plots the results given by our model into a multidimensional graph with values for each class and its probability of being a member of that class. It gives the user better understandability of the results. Finally, using NGROK, a python library, the localhost server running on our GTX 1060 machine was ported and assigned a separate link and became accessible by any another device using a dedicated IP-address-based URL. Figure 5 shows the screenshot of the website with output for the sentence “I am going to kill you”, scale being between 0 and 1 level of toxicity. When the performance metrics are compared, RNN performs better.

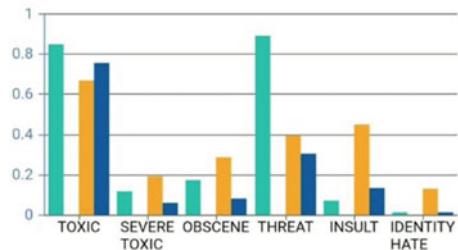
5 Analysis

RNNs have their internal memory, which allows them to “remember” previous inputs, hence making them suitable for sequential data. Hence, when they are considering input i1, they can access the result of input 0. Similarly, while working on input 3, they

Fig. 5 Website output

The screenshot shows a web-based application. At the top, there's a header with a link to '39392785.ngrok.io/result' and some icons. Below the header is a table with two rows. The first row has 'INPUT' in the left column and 'OUTPUT' in the right column. The second row contains the text 'I am going to kill you' in both columns. Underneath this table, it says 'Created By:' followed by four boxes, each containing a name and a team ID: C033 - Raj Paresh Mehta; C050 - Meet Ashok Sanghvi; C051 - Darshin Kalpesh Shah; and C054 - Prasham Sanjay Shah. At the bottom center is a green 'Home' button.

TOXIC COMMENT CLASSIFICATION



can access the results of previous inputs. There are limitations to exactly how much history they have access to. This also allows them to create a deeper comprehension of an ordered input along with its context when compared with other models.

The same can be seen in Figs. 6, 7, 8, and 9 that show the respective performance metrics.

Fig. 6 Performance metrics of logistic regression

Prediction with Valid Set to calculate different Metrics

Calculation of Metrics:

Mean Absolute Error: 0.0941257457975

Explained Variance Score: -0.0120894945886

Mean Squared Error: 0.0478566785787

Mean Squared Logarithmic Error: 0.0234638201526

R2 Score: -0.0121188790783

Precision: 0.0515458709137

AUC ROC: 0.502531327362

Starting to Predict on Test Set

Prediction with LR Complete

```
mean_absolute_error: 0.0419 - mean_squared_error: 0.0208 - mean_squared_logarithmic_error: 0.0102
- val_loss: 0.0746 - val_acc: 0.9705 - val_mean_absolute_error: 0.0409 - val_mean_squared_error:
0.0215 - val_mean_squared_logarithmic_error: 0.0106
```

Fig. 7 Performance metrics of LSTM

Epoch 10 loss 0.08937794586455744 best_loss 0.07436454759985399

Fig. 8 Performance metrics of RNN**Fig. 9** Performance metrics of extra trees

```
Starting Extra Tree Classifier
Reading Files
Selecting Comments and Storing in List
Calculation of Metrics:
Mean Absolute Error: 0.0317128691232
Explained Variance Score: 0.39173485095
Mean Squared Error: 0.0154394379075
Mean Squared Logarithmic Error: 0.00732898330795
R2 Score: 0.391487770016
Precision: 0.548291609607
AUC ROC: 0.923890271157
```

6 Conclusion

The research community and the industry has been trying to curb the phenomenon of online harassment, doxing, etc. By performing this research, we aim to accomplish the same by automating the process of toxic comment classification. This was a competition posted on Kaggle Website, where many participants worked with the dataset provided; our research stands out since we combined the Kaggle dataset with Facebook Shared Task dataset. We implemented two baseline algorithms—Logistic regression and Extra Trees and two deep learning algorithms LSTM and RNN. Out of these, RNN gave the best performance. The model was combined with a Graphical User Interface for the ease of understanding and letting a nonprofessional operate it. GUI was implemented using Flask Framework in python and ngrok. The resolution can be deployed on websites where people post their comments regarding movies, books, places, to name a few.

7 Future Work

There are many improvements that can be made to the current model, a few of which have been discussed here. Trying to solve the problem of sarcasm detection and metaphor detection and personality recognition is an important aspect, along with figures of speech detection. Also, developing an API can be useful, since it can be easily integrated directly into keyboards itself. Lastly, developing a model that works on regional languages like Hindi, Gujarati, to name a few will also prove to improve the overall spectrum in which the system can be deployed.

References

1. Online harassment pew research center (Online). Available <http://www.pewinternet.org/2017/07/11/online-harassment-2017/>
2. Social Network Ranking (Online). Available <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>
3. F. Mohammad (2018) Is preprocessing of text really worth your time for online comment classification? arXiv preprint [arXiv:1806.02908](https://arxiv.org/abs/1806.02908)
4. A. Lenhart, M. Ybarra, K. Zickuhr, M. Price-Feeney, Online harassment, digital abuse, and cyberstalking in America. Data and Society Research Institute (2016)
5. Teen Internet Safety Survey 2014 (Online). Available <https://www.cox.com/content/dam/cox/aboutus/documents/tweeninternet-safety-survey.pdf>
6. E. Wulczyn, N. Thain, L. Dixon, Ex machina: personal attacks seen at scale, in *Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee* (2017), pp. 1391–1399
7. Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G., Plagianakos, V.P. Convolutional neural networks for toxic comment classification, in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence* (ACM, 2018), p. 35
8. S. Vijayarani, M.J. Ilamathi, M. Nithya, Preprocessing techniques for text mining—an overview. *Int. J. Comput. Sci. Commun. Netw.* **5**(1), 7–16 (2015)
9. J.H. Park, P. Fung, One-step and two-step classification for abusive language detection on twitter. Hong Kong University of Science and Technology
10. P. Fortuna, J. Ferreira, L. Pires, G. Routar, S. Nunes, Merging datasets for aggressive text identification, in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)* (2018), pp. 128–139
11. C. Guibin, Y. Deheng, Z.X.C. Jieshan, C. Erik, Ensemble application of convolutional and recurrent neural networks for multilabel text categorization, in *International Joint Conference on Neural Networks (IJCNN)* (2017)

Model of Speed Spheroidization of Metals and Alloys Based on Multiprocessor Computing Complexes



Gennady Shvachych, Boris Moroz, Andrii Martynenko, Iryna Hulina, Volodymyr Busygin , and Dmytro Moroz

Abstract This paper is aimed at studying the problem of developing a heat treatment model of metal. Advantageously, such a model can be used for recrystallization and spheroidizing annealing of calibrated steel. The proposed model is based on the heat treatment method of a billet from low- and medium-carbon steels intended for the fasteners' manufacture. The achieved technical result provided a process for monitoring parameters in the heat treatment of metal. In particular, the temperature in the center of the metal product cross-section is controlled that provides the material with the necessary properties. The proposed approach allows avoiding low-productivity furnaces, which are applied to anneal steel before it is stamped. The multiprocessor computing complex with its specially oriented software allows managing the temperature of heating, holding and cooling along the entire plane of the billet section, thereby improving the quality of heat-treated steel. This model application allows reducing the duration of the technological process of spheroidizing metal annealing, reducing energy consumption, and significantly improving operating conditions, including the environmental state. Besides, the presented model allows synchronizing the technological process of metal heat treatment with the technological process of manufacturing fasteners, i.e., combine wire production into a standard technological line, which includes the preparation of wire rod, the billet heat treatment, and cold drawing of steel. There were carried out the experimental studies of the heat treatment of a long product. For this, a wire of a 20 mm diameter from steel 20G2G was subjected to heat treatment. Experimental studies have shown that metal while saving the required hardness, has the necessary elasticity properties.

G. Shvachych

National Metallurgical Academy of Ukraine, Dnipro, Ukraine

B. Moroz · A. Martynenko · I. Hulina

University of Technology, Dnipro, Ukraine

V. Busygin · D. Moroz

Oles Honchar Dnipro National University, Dnipro, Ukraine

e-mail: busygine2009@gmail.com

Keywords Heat treatment · Multiprocessor computing complex · Mathematical model · Parameter control · Technological process · Steel · Temperature mode · Technological properties of metal · Software · High-Speed spheroidization · Fine-Grained structure · Phase transformations

1 Introduction

Nowadays, production practice confronts researchers with various problems, which complete solutions, in most cases, is only possible via multiprocessor computing systems [1, 2]. So, for example, metallurgy deals with a wide variety of processes. This is heat conduction, and heat and mass transfer, including hydrodynamic processes in melts, taking into account changes in the aggregation state of a substance and deformation phenomena under force and thermal loads, etc. Most of these processes can be described by differential equations of continuum mechanics, which reflect the objective laws of conservation of mass, momentum, and energy [3–5]. The solution to these problems using well-known standard approaches is complicated, which overcoming could be done through up-to-date multiprocessor computing technologies. Meanwhile, such technologies allow increasing both the speed and performance of computations. Besides, the noted features let prerequisites be generally created for the development of new promising technological processes.

Considering the mentioned in this paper, the problem of introducing new technological processes of metal heat treatment (HT) is considered. It demands to create a model of heat metal HT used in the manufacture of high-strength fasteners by cold forming without final HT. The multiprocessor computing system with its software based on a mathematical model of a billet heating process, allows in production conditions, controlling metal heating by the time of its transition to austenitic state and the phase recrystallization temperature setting in, followed by controlling required isothermal holding regime in annealing temperature range.

The research scientific novelty is that for the first time a real-time model of metal HT was created based on multiprocessor computing system allowing to control of technological parameters in different modes of metal HT. Unlike traditional approaches, opportunity was realized to improve technological properties of rolled metal by ensuring high dispersion and uniformity of a billet structure over the entire plane of its section.

2 Research Problem Analysis

Currently, metal HT is the most promising way to improve consumer qualities of metal products radically. In this case, steels for cold heading must have in the initial state the structure of granular perlite of a certain point, i.e., transfused with the globular form of the carbide phase of a certain size point. To obtain a structure

from a partially or fully spheroidized carbide phase, metal products are amenable to spheroidizing annealing

The traditional technology of spheroidizing steel annealing involves cage furnaces (bell-type or shaft-type) [6, 7]. The process includes heating the material to such a temperature value: $Ac_1 + (10-30)$ °C; holding for 20–30 h, followed by further delayed cooling. Besides, HT of wire coils, for example, in shaft-type furnaces, is accompanied by significant temperature non-uniformity in load volume.

The traditional methods disadvantages of preparing metal for cold deformation:

1. The inability to synchronize the technological process of metal HT with the technological process of fasteners production.
2. Significant energy and gas losses during metal annealing in furnaces.
3. Poor furnace performance.
4. The difficulty of ensuring uniform heating and cooling of wire coils.
5. The furnaces for metal HT are environmentally harmful.

Progressive technologies that will fundamentally change the traditional low-productive energy-intensive processes include the methods of rapid heating of metal (wire, rods) adapted to the conditions of hardware manufacturing. In this case, an HT method alternative to the furnace heating of products is the electrothermal method: electrical contact heating or induction [7]. In-line industrial introduction of induction heating for HT is practically well-known [9]. A heat-treated long product arrives at the next stage of the technological process for the manufacture of high-strength fasteners without final heat hardening.

At the same time, the described HT process for the technology of spheroidizing and recrystallization annealing has certain disadvantages, namely:

1. Lack of control of heating, holding, and cooling during metal HT.
2. The technological process of metal processing does not provide for spheroidizing annealing in the interval of perlite structural steel transformations.
3. Steel annealing provides for a considerable duration of technological process (according to authors, from 35 to 100 min). That excludes the possibility of synchronizing the technological process of metal HT with manufacturing fasteners technological process, i.e., wires production are combined into a standard technological line.

Also, there are other approaches to the implementation of the proposed method, e.g., an installation for the calibrated steel HT [10]. The following disadvantages of the proposed installation in its application for spheroidizing annealing:

1. The installation did not realize capabilities of steel HT associated with application of annealing temperature range to acquire a spheroidizing perlite structure [11].
2. The steels annealing mode is notable by a significant duration of the process.
3. There is no possibility to control the temperature of heating, holding, and cooling of steel within the billet section plane.
4. High energy intensity of the process.

3 The Research Problem Purpose and Statement

In this paper, the development of the latest metallurgical technologies based on the use of multiprocessor computing systems is illustrated based on billets HT. In this case, the billet for cold heading should have in the initial state the structure of granular perlite of a particular grade, i.e., such that has a globular form of a carbide phase of a specific size. In order to obtain the indicated structural condition, the preform is subjected to spheroidizing annealing.

This research's primary purpose lies in development an HT model of metals and alloys based on multiprocessor computing complexes used for the manufacture of high-quality fasteners without final hardening.

At the same time, the multiprocessor computing system is designed as a separate module and, using specially oriented software, allows setting and controlling the necessary temperature regimes over the entire cross-sectional plane of a billet as a result of heating and exposure. In addition, the multiprocessor system controls the temperature regime of steel processing in the annealing temperature range. Thus, the main scientific novelty of this approach is that it provides the heating, holding, and cooling of the entire metal mass to the temperatures specified by the required phase transitions' regimes. Then, steel billets being heat treated with almost the same hardness acquire a finely dispersed structure, which provides a higher level of metal ductility.

Presented model main objective aims at a significant improvement in the technological properties of a metal. While, the technological process of metal HT acquires such advantages as high productivity, reduced energy consumption, and improved performance.

4 Statement of the Research's Primary Material

To solve the problems, a model of HT of metals and alloys based on a multiprocessor computing complex was developed [12]. At the same time, researches aimed at studying the features of metal HT using such a setup have been considered in papers [13–15]. The developed installation is intended for the steel HT of billets and can be used for recrystallization and spheroidizing annealing of long-steel wire, which is used for the manufacture of high-quality fasteners.

The problems are solved because the installation for HT of metal and alloys additionally contains isothermal holding cameras, intensive spheroidization devices, a multiprocessor computing complex in the form of a separate module. A multiprocessor computing system with its software allows, based on a mathematical model of the billet heating process, under the manufacturing conditions to control the product heating to the temperature of phase recrystallization in the austenitic domain, and then, having solved the inverse problem of thermal conductivity (IPTC), monitoring the required regime of isothermal holding in the annealing temperature range. Such

a HT regime of metal and alloys allows providing it with the necessary mechanical properties for the transition to the subsequent technological operation

Technical result achieved by implementing proposed model is that due to special HT, metal and alloys acquire structure uniformity and a high level of dispersion.

For given temperature ranges, austenite continuously loses carbon and, upon reaching the necessary concentration of the latter, experiences polymorphic transformations, turning into ferrite. Moreover, high-speed spheroidization determines a more uniform distribution of cementite globules in a ferrite matrix. During isothermal exposure, the formation of a quasi-eutectoid (perlite) is completed, which includes the carbon high concentration zones and ready cementite particles. Spheroidization of the carbide phase of the metal performed in the indicated manner under the conditions of corresponding modes of a billet HT provides the structure of granular perlite. Steel billets of almost the same hardness after such processing acquire a finely dispersed structure, which provides a high level of ductility of the metal. Rapid heating of a billet and incomplete austenization of steel let certain changes in the morphology of the carbide phase occur from lamellar to finely dispersed globular. Here, the multiprocessor system software controls necessary temperature conditions for given phase transformations is run in the center of a billet section plane. The noted approach provides the necessary metal properties for subsequent cold deformation.

At the same time, the technological process of metal HT should acquire such advantages as high productivity, reduced energy consumption, and improved performance. A multiprocessor computing complex ensures the noted properties of the metal rolling heating process. Such a system is implemented as a separate module and allows using the special software to set and control the necessary temperature conditions of the metal. If necessary, this complex can control the thermal regime of processing metal in annealing temperature range. Multiprocessor computing complex includes mathematical models in the following formulation:

$$\frac{\partial T}{\partial \tau} = \frac{\partial^2 T}{\partial z^2} + \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial T}{\partial r} + W \quad (1)$$

with the criterion $\tau = \frac{a_t}{R^2}$ of Fourier, if $\tau > 0$, W considered as extraordinary power being the sources of the heat, W/m^2 .

The boundary conditions of the equation are as the following:

$$\begin{aligned} T(O, r, z) &= f(r, \tau); \\ T(\tau, 1, z) &= \text{var}; \\ \frac{\partial T(\tau, 0, z)}{\partial r} &= O; \\ T(\tau, O, z) &\neq O. \end{aligned}$$

The latter two equations in the boundary conditions depict that throughout the entire HT process, the temperature must be finite in the axis of the cylinder.

Efficient algorithms' construction that requires little computer time is based on splitting problem (1) in a specific timing ($t_{p-1} \leq t \leq t_p$) into sequence of simpler ones.

Moreover, reduction of complex problems to simpler ones is possible when original operator of a multidimensional problem can be written as the sum of simple operators. At one time, notable progress in solving multidimensional spatial problems was a series of proposals that were not entirely equivalent to each other, the purpose of which was stereotypical—to reduce multidimensional distribution problem of a sequence of schemes that include unknown quantities in only one direction. This approach, adopted as a methodological basis, allows integrating Eq. (1) as a sequence of locally one-dimensional equations of a more straightforward form. Obviously, this raises the problem of developing practical and economic algorithms for solving locally one-dimensional problems that found their development in [16–18]. Note that, here, the task of controlling the temperature processes of the developed model can be attributed to the inverse class. The circumstances can explain that it is aimed at determining control parameters based on a predetermined result (inverse control problem). But, quite often, the basis of mathematical support for non-stationary thermal experiments is represented by methods for solving the inverse problem of thermal conductivity, including determination of boundary thermal conditions, identification of heat and mass transfer processes, restoration of external and internal temperature fields, etc. Mainly, the thermal conductivity problem can be applied in the processing and interpretation of thermal experiments' results. It was here that the most significant theoretical and applied successes were achieved in terms of effectiveness of methods and breadth of their practical use. Heat conduction inverse problems formulation of in metallurgical, thermal physics can be formulated from cause–effect relation concept.

Moreover, according to the adopted model, the boundary conditions and their parameters, initial conditions, thermal physical properties, etc., should be attributed to the causal characteristics of the heat exchange process. In this setting, the determination of cause–effect relationships is the primary goal of direct heat transfer problems. Also, on the contrary, if, according to certain experiments on the temperature field, it is necessary to restore causal characteristics, then a specific formulation of inverse problem of thermal conductivity is used.

Today, stable methods of solving (regularization methods) of various ill-posed problems have been developed. On this basis of the need of reducing them to extreme settings, the inverse problem of thermal conductivity solution is interpreted as an optimal control problem. The authors proposed a simplified method for solving them, which reduces the problem of minimizing the functional with the help of solving direct problems in minimizing the functions of many variables. Studies have shown that among them, the residual principle is central: the regularization parameter (the moment iterations are stopped) is selected so that the residual of the approximate solution is comparable in magnitude with the degree of accuracy of the initial data of the problem. Such a choice of the regularization parameter is easily realized when modeling solutions on a multiprocessor computer complex. This is especially clear in iterative methods, and given the circumstances described above, this is precisely

what gives the results that are close in accuracy to optimal. Therefore, if we have a computing system with the number of processors p , then we can simultaneously calculate p values of functions that realize the separation of the minimum functional using solutions of controlled mathematical models. The parallel algorithm of numerical minimization is based on the procedure for establishing the minimum function of many variables.

Experimental studies of HT of the metal products. To test the proposed installation functions, several experiments were performed. A wire of 20G2G steel was subjected to direct HT. As an illustration, we present two characteristic experiments.

Experiment 1 At the first stage, the ferritic-bainitic (martensitic) structure of the steel served as the base. The billet was heated to the intercritical temperature zone. For the chosen billet there were identified the following critical points: $Ac_1 = 725\text{ }^{\circ}\text{C}$; $Ac_3 = 795\text{ }^{\circ}\text{C}$. The heating was to value: $Ac_1 + (10 - 30\text{ }^{\circ}\text{C})$. At the subsequent technological cycle of billet processing, isothermal exposure was run for 45 s. Under-developed technological process, a billet was cooled at a speed of 20–30 $^{\circ}\text{C/s}$ to a temperature of 620 $^{\circ}\text{C}$. Subsequently followed by isothermal holding for 45 s. At the final cycle of HT, the billet was heated at a rate of 15–25 $^{\circ}\text{C/s}$ to subcritical temperatures.

During the experiment, structure formation in the material was analyzed. Granular perlite structure with a standard 2 points score; its hardness is 148–169 HB. So, the performed spheroidization of metal carbide phase by corresponding HT regimes of a billet gives material a granular perlite structure. Wherein, speed spheroidization predetermines a more uniform cementite globules distribution into a ferrite matrix. Similar hardness steel samples after being heat-treated got highly dispersed structure providing a high metal ductility. Fast heating and partial austenitization of steel provided certain changes in the carbide phase morphology from lamellar to high globular.

Experiment 2 The ferrite-pearlite structure was adopted as the basic one. The metal HT itself was carried out according to the technology introduced in Experiment 1. For an experimental sample, the following critical temperature points were established: $Ac_1 = 725\text{ }^{\circ}\text{C}$ and $Ac_3 = 795\text{ }^{\circ}\text{C}$.

In this study, the billets structure metallographic analysis was run. According to the official estimate, billets' grain microstructure is 5 points; its hardness: 150–169 HB.

Hence, the experiments showed that the initial structure of steel in a certain way affects the further structure formation and mechanical properties of the material.

5 Conclusions and Prospects for Further Research

The improvement of available, and developing of new technological processes for metal HT demand high costs on running diverse experiments in production conditions. A severe reduction of laboratory and experimental research can be run by developing new models of metal HT based on multiprocessor computer systems. Paper develops a model aimed at study high-speed modes of HT of metals and alloys.

Proposed model introduction for spheroidizing annealing of long products let:

- monitor the technological parameters of the metal HT modes, providing it with the necessary properties, for further cold deformation;
- by a multiprocessor computing complex to control temperature of heating, holding and cooling of a billet, hence improving quality indicators of heat-treated steel;
- reduce the process duration of the spheroidizing metal annealing;
- reduce power consumption;
- synchronize technological process of metal HT with technological process of fasteners manufacturing.

6 Prospects for Further Research

The developed approach for thermal metal processing based on the multiprocessor computing system creates the problem of matching capabilities of processors and multiprocessor system network interface. Hence, promising further studies are ways to solve this problem using the example of multiprocessor systems with different types of processors. Here, it is vital to derive analytical relations to determine optimal number of nodes of a multiprocessor system, considering computing capabilities of given processors. It is also necessary to determine performance indicators of a multiprocessor system in solving the problem of metal products HT.

References

1. V. Ivaschenko, N. Alishov, G. Shvachych, M. Tkach, Latest technologies based on use of high-efficient multiprocessing computer systems. *J. Qafqaz Univ. Math. Comput. Sci.* Baku Azerbaijan **1**(1), 44–51 (2015)
2. V. Ivaschenko, G. Shvachych, M. Tkach, Prospects of network interface Infiniband in multiprocessor computer system for solving problems of calculations' area spreading. *Syst. Technol. Dnipropetrov's'k* **2**(91), 32–43 (2013)
3. G. Shvachych, A. Shmukin, Features of designing parallel computational algorithms for PCs in heat and mass transfer problems. *EEJAT* **2**(8), 42–44 (2004)
4. G. Shvachych, A. Shmukin, On the concept of unlimited parallelism in heat conduction problems. *East Eur. J. Adv. Technol.* **3**(9), 81–84 (2004)

5. V. Ivaschenko, G. Shvachych, M. Tkach, Specifics of constructing of maximally parallel algorithmic forms of the solving of the applied tasks, in *System Technology: Regional Interuniversity Collection of Proceedings*, 3–9, 91 (2014)
6. I. Dolzhenkov, V. Bolshakov, V. Dolzhenkov, Equipment of thermal shops. Dnepropetrovsk: PGASiA, 320 p. (2004)
7. G. Shvachych, A. Sobolenko, Investigation of temperature regime of annealing furnaces of mine type of wire riots. *Theory Pract. Metall.* **1**, 59–62 (2003)
8. G. Khasin, A. Dianov, T. Popova, L. Kukartseva, A. Shamov, Electro-thermal treatment and warm drawing of steel. *Metallurgy* 152 (1984)
9. M. Bobylev, V. Grinberg, D. Zakirov, Y. Lavrinenco, Preparation of structure during electrothermal treatment of steels for high-strength fasteners. *Steel* **11**, 54–60 (1996)
10. D. Zakirov, M. Bobylev, Y. Lavrinenco, L. Lebedev, V. Syuldin, Patent of Russian Federation #2137847, cl. C 21 D 1/32, C 21 D 9/60, C 21 D 11/00. Installation for heat treatment of calibrated steel. Patent holder: Open Joint-Stock Company Avtonormal. No. 98117255/02; declared 09/16/1998; publ. 09/20/1999
11. I. Dolzhenkov, Spheroidization of carbides in steel. *Metallurgy* 143 (1984)
12. V. Ivashchenko, E. Bashkov, G. Shvachych, M. Tkach, Patent 61944 of Ukraine, IPC C21D 1/26 (2006.01), G06F 15/16 (2006.01). Installation for heat treatment of long-steel products. Owners: National Metallurgical Academy of Ukraine, Donetsk National Technical University. No.u201014225; declared 11/29/2010; publ.08/10/2011, # 15
13. V. Ivaschenko, G. Shvachych, A. Sobolenko, D. Protopopov, Information system of intelligent support of decision-making for rolling process. *East. Eur. J. Enterp. Technol.* **3**, 4–9 (2003)
14. G. Shvachych, A. Sobolenko, D. Protopopov, A. Chuev, Information system for tracking a pipe rolling with tandem conditions. *Metall. Pract.* **5-6**, 76–82 (2003)
15. G. Shvachych, V. Kolpak, M. Sobolenko, Mathematical modeling of high-speed modes of heat treatment of long products. *Theory Pract. Metall. Natl. Sci. Tech. J.* **4-5**(59–60), 61–67 (2007)
16. G. Shvachych, On the algebraic approach in the concept of distributed modeling of multidimensional systems. *Theory Pract. Metall.* **6**(61), 73–78 (2007)
17. G. Shvachych, Mathematical modeling of problems of metallurgical thermophysics based on multiprocessor parallel computing systems. *Modeling* **1**(18), 60–65 (2008)
18. G. Shvachych, A. Shmukin, Some features of the design of algorithms for multiprocessor computing systems, in *Interstate Scientific-Methodological Conference “Problems of Mathematical Modeling”*. Dneprodzerzhinsk, pp. 112–114 (2011)

Prediction of Sales Using Stacking Classifier



Rajni Jindal, Isha Jain, Isha Saxena, and Manish Kumar Chaurasia

Abstract This paper aims to explore approaches to machine learning for predictive analysis of sales. The ensemble learning technique known as stacking is considered to enhance the performance of the sales forecasting predictive model. A stacking methodology was studied to build a single model regression ensemble. The findings indicate that we can improve the performance of predictive sales forecasting models using stacking techniques. The concept is that it is useful to merge all these findings into one with various predictive models with different sets of features.

Keywords Sales · Machine learning · Regression · Stacking · Forecasting

1 Introduction

Potential revenue prediction is what is called sales forecasting. An estimate of the sales is likely to take place in the near future by each manufacturer. This reports on a business company's operations. A company has to work at random in the absence of a revenue forecast. Forecasting is one of the main operational aspects. Measuring and predicting market demand is the corner-stone for successful marketing planning.

In this paper, various regression machine learning approaches like Random Forest, ADA Boost, Support Vector Machine, Naïve Bayes, etc. were first applied for predicting sales using dataset from Kaggle and their effects were studied by

R. Jindal · I. Jain (✉) · I. Saxena · M. K. Chaurasia

Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi 110042, India
e-mail: ishajain_bt2k16@dtu.ac.in

R. Jindal
e-mail: rajinijindal@dce.ac.in

I. Saxena
e-mail: ishasaxena_bt2k16@dtu.ac.in

M. K. Chaurasia
e-mail: manishkumar_bt2k16@dtu.ac.in

comparing the accuracies obtained from each classifier. Then, the ensemble learning technique known as stacking was considered for improving the results of sales forecasting.

The simplest type of stacking can be defined as an ensemble training technique where multiple classifier predictions (referred to as level-one classifiers) are used to train a meta-regressor as new features. Their predictions are stacked and used to train the meta-regressor which makes the final prediction. The classifiers used in our stacking classifier boosted algorithms such as AdaBoost (Adaptive Boosting), Gradient Booster, and XGBoost at level one. Boosting is an ensemble method to improve any given learning algorithm's model predictions. The concept is to train sequentially weak students, each attempting to correct their predecessor.

Random Forest Regressor was the meta-regressor used to make the final prediction. The basic idea behind this is to combine multiple decision trees instead of depending on individual decision trees to determine the final production.

2 Related Work

Sales forecasting is widely recognized, and it has evidently improved the quality of business strategy [1]. Analysis and inference of market sales data are important. Future sales of customers can be predicted by considering previous sales of customers [2]. Lack of information, incomplete data, and outliers make it a complicated problem. Different combinations of forecasting methods were investigated in [3]. It is shown that one can receive significant benefits in accuracy when different models are based on different data and algorithms. Specific ensemble-based methods were considered for classification issues in [4–7].

Sales forecasting is a topic of regression. Machine-learning algorithms allow patterns to be identified in the time series. With supervised machine-learning methods, we can find complicated trends in the sales records. Some of the most common are machine-learning algorithms based on trees [8, 9], for example, Gradient Boosting Machine and Random Forest [10]. Also, the method of sales forecasting using fuzzy logic, data warehouse and Naïve Bayesian classifier has been explored in [11]. One of the key premises of regression methods is that trends will be replicated in the future in past records. In [12], we examined the operational analysis of factory failure detection.

3 Proposed Method

Our proposed algorithm includes the following steps.

3.1 Data Pre-processing

For our analysis, we used the sales data of a leading retailer in the USA from Retail Case Study Kaggle competition [13]. This data describes sales of different product categories like women clothing, men clothing, and other clothing.

The sales data table contains information like Year, Month, and Product category and Sales (In Thousand Dollars), provided for the period 2009–2014. Statistics such as CPI, GDP, cotton production, mill use, unemployment rate, etc. are given for the period from 2009 to 2016. The weather data includes attributes like a month, year, temperature, precipitation, humidity, etc.

Data Pre-processing is a method used for converting the raw data into a clean dataset. In other words, when the data is collected from different sources, it is collected in a raw format that cannot be analyzed.

Pre-processing data included filling the columns with null values with correct values, aggregating weather data columns by year and month, and then encoding the months with corresponding numbers. Missing values must be treated because for any of our performance metrics they reduce the quality. It can also lead to incorrect prediction or classification and can also result in a high bias for any given model used. The weather data is then grouped by year and month, i.e. data is split into groups based on year and month. After aggregating the columns, label encoding translates the month label into numerical form in order to transform them into a machine-readable form. It's an important preprocessing phase in machine learning for the structured dataset.

3.2 Merging Data

In sales and even pricing, the weather can also play a major role. The weather will decide how consumers buy from hot summers to rainy winters, what they buy, when retailers introduce new lines and when selling cycles begin. Therefore, both these data tables were merged together with the sales data on the month and year columns to form the main dataset after proper pre-processing of the weather and macro data. This is done to take into account the weather impact on clothes sales.

3.3 Modelling Data

The merged dataset is then split into training and testing dataset using train_test_split by sklearn library. In our model, 70% of data is used for training and 30% for testing.

The machine learning algorithms like Random Forest, Support Vector Regression, Linear Regression, AdaBoost (Adaptive Boosting), Gradient Boosting and XGBoost are trained using the training dataset. Hyperparameter tuning is also done for some machine learning algorithms.

3.4 Forecasting Using Stacking Classifier

Stacking is another ensemble model in which a new model is learned from two (or more) previous model's combined predictions. For each sequential surface, the model predictions are used as inputs and combined to form a new collection of predictions. It can be used on additional layers, or with a final result, the cycle can end here. Their predictions are stacked and used as characteristics to train the meta-classifier making the final prediction.

The stacking training technique of the ensemble blends multiple models of classification or regression through a meta-classifier or meta-regressor. Based on a complete training set, the base level models are trained, then the meta-model is trained as features on the outputs of the base-level model. The base-level also consists of various learning algorithms, so often heterogeneous are stacking ensembles.

In our stacking classifier, at level one we used boosting algorithms, i.e. AdaBoost, Gradient Boosting and XGBoost along with Random Forest and Bagging Regressor. Then predictions from the above five models were used in the classifier in the next layer which is the Random Forest Regressor which is the bagging ensemble technique.

4 Results

In this section, we test few machine learning algorithms like Random Forest, Support Vector Regression, Linear Regression, AdaBoost (Adaptive Boosting), Gradient Boosting and XGBoost for training and testing dataset. The machine learning algorithms are compared using R-squared metric. The R-squared metric was used to analyze the results of using different approaches to machine learning. R-squared metric measures the data points dispersion around the fitted axis of regression. The results obtained after training dataset with the machine learning algorithms can be visualized using the graph in Fig. 1 and the R2 score observed are displayed in Table 1.

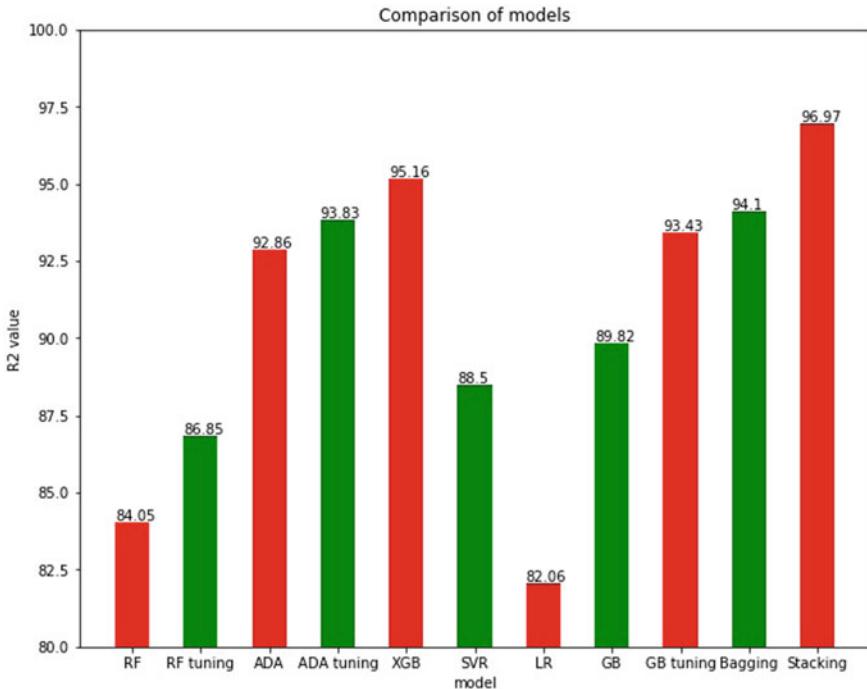


Fig. 1 Comparison of models

Table 1 R2 score of different regressors

| Regressor used | R2 score (in %) |
|---------------------------------|-----------------|
| Random Forest (RF) | 84.05 |
| Random Forest with Tuning | 86.85 |
| Support Vector Regression (SVR) | 88.50 |
| Linear Regression (LR) | 82.06 |
| AdaBoost (ADA) | 92.86 |
| AdaBoost with Tuning | 93.83 |
| Gradient Boosting (GB) | 89.82 |
| Gradient Boosting with Tuning | 93.43 |
| Bagging Regressor (BR) | 94.10 |
| XGBoost (XGB) | 95.16 |

Table 1 shows the R2 score obtained by using different machine learning algorithms.

The r2 score of different regressors is shown, and as we can see the r2 score of XGboost is maximum among other regressors. XGBoost is a gradient-boosted decision tree optimized for speed and performance. Hence XGboost performs better

and gives maximum accuracy among others. Then, using the ensemble technique known as stacking, we used the above algorithms at first level and Random Forest Regressor at the next level.

A set of Stacking classifiers is created each with a unique combination/stack of classifiers in the first layer. For each Stacking classifier, the meta-regressor will be tuned. The results of various combinations can be compared using the graph given in Figs. 2 and 3.

The various combinations of regressors used in the first layer of stacking classifiers are Bagging Regressor with Linear Regression which gave r2 value of 96.56, Random Forest with AdaBoost and Gradient Boosting which gave r2 value of 96.87 and so on.

After analyzing the results of various combinations, it was observed that the best R2 score was obtained by using the Stacking classifier where Random Forest, ADABOOST, Bagging Regressor, Gradient Boosting and XGBoost at first level followed by Random Forest Regressor at base level which was 96–97%.

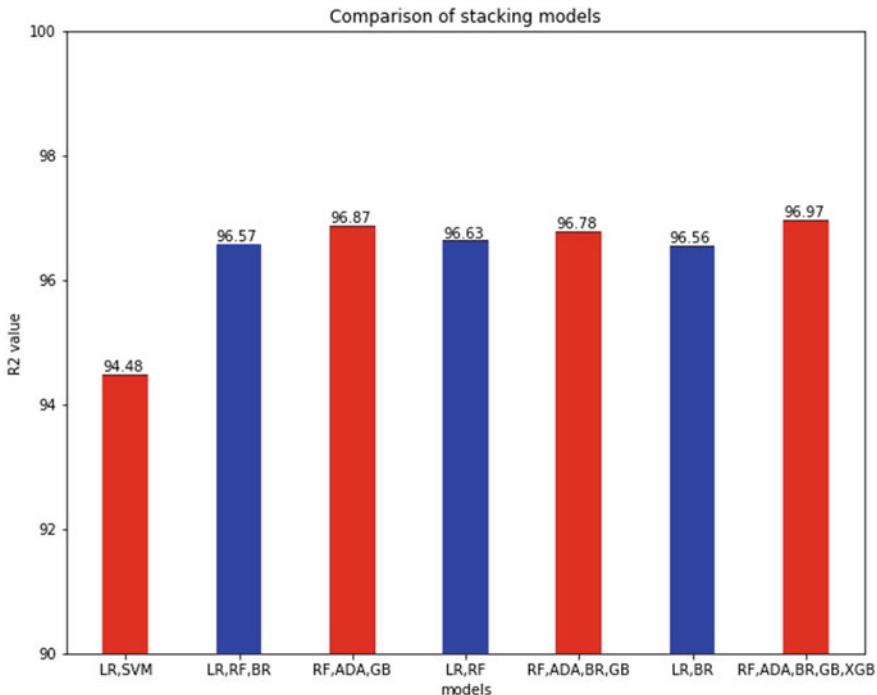


Fig. 2 Comparison of stacking models

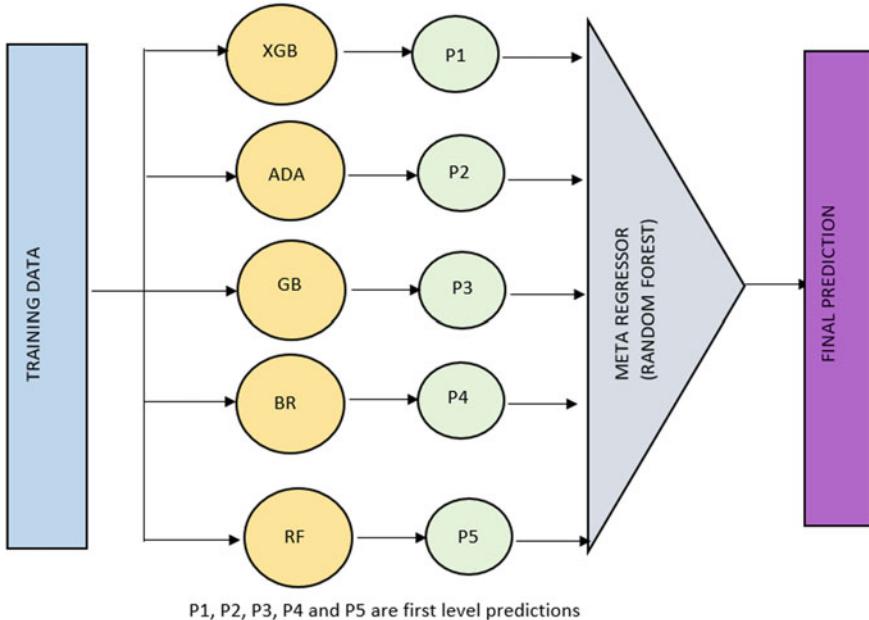


Fig. 3 Stacking classifier

5 Conclusion

In this paper, we propose an ensemble technique that is stacking for predicting sales of an organization. From the results, we conclude that stacking of boosters performs better than stacking of baggers. The use of stacking allows the results variations of multiple models with different sets of parameters to be taken into account and the accuracy of validation and out-of-sample data sets enhanced.

Boosting is an ensemble method to boost any given learning algorithm's template predictions whereas Bagging technique can be an effective approach to decreasing a model's variability, avoiding overfitting, and improving unstable model accuracy.

Boosting attempts to enhance the predictive versatility of simple models. It trains sequentially multiple weak learners. A weak learner is a constrained model in which the max depth of each decision tree can be reduced. Every weak learner in the series is based on learning from the mistakes of the previous one. Boosting then puts together all of the weak learners into one powerful and strong learner.

Bagging is a technique where a significant number of strong learners are trained in parallel. It then puts together all the strong learners to smooth out their forecasts. All the models are integrated at the end, resulting in higher stability and lower variance compared to the individual models. Bagging uses complicated base models and attempts to smooth out their forecasts while boosting uses simple base models and attempts to improve their aggregate complexity.

And as we can see boosters perform better than baggers when used to stack combinations at the first stage. Ultimately, we obtained the best result when we used first-stage boosters and random forest regressors as meta regressors.

Acknowledgements We are grateful to the Department of Computer Science & Engineering (Software Engineering) at Delhi Technological University for presenting us with this research opportunity which was essential in enhancing learning and promote research culture among ourselves.

References

1. A.D. Lacasandile, J.D. Niguidula, J.M. Caballero, Mining the past to determine the future market: sales forecasting using TSDM framework, in *Proceedings of the 2017 IEEE Region 10 Conference (TENCON)*, Malaysia (2017)
2. H. Koptagel, D.C. Civelek, B. Dal, Sales prediction using matrix and tensor factorization models, in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, Sivas, Turkey (2019), pp 1–4
3. A. Graefe, J.S. Armstrong, R.J. Jones Jr., A.G. Cuzán, Combining forecasts: an application to elections. *Int. J. Forecast.* **30**, 43–54 (2014)
4. O. Sagi, L. Rokach, Ensemble learning: a survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **8**, e1249 (2018)
5. L. Rokach, Ensemble-based classifiers. *Artif. Intell. Rev.* **33**, 1–39 (2010)
6. D.H. Wolpert, Stacked generalization. *Neural Netw.* **5**, 241–259 (1992)
7. H.M. Gomes, J.P. Barddal, F. Enembreck, A. Bifet, A survey on ensemble learning for data stream classification. *ACM Comput. Surv. (CSUR)* **50**, 23 (2017)
8. G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112 (Springer, Cham, 2013)
9. S. Mortensen, M. Christison, B. Li, A. Zhu, R. Venkatesan, Predicting and defining B2B sales success with machine learning, in *2019 Systems and In-formation Engineering Design Symposium (SIEDS)*, Charlottesville, VA, USA (2019), pp. 1–5
10. W. Huang, Q. Xiao, H. Dai, N. Yan, Sales forecast for O2O services-based on incremental random forest method, in *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, Hangzhou (2018), pp. 1–5
11. V. Katkar, S.P. Gangopadhyay, S. Rathod, A. Shetty, Sales forecasting using data warehouse and Naïve Bayesian classifier, in *International Conference on Pervasive Computing (ICPC)* (2015)
12. B.M. Pavlyshenko, Linear, machine learning and probabilistic approaches for time series analysis, in *Proceedings of the IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, Lviv, Ukraine, 23–27 Aug 2016 (IEEE, Piscataway, NJ, USA, 2016), pp. 377–381
13. Kaggle: your machine learning and data science community <https://www.kaggle.com>

How to Use LDA Model to Analyze Patent Information? Taking Ships Integrated Power System as an Example



Danyang Li and Xinlai Li

Abstract It is an important strategic issue for enterprises and countries to analyze and study the patent of ships integrated power system, understand the process of technological development, and identify potential research and development hotspots. This paper discusses the development status of the patent of ships integrated power system from the aspects of language distribution, patentee and organization distribution, patent classification number distribution, etc. On the basis of using the LDA model to explore the core technology, the technical development level in this field is understood. The research on relevant patents shows that although the technical level of ships integrated power system in China has been greatly improved, there is still a certain gap compared with the leading countries.

Keywords Patent · Ships · Integrated power system · LDA model

1 Introduction

The comprehensive electric power system of ship refers to combining the independent mechanical propulsion system and electric power system of traditional ship into one in the form of electric energy, providing electric energy for ship propulsion, communication and navigation, special operations and daily equipment through electric power network, realizing the comprehensive utilization of energy of the whole ship.

The content of patent literature is accurate, accurate and long-term, extensive and progressive [1]. Analyze and research the application of patents in related fields, understand the process of technological development, analyze the research hotspots in the field, and gain insight into the level of technological development and occupy

D. Li (✉)

School of Information Management, Wuhan University, Wuhan 430072, China

e-mail: whusimldy@163.com

X. Li

Research Center for Chinese Science Evaluation, Wuhan University, Wuhan 430072, China

the forefront of technological research. This paper increases power ships, electric propulsion and other related terms based on integrated power system (IPS), and collect relevant patent data, and explores its technology development situation. On the basis of using LDA model to excavate the core technology, it has a more comprehensive understanding of the technical development level in this field, which is of certain practical significance.

2 Data and Methods

The research object of this paper needs to obtain patent data of many countries around the world, especially in the leading countries of shipbuilding industry (the United States, Japan, South Korea, etc.). So we choose to use Derwent Innovations Index (DII) to search.

In this paper, an exhaustive search strategy is adopted to improve the comprehensiveness of the search results based on the English search keywords related to the topic, and the search characteristics of the database are adjusted. As of April 12, 2018, a total of 1773 records had been retrieved, and the result after removing the duplication was 1349.

3 Analysis of Patent Technology Development

3.1 *Language Distribution of Patent Announcement*

Derwent Innovations Index provides patent disclosure language in PD field, which can reflect the technical development level of the country or region to some extent. Of the 1349 patents granted for ships integrated power system, 220 were missing in the public language. Since a patent can be applied repeatedly in different countries or regions, thus expanding the scope of technical protection, it may occur that a patent has multiple open languages or multiple occurrences of one language. The occurrence frequency of each language is shown in Fig. 1.

As can be seen from Fig. 1, the number of public patents in English is the largest, followed by Chinese and Japanese, accounting for more than 3/4 of the total. The technology related to ships integrated power system is advanced in English speaking countries, China and Japan.

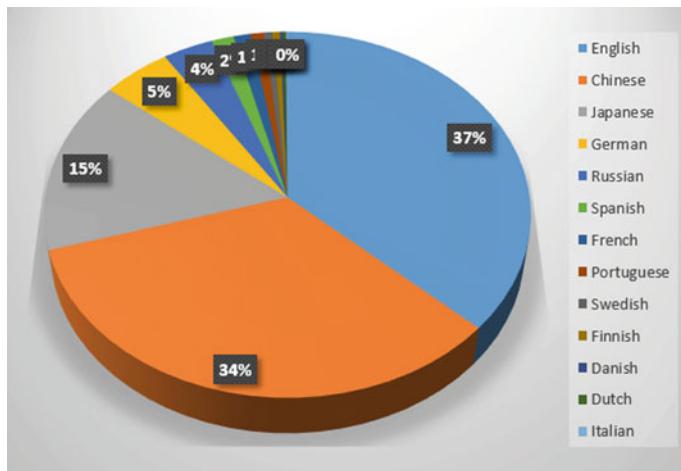


Fig. 1 Languages of patent

3.2 Distribution of Patentees and Institutions

Technical strength often refers to the patentee's patent application, possession and patent attributes, such as the applicant, inventor, technology classification, age, citations, etc. Generally speaking, the more patents the patentee applies for or owns and the larger the proportion of patents, the stronger the patentee's technical strength. The patentee can be a country, an institution, an enterprise or an individual. By comparing the patentees at the same level, the strength of each research object can be found, and competitors and partners can be identified.

In this paper, the technical power distribution of deduplication data is plotted with the help of the patent holder code of DII. The code is a four-letter code developed by DII for each patentee of each patent document included. There are standard codes and non-standard codes, which are usually determined based on the name of the patentee and can refer to the same patentee. Therefore, except for special cases, this article defaults a patent code to be a patentee. A total of 755 patentee codes were counted. In this paper, standard code, non-standard code and personal code were counted respectively, and the top 20 were counted according to the number of patents, as shown in Table 1.

According to statistics, a total of 415 patent agencies have patents related to the ships integrated power system. As can be seen from Table 1, CSHI-C has the largest number of patents, with 64. The institutions represented by FJIE-C, GLDS-C, GENE-C, WTNE-C and DEWO-C codes, in order, have more than 30 patents. Quantitatively, the institutions represented by the first 10 patentee codes may have high technology related to the ships integrated power system.

In the case of non-standard code, it is not a general term for an institution, but a combination of the first four letters of the patentee of a patent institution into a single

Table 1 Statistical table of patentee

| Standard code | Number | Non-standard code | Number | Personal code | Number |
|---------------|--------|-------------------|--------|-----------------|--------|
| CSHI-C | 64 | SHAN-Non-standard | 46 | ZHAN-Individual | 23 |
| FJIE-C | 52 | BEIJ-Non-standard | 29 | PANG-Individual | 12 |
| GLDS-C | 38 | JIAN-Non-standard | 24 | WANG-Individual | 10 |
| GENE-C | 33 | NISH-Non-standard | 20 | CHEN-Individual | 9 |
| WTNE-C | 33 | ANHU-Non-standard | 16 | JIAN-Individual | 8 |
| DEWO-C | 32 | GUAN-Non-standard | 15 | SUMI-Individual | 6 |
| USHM-C | 29 | YICH-Non-standard | 14 | CARL-Individual | 5 |
| MITO-C | 19 | CONV-Non-standard | 13 | CHAN-Individual | 5 |
| SIEI-C | 18 | SHEN-Non-standard | 13 | KOCH-Individual | 5 |
| TOKE-C | 18 | UYJI-Non-standard | 11 | HUAN-Individual | 4 |
| CRRC-C | 15 | WUXI-Non-standard | 10 | KIMJ-Individual | 4 |
| HIIH-C | 15 | CHON-Non-standard | 9 | KURO-Individual | 4 |
| RORO-C | 14 | NING-Non-standard | 9 | PERE-Individual | 4 |
| UHEG-C | 14 | ZHEN-Non-standard | 9 | YANG-Individual | 4 |
| SMSU-C | 13 | WUHA-Non-standard | 8 | YUHH-Individual | 4 |
| UWHT-C | 12 | CHEN-Non-standard | 7 | ZHOU-Individual | 4 |
| WART-C | 10 | NANT-Non-standard | 7 | BREM-Individual | 3 |
| ALLM-C | 9 | ZHON-Non-standard | 7 | DING-Individual | 3 |
| HITA-C | 9 | CNRS-Non-standard | 6 | FOGA-Individual | 3 |
| YANM-C | 9 | FOSH-Non-standard | 6 | JAMM-Individual | 3 |

code. Therefore, it is of little significance to analyze the institution contained in each non-standard code. However, it can be seen from Table 1 that the non-standard institution code with the largest number of patent rights is SHAN-Non-standard, and there are 46 patents related to the integrated power system in ships, including some research institutions and technology companies in Shanghai and a small number of institutions in Shandong province.

According to statistics, a total of 340 people hold 489 patents related to the ships integrated power system. Compared with institutions, the number of patents owned by individuals is much lower. The codes are ZHAN-Individual, PANG-Individual and WANG-Individual, which are divided into the top three. It indicates that the technology of ships integrated power system is mainly based on companies or research institutions, while the research level and technology of individuals are relatively limited.

3.3 Distribution of DC and MC Classification Number

Table 2 shows the first fifteen occurrences of the two classification Numbers. In the first 15 high-frequency classification Numbers, the main areas involved are [2].

DC classification number “W06 (Aviation, Marine and Radar Systems)”, “Q24 (Ships; Waterborne vessels; Related equipment (B63))” and MC classification Numbers “W06-C01”, “W06-C08”, “X11-U15” and “X13-U04” are directly related to the ship theme. Where, the upper class of the first two MC is “W06-C”, and the term is “Shipping”, whose subclasses are related to ships. DC classification number “X13 (Switchgear, Protection, Electric Drives)” and other X fields (Power works), as well as “U24 (Amplifiers and Low Power Supplies)” refer to parts of the general Electric propulsion, such as Power Drives, Power storage and circuit fittings. T fields (computing and Control) and W fields (communication) may be associated with decision support system parts, such as control system of “T06 (Process and Machine

Table 2 DC and MC classification number

| DC | Number | MC | Implication | Number |
|--|--------|-----------|---|--------|
| W06 (Aviation, Marine and Radar Systems) | 652 | W06-C01C7 | Electric propulsion | 307 |
| X12 (Power Distribution/Components/Converters) | 201 | W06-C01C | Electrical equipment (incl. lighting) | 234 |
| T01 (Digital Computers) | 195 | T01-J07D1 | Vehicle microprocessor system | 147 |
| X16 (Electrochemical Storage) | 194 | X13-U04 | Ships and boats | 135 |
| X13 (Switchgear, Protection, Electric Drives) | 186 | X11-U05 | Ships and boats | 116 |
| Q24 (Ships; waterborne vessels; related equipment (B63)) | 162 | X16-B01 | Cells | 65 |
| X11 (Power Generation and High Power Machines) | 153 | X16-G | Battery chargers | 63 |
| U24 (Amplifiers and Low Power Supplies) | 114 | X12-H01B | Multisource systems, system inter-connections, power transfer | 61 |
| X15 (Non-Fossil Fuel Power Generating Systems) | 79 | X21-A01F | Electric vehicle | 55 |
| S01 (Electrical Instruments) | 77 | W06-C01A | Control systems | 51 |
| X21 (Electric Vehicles) | 75 | W06-C08 | Marine vessel manufacture | 49 |
| V06 (Electromechanical Transducers and Small Machines) | 53 | U24-H | Low power systems | 48 |

(continued)

Table 2 (continued)

| DC | Number | MC | Implication | Number |
|--|--------|-----------|---|--------|
| L03 [Electro-(in)organic-chemical features of conductors, resistors, magnets, capacitors and switches, electric discharge lamps, semiconductor and other materials, batteries, accumulators and thermoelectric devices, including fuel cells, magnetic recording media, radiation emission devices, liquid crystals and basic electric elements. Growing of single crystals of semiconductors and their doping are included, but semiconductor devices, where the manufacture is not claimed are excluded. Electrography, electrophotography, magnetography, electrolysis, electrophoresis, power plant, X-ray and plasma-techniques, ion exchange resins, polyelectrolytes, electroplating, metal electrodeposition, electroforming, anodising, electrolytic cleaning, cathodic protection and electrolytic or electrothermic production or refining of metals are all covered elsewhere (Sections G, J, K and M).] | 45 | W06-C01C3 | Electrical power generation and distribution | 48 |
| T06 (Process and Machine Control) | 41 | W06-B01C | Electrical equipment (incl. de-icing, lighting) | 44 |
| Q51 (Internal combustion engines, reciprocating engines, rotary engines (F01, F02B,D,F,G,M,N,P)) | 40 | X11-G | Permanent magnet synchronous machines | 39 |

Control)", alarm system of "W05 (Alarms, Signalling, Telemetry and Telecontrol)", as well as the system and fault monitoring of "T01—N02B2B" [3].

4 Distribution of Patent Core Technology

Although the exhaustive strategy was adopted to ensure the completion rate of retrieval, the accuracy rate was not improved. In this regard, we select all DC classification Numbers to formulate co-occurrence matrix, conducts aggregation subgroup analysis, and obtains classification number clustering related to the topic.

168 DC classification Numbers were divided into 8 clusters based on the co-occurrence matrix of DC classification Numbers. DC classification Numbers with

frequency greater than 50 were selected for further analysis, mostly concentrated in three regions. We think that the classification Numbers of these three regions are related to the research topic of this paper, with a total of 48 DC classification Numbers. In order to improve the pertinently of patent technical analysis, the deduplication data were screened according to the above classification number. Finally, 453 records with strong correlation classification number were obtained.

Abstract of the patent specification is an overview of the content of the patent specification, including the name of the invention or utility model, the technical field, the technical problems to be solved, the main technical features and USES of the invention or utility model [4]. Each patent brief can be viewed as a document that provides a brief overview of the technology and can be used to train the LDA topic model to identify the technical topics of the integrated power system.

In this paper, NLTK library of python language was used to preprocess English documents such as word segmentation, and LDA topic distribution map of strongly related topics was obtained by training LDA model through gensim library and TF-IDF algorithm, as shown in Fig. 2. Each topic was represented by a circle, and the larger the circle, the more documents the topic contained. Among them, due to the absence of AB field (abstract data of patent specification) in eleven records, TI field (title) was used to supplement the missing item. According to the performance evaluation of the topic model, 2339 non-stopping words can be divided into 5 disjoint topic classes, and the perplexity of topic classification is 939.3, which indicates that the topic classification is more reasonable.

After the TF-IDF algorithm weakens the weight of terms that appear frequently in all documents at the same time, among the first 30 most prominent terms in all documents, “fuel”, “rotor”, “bus”, “auxiliary”, “stator”, “gas”, “cell”, “gear”,

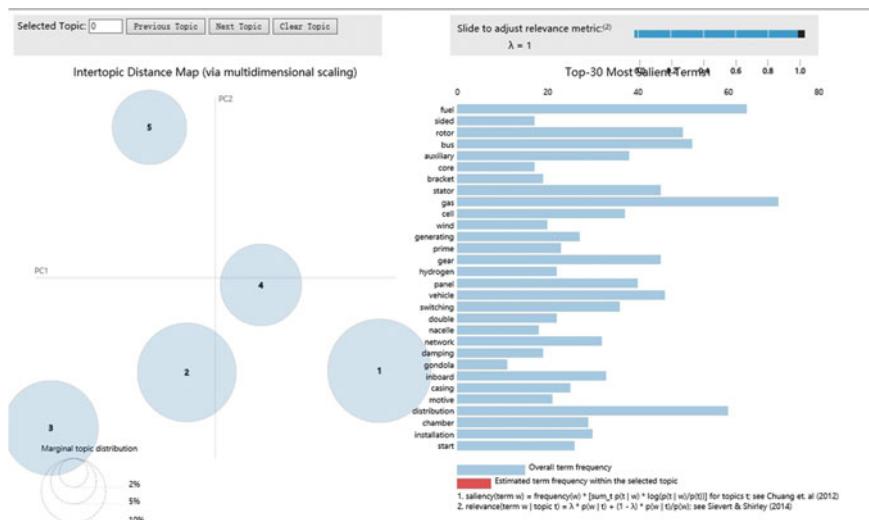


Fig. 2 The distribution of LDA themes and the first 30 salient words

“panel”, “vehicle”, “switching”, “network”, “inboard”, “distribution”, such as high-frequency terms. Among them, it has a higher correlation with the internal system and structure of the ship, and also has a relationship with road vehicles. It may be that the comprehensive power system of the ship and some of its components and principles are applied to other fields, such as automobiles, or electric vehicles or power systems of other fields and their components and principles are applied to the ship.

The LDA topic model predicts the topic of the document according to the probability distribution of words, and each word will appear in each topic with a certain probability [5]. Therefore, the LDA topic model lists the top 30 most relevant terms for each topic, in descending order of occurrence, as shown in Table 3 and Figs. 3, 4, 5, 6 and 7. Based on the terminology, the following text explains the general meaning of the five topics one by one, and predicts the distribution of core technologies of the integrated power system.

Theme 1 before 30 high-frequency related terminology, contains “converter”, “inverter”, “generator”, “unit”, “control”, “shaft”, “engine”, “transformer”, “switch”, “circuit” and “propeller”, etc., related to the internal structure of electric propulsion system and composition. For example, the combination of a power unit (direct and alternating current generator set, diesel generator set), the equipment composition of the generator set, and the devices controlling the synchronization of the power unit or power supply system with the propulsion system.

Theme 2 contains “fuel”, “engine”, “gas”, “generator”, “motor”, “diesel”, “supply”, “power”, “battery”, “output”, “switching”, “storage”, “current” and “apparatus” terms, power modules (engines, generators and motors) and distribution units (wiring, batteries, switches, etc.) of electric drive systems, as well as generation, storage and distribution of electricity. In general, theme 2 focuses on the structural design of electric propulsion systems. At the same time, in terms of theme layout, theme 2 is adjacent to theme 3 and theme 4 respectively, and the first 30 high-frequency related words partially overlap, so the three themes are similar to a certain extent.

Theme 3 is similar to theme 2 and contains terms such as fuel, diesel, motor, engine, battery, storage, etc. However, it also includes “load”, “signal”, “charging” and “bracket”. It involves the technology of output, reception and monitoring signals to control power distribution units, electrical equipment or power systems, and devices to monitor the movement of ships. In general, the theme may be: automatic devices and methods to realize real-time monitoring and synchronous operation of ship power system.

Theme 4 is adjacent to theme 2 in layout and has some of the terms of theme 2, such as generator, engine, engine, battery, unit, control, propulsion. Theme 4 is similar to theme 2. Different terms include “frequency”, “voltage”, “current” and “power”, etc., which involve the analog calculation or control of the physical quantity of power in the electric propulsion system, and have a structure device for monitoring, controlling or adjusting the physical quantity of power. Therefore, theme 4 may be about: control, adjustment and management of electric drive devices by monitoring and calculating the physical quantities of electricity.

Table 3 Top 20 high-frequency words of each topic and probability

| Theme | Top 20 high-frequency-related words | Probability |
|---------|-------------------------------------|-------------|
| Theme 1 | Electric | 0.003 |
| | Converter | 0.003 |
| | Bus | 0.003 |
| | Inverter | 0.003 |
| | Power | 0.003 |
| | Generator | 0.003 |
| | Propulsion | 0.003 |
| | Unit | 0.003 |
| | Control | 0.003 |
| | Shaft | 0.003 |
| | Drive | 0.003 |
| | Gear | 0.002 |
| | Motor | 0.002 |
| | Auxiliary | 0.002 |
| | System | 0.002 |
| | Speed | 0.002 |
| | Voltage | 0.002 |
| | Current | 0.002 |
| | Engine | 0.002 |
| | Supply | 0.002 |
| Theme 2 | Fuel | 0.003 |
| | Electric | 0.003 |
| | Circuit | 0.003 |
| | Engine | 0.003 |
| | Gas | 0.003 |
| | Motor | 0.003 |
| | Generator | 0.003 |
| | Unit | 0.003 |
| | Control | 0.003 |
| | Diesel | 0.002 |
| | Supply | 0.002 |
| | Power | 0.002 |
| | System | 0.002 |
| | Battery | 0.002 |
| | Propulsion | 0.002 |
| | Output | 0.002 |
| | Shaft | 0.002 |

(continued)

Table 3 (continued)

| Theme | Top 20 high-frequency-related words | Probability |
|---------|-------------------------------------|-------------|
| Theme 3 | Switching | 0.002 |
| | Speed | 0.002 |
| | Storage | 0.002 |
| Theme 3 | Battery | 0.003 |
| | Gas | 0.003 |
| | Distribution | 0.003 |
| | Motor | 0.002 |
| | Vehicle | 0.002 |
| | Propulsion | 0.002 |
| | System | 0.002 |
| | Electric | 0.002 |
| | Control | 0.002 |
| | Converter | 0.002 |
| | Fuel | 0.002 |
| | Diesel | 0.002 |
| | Engine | 0.002 |
| | Storage | 0.002 |
| | Module | 0.002 |
| | Voltage | 0.002 |
| | Device | 0.002 |
| | Load | 0.002 |
| | Signal | 0.002 |
| | Unit | 0.002 |
| Theme 4 | Generator | 0.003 |
| | Frequency | 0.003 |
| | Electric | 0.003 |
| | Battery | 0.002 |
| | Voltage | 0.002 |
| | Current | 0.002 |
| | Power | 0.002 |
| | System | 0.002 |
| | Motor | 0.002 |
| | Propulsion | 0.002 |
| | Control | 0.002 |
| | Engine | 0.002 |
| | Panel | 0.002 |
| | Propeller | 0.002 |

(continued)

Table 3 (continued)

| Theme | Top 20 high-frequency-related words | Probability |
|---------|-------------------------------------|-------------|
| | Energy | 0.002 |
| | Unit | 0.002 |
| | Apparatus | 0.002 |
| | Electricity | 0.002 |
| | Rotor | 0.002 |
| | Supply | 0.002 |
| Theme 5 | Electric | 0.003 |
| | Rotor | 0.003 |
| | Stator | 0.002 |
| | Propeller | 0.002 |
| | Propulsion | 0.002 |
| | Apparatus | 0.002 |
| | Shaft | 0.002 |
| | Power | 0.002 |
| | System | 0.002 |
| | Motor | 0.002 |
| | Sided | 0.002 |
| | Generating | 0.002 |
| | Core | 0.002 |
| | Generator | 0.002 |
| | Machine | 0.002 |
| | Vessel | 0.002 |
| | Battery | 0.001 |
| | Double | 0.001 |
| | Torque | 0.001 |
| | Installation | 0.001 |

In addition to the generator, motor, battery, propulsion, propeller and other power systems mentioned in the previous theme, theme 5 appeared “rotor”, “stator”, “vessel”, “torque”, “cable” and other terms. Theme 5 includes rotor, stator and torque patents, involving the ships electric propulsion system (motor or generator) design or control module, cooling of the motor propulsion system, and the method of detecting motor parts damage, there are a few is about applied to wind turbines and ship the motor (generator or motor) structure design of the motor. Therefore, we speculate that the theme is: the structural design and control test module of the motor (motor or generator) of the ships electric propulsion system.

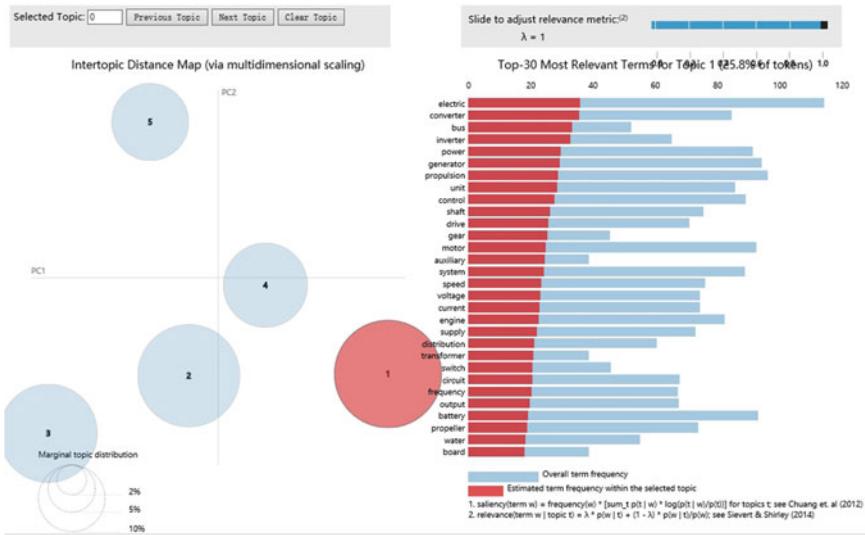


Fig. 3 Top 30 high-frequency related terms (theme 1)

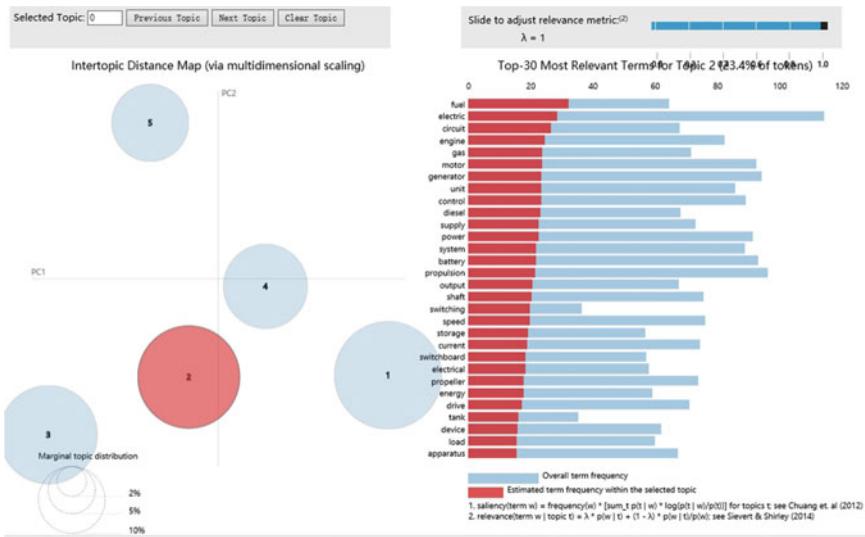
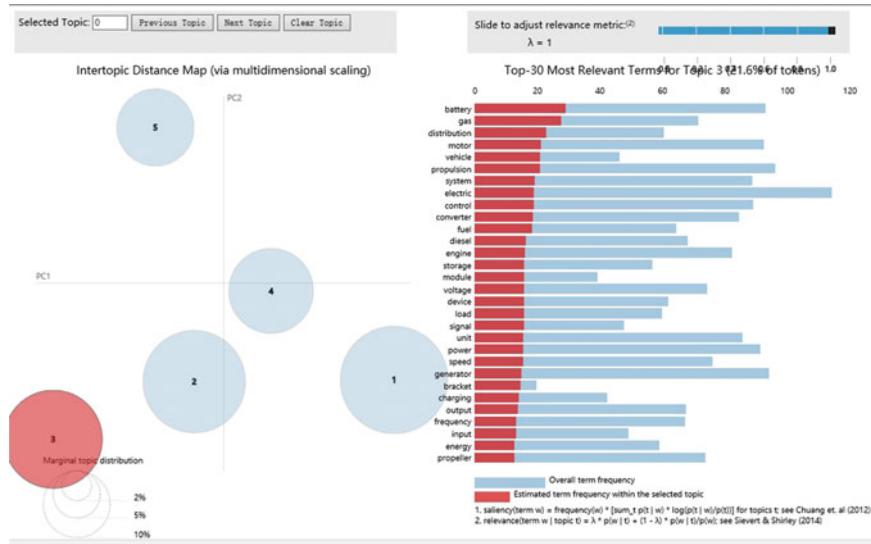
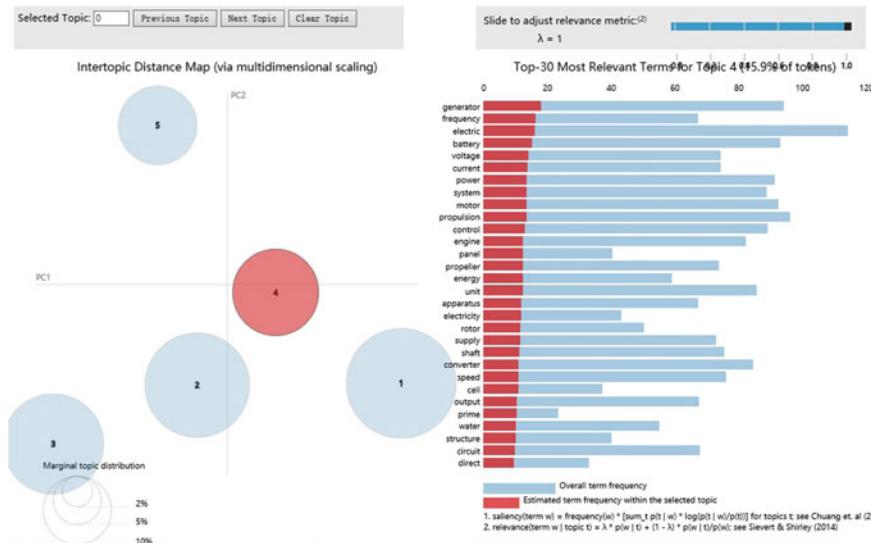


Fig. 4 Top 30 high-frequency related terms (theme 2)

5 Conclusion and Discussion

Ships integrated power system at the core of the patent technology to roughly five categories, one is the internal structure of electric propulsion system, second, the

**Fig. 5** Top 30 high-frequency related terms (theme 3)**Fig. 6** Top 30 high-frequency related terms (theme 4)

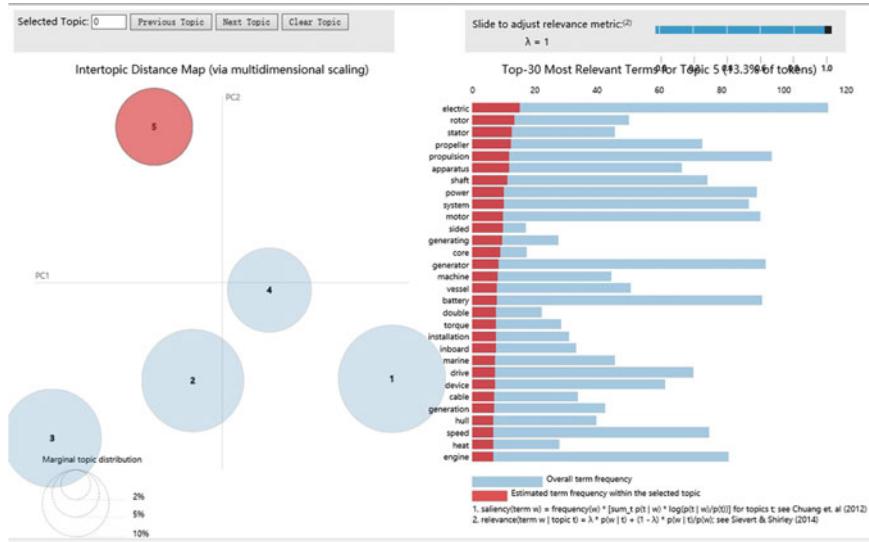


Fig. 7 Top 30 high-frequency related terms (theme 5)

structure design of electric propulsion system, three is a real-time monitoring system of electric power system and automation devices and methods of the synchronous operation, four is the drive for power control, adjustment and management, five is the structure of the machine and control detection module design. It is necessary to realize the design, control, detection and management of the electric propulsion system, to meet the automation needs of the generation and distribution of ship power, and to pay attention to the design and control of power equipment with the generator and motor as the core.

References

1. Y. Lin, Review of knowledge measurement based on patent data. *Sci. Technol. Manage. Res.* **9**, 91–93 (2008)
2. L. Yun Liu, Z.Y. Liu, Analysis of the characteristics of global carbon nanotube technology innovation based on patent measurement. *Sci. Res. Manage.* **S1**, 337–345 (2016)
3. Manual Code Lookup. <http://scientific.thomson.com/cgi-bin/mc/displaycode.cgi>
4. Y. Zhao, The writing of patent specification and its abstract. *Appl. Writ.* **09**, 25–26 (2005)
5. X. Li, C. Li, L. Li, K. Zhang, Text cluster retrieval based on LDA model. *Comput. Modernization* **6**, 7–11 (2018)

A Comparison Study on Various Continuous Integration Tools in Software Development



Sheeba, Ganeshayya Shidaganti, and Ankur P. Gosar

Abstract Continuous integration (CI) frameworks are currently the basis for a few software development projects that reduce the time and cost of the software. Most of the companies provide CI tools with similar facilities and some companies build their own CI tool. However, it does not mean that the costliest product is better than a low-cost or open-source device. Generally, all of these tools are intended to make the Product Building Software Process easy and automated by quality assurance and time reduction. In spite of this fact, ongoing integration tools have their own merits and demerits, so it is very important and sometimes difficult to choose the right CI tool for a project. The wrong choice of tools can reduce the overall flexibility and quality of the software. A few continuous integration tools, such as Jenkins, TravisCI, Codeship, Bamboo, TeamCity and CircleCI, which implement CI practices, are generally adopted by programmers, developers. The main objective of this paper is to demonstrate the analysis, usefulness and comparison of selected tools and to help in selecting the most compatible CI implementation tool that meets project requirements.

Keywords Software development projects · Continuous integration · Software project · Continuous integration framework · CI tool

Sheeba (✉) · G. Shidaganti

Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, Karnataka, India

e-mail: sheebabegum4@gmail.com

G. Shidaganti

e-mail: ganeshayayashidaganti@msrit.edu

A. P. Gosar

Intel Technologies, Bangalore, India

e-mail: ankur.p.gosar@intel.com

1 Introduction

These days numerous tasks are performed by the assistance of different programming languages and different software in every part of life. Software contribution is becoming very much important; it prompts a necessity to release products quicker to the software market, by analyzing the powerful challenges and competition of the organizations. Therefore, organizations begin to upgrade the advantage of software products by making use of Continuous Integration. There are numerous Continuous Integration frameworks or tools which can be used effectively in the product integration workflow and picking correct and appropriate tool is becoming a difficult job [1].

As indicated by Fowler [2], with this practice of Continuous Integration, developer can reduce time consumption and can save money during integration stage, according to him integration stage is normally very time-consuming and unpredictable part of software development. At the point when the product increases in size, complexity of the project is expanded, it will be difficult for developers to assure good project quality [2]. CI is a very important technique in software organizations. The main advantage of CI is that individual codes are integrated after every change in modules or subsystems [2]. Every new change submitted by a developer that is built using any programming language is repeatedly checked by determined techniques like static code analysis, automated testing, etc. Continuous Integration tools are automated with some utilities and the configuration files which can be modified depending on the project and involvement of a dedicated developer to interfere with CI process are not required [3].

There are so many tools available for Continuous Integration in the market that can be used for the integration procedure. Evidently, with these existing tools, it will be very difficult to select the best tool that suits our requirements. However, it cannot be predicted that the product which is much expensive is better than the product which is a freely available open-source tool [3]. Basically, the main aim of these tools is making the build process of software easy and with less time consumption and also providing quality of the software. Continuous Integration tools have some of their own advantages as well as disadvantages, due to this selecting effective and appropriate CI framework/tool for the project is very significant and can be difficult sometimes. The wrong choice of the tool makes the project ineffective and can reduce the overall performance of the project which decreases the software quality.

2 Continuous Integration

The main idea of using Continuous Integration in Software Development is, it essentially accelerates the complete life cycle of software, enabling developers to merge their code automatically with others code by checking whether the submitted code breaks other developers' code and creates an error in the existing code. This reduces

the manual work of the developers who need to check the code for dependencies, failure, and code breakage every time a new code base is submitted and backtracking the process to know whose code is breaking the overall software. This helps the Developer to identify the error and problems that occurred when the new code is submitted and automatically that code is not permitted for merging with another source code.

2.1 Elements of Continuous Integration System

An overall Continuous Integration begins with committing and pushing code to the repository. Any team member involved in project development can activate workflow of Continuous Integration provided they should have authentication. The Elements of CI system are illustrated in Fig. 1

Continuous Integration system [3] consists of the following steps:

(1) Software Developer commits and pushes the code base to the repository. (2) CI Server checks the repository for modifications in the existing code using some configuration files. (3) If any modifications are noticed by CI server within the repository, then this server fetches the latest form of source code from the repository, and then runs the build on the code. Configuration or utility Scripts, liable for integration is developed. (4) Then, Integration server directs the result of build to the developers by messages, mails, etc. who are involved in the project. (5) Later, server performs

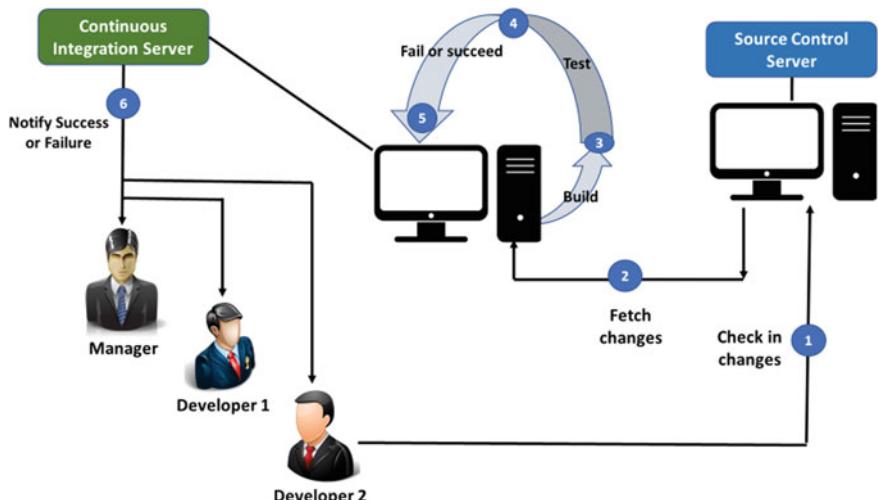


Fig. 1 Elements of continuous integration system

the same cycle from step 2 and performs the workflow as in the above steps until new modifications or code is pushed to the repository.

Each Element description of Continuous Integration System is shown below [3].

- Main task of **developers** is to develop and run code. Every developer must run some private test cases on code after developing it. If integration happens successfully with others code, then the developer can push his code to repository. If the code breaks the existing build, then it prevents the integration of new source code with an existing repository.
- **Repository** plays a vital role in CI system. It's also referred to as Software Configuration Management. Normally, all software projects have repository, some projects have their own repository, and some have master repository which is common to all the similar projects. Repository is mainly used to prevent merging existing code with erroneous code.
- **Continuous Integration Server** is the most essential part of a complete system. This Integration Server produces the latest build by integrating with other builds when developer makes changes in the repository. Generally, CI Server is configured to check if any changes are made in repository. Repository can notify CI server that a build has to be created. CI server gets source code, configuration file, recipe file of software project and runs build scripts.
- **Feedback Operation** gives the developer a way to make changes quickly. Whenever the code which is pushed to the repository is failed then this feedback operation sends feedback to the developer in the form of mail, message, or any other mode available so that he can fix the error.

Process of Integration Duvall [3] in his research “Continuous Integration: improving software quality and reducing risk” states that to implement CI four properties is necessary. First, CI server has to be connected to the repository. Second, script has to be built. Third, providing feedback operation. Fourth, modifies source code are integrated by CI server. These features are very essential for a CI system to perform efficiently.

Integration process components are as follows

- **Compilation of Code** performs an important role in development life cycle [3]. It is a method of converting source code to human-understandable and readable format. Compilation of source code depends on some parameters like programming language used in the development of project, debugger tool, etc. Compilation process behaves differently for different programming approaches. In case of dynamic programming approach, no binary code is created but compilation occurs by strictly checking the code.
- **Database Integration** database is an important part of any software application. It provides integration of databases that are linked for different projects. There are three major key principles in Continuous Integration as listed below:
 - After the developer makes any changes, DB Integration verifies it for the latest state in database.

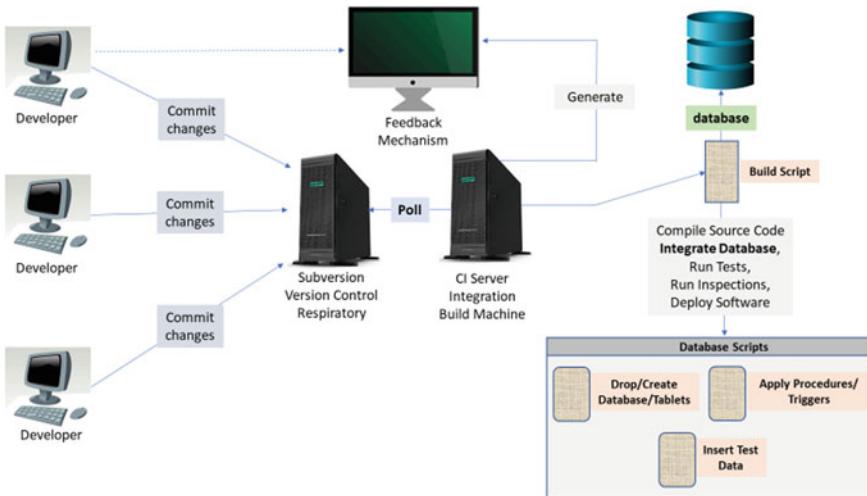


Fig. 2 Database integration

- Once the changes are made, a test is run by the server.
- If any changes are made in target database, it updates the repository by combining those changes with the existing one.

Starting from Deployment phase to Production phase some of the problems might not be identified by the DB admin but Continuous Integration Server detects it. Figure 2 shows Database Integration process in which CI enables DB integration in an ongoing build.

- Another important aspect of this system is **testing**. Developer's runs *Unit test* cases to checks the dependability and consistency of their code, developers perform *component tests* to identify the search flaws and to check whether the correct functionality is provided by the application. Functional requirements as well as non-functional requirements of whole system are checked by running *System tests*. To check the functionality achieved by the different functions, *Functional tests* are used by the developers. Functional tests can be used at each testing level.
- **Inspection** based on some pre-determined protocols. This pre-determined protocols and automated source code check enhances the overall software quality. For example, developers might set the protocol for a project that class size should not be more than one thousand lines. If any duplication occurs in the source code, then it can be identified by checking the code using Inspection. Along with this, inspection helps to retain the architecture safe and unbroken. The time taken to identify the problem and recover the problem is mitigated by inspection.
- After the code is inspected now it has to be **deployed**. Deployment occurs in six stages. First stage is elements of repository label which is mainly performed at the last stage of project. Second stage is preserving neat environment, which indicates that better condition is maintained for making the integration successful. This can

be achieved by deleting some recipe files, configuration files, and servers from the integration framework. Third stage is designing a build from repository and then installing this on a local computer. If fourth stage developers run complete model tests and automated tests to become more confident about the modifications, they make in the code. Fifth stage is producing the developer a way to make changes quickly. Whenever the code which is pushed to the repository is failed then this feedback operation sends feedback to the developer in the form of mail, message, or any other mode. Sixth stage provides the possibility to roll back committed build. If these stages are performed well then it will result in effective deployment.

3 Different Continuous Integration Tools

3.1 Jenkins

Jenkins advanced from a Continuous Integration framework to Continuous Delivery Platform, by approving the latest design in which the build, release, and delivery process of a product is automated. Jenkins is a free open source continuous integration built with Java. For the previous few years, Jenkin embraces the high position because this tool is freely available, open-source and modular. It can be used in build, release, and test software projects continuously, by making the integration changes easier [4]. One good thing about Jenkins tool is the configuration files of build are on the disk, which provides huge build cloning and easy reconfiguring. However, Jenkins has merits, it also has some demerits: The language which is used to build the User interface is old and it is not user friendly as compared to the latest UI trends. Yet it is an open-source many people don't prefer to implement Jenkins as its installation and configuration is difficult. Whenever Continuous integration of any project break or fail due to some small changes in its settings [5].

Jenkins properties are [6]: (1) It can be easily installed, upgrade, and can be configured easily. (2) Builds can use parallelly. (3) External ongoing builds can be monitored efficiently. (4) Tool environment can be customized based on the project requirement. (5) Once the build is committed, this tool does not provide other users in Github to make any changes to it. (6) Only those users have authentication of those who have faceless accounts. (7) Jenkins tool offers API functionalities to access remotely. (8) Supports easy Continuous Integration as well as Continuous Deployment for large projects. (9) Developers can add or modify extensions accordingly. (10) This tool is Companionable with different programs like Libvirt, Kubernetes, Docker, etc.

3.2 *Travis*

Continuous Integration, a normally received practice in Software Development [7]. CI expects to assure that every modification in the product framework can be filtered through regular inspections. CI administrations like Travis CI¹ incorporate with GitHub² to encourage this procedure by permitting clients to content form schedules for their project. At whatever point when modified code is pushed or pulled based on the received request, commitment is verified by configuring systems as well as implementing manufacture contents as communicated in Travis. Results of integration whether the build is failed or success will be informed to clients by means of mails, messages, etc. Travis tool is facilitated, appropriated Continuous Integration administration that is used in GitHub to construct and test programming ventures. Travis is freely available on GitHub, also paid service for personal or corporate repositories. Travis is one of the major well-known Continuous Integration stages as per the latest analysis.

Properties of Travis (1) Travis helps users to monitor the projects in GitHub. (2) Executes Test as well as generate rapid results. Users can execute tests parallelly. (3) Build objects, checks for quality of code. (4) Deployment of build to cloud services is easy. (5) This tool recognizes both minor as well as major changes in source code. (6) Travis tool provides Developers to watch the tests even in running state.

3.3 *Codeship CI*

Codeship CI is used by web applications as Continuous Integration tool as well as Continuous deployment tool. Like other tools Codeship also makes the code tests run automatically as the code is modified in the repository. scaling and managing the infrastructure is done efficiently, so it becomes easier for the developers to submit code and get feedback on whether the build is failed or a success in the Development process [8].

Codeship has some following properties [9] (1) Test cases of code are run automatically. (2) Setting up Codeship tool with Github is easy. (3) When the code is pushed to the repository it initiates tests automatically. (4) pipelines setup for deployment is easy and this setup can be done many times in a day whenever setup is necessary. (5) Team gets notified every time when any team member makes any modification in the repository. (6) Prevents the user to access Repository who doesn't have authorization.

3.4 Bamboo CI

Bamboo CI tool also is known as a commercial tool that is developed through Atlassian for Continuous Integration; this tool is freely available for any open-source projects. Bamboo tool links automated builds automated tests as well as releases of a project as a single workflow. The best JIRA integration, Top-class support for deployments, Automated merging of modules, CL & CD pipelines are flexible, customizations without interrupting the process, Effective build agent management, Extraordinary support, and freely available resources, Faster import from Jenkins, Basic and unconstrained drag and drop web interface without a lot of complexity or effort. Jobs can be hauled or dropped between stages, stages, and undertakings can be hauled or dropped to change the request for assembling, on the off chance that what you're attempting to do doesn't bode well, at that point pop up warnings can be shown. Bamboo consequently identifies new branches and manufactures them. It is feasible for a similar pipeline to carry on contrastingly when the branch is built. Bamboo tool is used by companies like LeapFrog, National Public Radio, Cisco, Kaiser Permanente, BMW, NASA [9].

Bamboo offers the following properties

(1) Batch examinations can be run parallel. (2) Bamboo CI tool set up is very simple. (3) Per-environment approvals properties permit developers to deploy the source code to local repository. (4) When the code is pushed or modified in the repository it initiates build automatically. (5) This tool can be obtained as installed as well as hosted and runs on local repository. (6) Enables present collaboration as well as integrated with HipChat. (7) It has some Built-in workflows and branching of Git. It naturally integrates the branches.

3.5 TeamCity

TeamCity is a multifunctional Continuous Integration tool. This tool is prepared to work as soon as it is installed. TeamCity supports numerous frameworks, verification, deployment as well as code test beyond thinking. TeamCity is extensible, for some activities it doesn't have to support Java. TeamCity setup is easy and for small project teams and open source projects, it is freely available. The most common property in TeamCity is the automation of various features of life cycle of software development is from build phase till acceptance phase. However, it has some shortcomings which is discussed in [10] that some of the reviewers will not be satisfied with the installation procedure, TeamCity setup and making it compatible is also very complex.

Following are the TeamCity Properties (1) It delivers many ways to reuse the configuration files of the parent project to subproject. (2) Build configuration is very flexible. (3) It includes building source code with different targets, shell scripting, etc. (4) TeamCity as a good CI tool for development teams of any size. (5) We can

run builds parallel and simultaneously in different environments. (6) Structured text can make the server to perform some actions Build status. (7) This CI tool suits the companies with different environments and services, since it can be extended to fit the projects. (8) Testers can be replaced with build agents to check the ongoing builds.

3.6 *CircleCI*

Setting up CircleCI server and maintaining it is very easy without many complications [11]. Circle CI tool simplicities building software, testing source code, and deploying software rapidly and thoroughly over other platforms. CircleCI makes the procedure of Continuous Integration basic and simple for any organization [12]. CircleCI is a very supportive CI framework, some of the reviewers assert it as a quicker tool than other tools like Travis and simple than CI tool Jenkins. On the opposite side, there are a few drawbacks. For example, there were a few situations where tests were broken as a result of CircleCI updates. In spite of the fact that it can break the code results on its update, yet it could be best if it defines all the environmental variables. Sometimes tests can't be passed because some additional configurations are required. CircleCI notifies to the team members who are involved in that project when a push is performed

Properties of CircleCI (1) Permits to select the Environment on which the developer wants to run the build. (2) Supports many languages like including JavaScript, C++, PHP, .NET, Python, and Ruby. (3) Support for Docker to configure customized environment as per the requirement. (4) Automatically cancels the build which is queued or running when a new build is activated. (5) It reduces the overall build time by splitting and balancing the tests across several containers. (6) It prevents non-admins from doing any changes to critical project settings.

4 Comparison of Continuous Integration Tools

This paper presents the comparison of different Continuous Integration tools as shown in Table 1. As a result, the option was not that easy as Jenkins as well as Codeship offer many valuable services and essential features. However, Jenkins was chosen over several other tools because it offers a stable budget and a start-friendly collection of CI tools. Second, it's an open-source project for free. Third, there are over 1000 plug-ins that enable us to build suitable configuration by user. Fourthly, it is a great start to learning which requirements are most important and essential, and then its good services, it has a free plan with reduced functionalities with open-source projects.

Table 1 Comparison table for continuous integration tools

| Parameter | Jenkins | Travis | Codeship | Bamboo | TeamCity | CircleCI |
|----------------|---|-------------------------------------|---|---|---|--|
| Cost | Free, expense is required for militance | \$129 per month | Starts for free then \$75 per month | Paid tool, price depend on the usage | \$299 per month | \$30 per month for Custom projects \$300 per month |
| Performance | Very high | Best for open source projects | High performance | Better performance | Higher than Bamboo | High performance |
| Time to Set up | Takes more time | Takes very less time | Takes only few minutes | Takes few minutes | Takes some time | Takes less than 20 s |
| Open source | Yes | Yes | Yes, but for some time | No | Free for small teams and open source projects | Yes, but for some builds |
| Github | Good for Github | Excellent for Github | Need to install the Codeship GitHub App | Excellent for JIRA, Stash and rest of Atlassian suite | Good for Github | Excellent for Github |
| Usage | Easy to use | Flexible to use | Easy and Simple | Flexible to use | Very flexible to use | Easy to use |
| Support | Wide support from the community. | Limited support from the community. | Limited support from the community | Very less support | Wide support from the community | Limited support from the community |

(continued)

Table 1 (continued)

| Parameter | Jenkins | Travis | Codeship | Bamboo | TeamCity | CircleCI |
|----------------------|------------------------------------|--|--|---|--|--|
| Usage plans | Free | Freely available for open source projects | Free for trial and small projects | Freely available for open source projects | Free for 100 builds and three build agents | Free for starting 1000 builds |
| Customization option | More | Less | More | Less | Less | Less |
| Control on system | Full | Very less | No | Very less | Very less | Very less |
| Server machine | Server-based | Cloud-based | Server-based | Cloud-based | Cloud-based | Cloud-based |
| Compatibility | Highly compatible | Not much compatible | Not much compatible | Compatible with Android devices and iPhone | Based on compatibility requirements | compatible |
| Languages supported | Python, Ruby, Java, Android, C/C++ | Android, C, C#, C++, Java, JavaScript, PHP, Python, Ruby | Java, JVM, JavaScript with NodeJS, PHP, Python, Ruby, Dart | Bamboo supports builds in any programming language using any build tool | C#, C++, JavaScript, Pearl, PHP, Python | Android, iOS, Java, JavaScript with NodeJS, Python, Ruby |

5 Conclusion

The purpose of this paper is to provide an overview of CI Tools and to compare them. All of these tools are designed to facilitate and automate the development of software by ensuring quality and time reduction. The selection of a good CI tool for a project is critical and sometimes difficult. The wrong selection of the tool can decrease overall flexibility and software quality. The survey results are going to enable some developers to make a tool decision or at least to acquire some knowledge about the tools in software development. On the basis of the comparison, Jenkins has been preferred over other tools, since it is a free open source project which offers great flexibility for many different development methodologies. The system of continuous integration suits the needs identified by a majority.

References

1. A. Phillips, M. Sens, A. De Jonge, M. Van Holsteijn, The IT Manager's Guide to continuous delivery: delivering business value in hours, not months. XebiaLabs (2015)
2. M. Fowler, M. Foemmel, *Continuous Integration*. Thought-Works Inc. (2006). (Online). Accessed in May 16, 2018 from <http://www.martinfowler.com/...html>
3. P.M. Duvall, S. Matyas, A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*. Pearson Education (2007)
4. V. Armenise., Continuous delivery with Jenkins: Jenkins solutions to implement continuous delivery, in *2015 IEEE/ACM 3rd International Workshop on Release Engineering* (IEEE, 2015, May), pp. 24–27
5. N. Pathania, *Learning Continuous Integration with Jenkins: A Beginner's Guide to Implementing Continuous Integration and Continuous Delivery using Jenkins 2*. Packt Publishing Ltd. (2017)
6. D. Polkhovskiy, Comparison between continuous integration tools. Master's thesis, 2016
7. M. Beller, G. Gousios, A. Zaidman, Oops, my tests broke the build: an explorative analysis of Travis CI with GitHub, in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)* (IEEE, 2017, May), pp. 356–367
8. W.E. Hart, J.D. Siirila, Some perspectives on Testing and Continuous Integration for Open Source Software (No. SAND2018-12452PE). Sandia National Laboratories (SNL-NM), Albuquerque, NM, USA (2018)
9. R. Varga, Changing Dashboard build system to Bamboo (No. CERN-STUDENTS-Note-2013-135) (2013)
10. V. Melymuka, *TeamCity 7 Continuous Integration Essentials* (Packt Publishing Ltd, 2012)
11. J.M. Belmont, *Hands-On Continuous Integration and Delivery: Build and release quality software at scale with Jenkins, Travis CI, and CircleCI* (Packt Publishing Ltd, 2018)
12. E.H. Kim, J.C. Na, S.M. Ryoo, Implementing an effective test automation framework, in *2009 33rd Annual IEEE International Computer Software and Applications Conference*, vol. 2. (IEEE, 2009, July), pp. 534–538

A Holistic Study on Approaches to Prevent Sexual Harassment on Twitter



Aishwariya Rao Nagar, Meghana R. Bhat, K. Sneha Priya,
and K. Rajeshwari

Abstract Sexual harassment is a serious social illness that has seeped into our digital societies as well. Research on sexual harassment in social media has been gaining popularity of late, due to the genuine long-term effects such harassment can have on an individual like depression, withdrawal, loss of self-confidence, self-destructive tendencies, and propagation of a conducive environment for sexual assault. The approaches to tackle this issue using technology range from simple lexicon-based models to sophisticated machine learning models. Lexicon-based systems utilize dictionaries that contain offensive words that are characteristic of sexual harassment. Rule-based approaches need to be supplemented with logical reasoning in order to define rules which can be matched with the text to distinguish sexual harassment. Machine learning approaches develop predictive models that can detect sexually offensive messages. This paper holistically studies approaches used to solve the problem of classification of sexual harassment in cyberspace and aims to determine the best directions for future research in this field. It also studies the methods of analyzing the social network in terms of the structure of ties between users that can be used to study specific user patterns and characteristics.

Keywords Sexual harassment · Twitter · Lexicon-based systems · Rule-Based Systems · Machine learning approaches · Social network analysis

A. R. Nagar (✉) · M. R. Bhat · K. Sneha Priya · K. Rajeshwari
B.M.S. College of Engineering, Bengaluru, India
e-mail: aishwariya.rao217@gmail.com

M. R. Bhat
e-mail: meghanarbhat@gmail.com

K. Sneha Priya
e-mail: kannansneha.1997@gmail.com

K. Rajeshwari
e-mail: rajeshwarik.ise@bmsce.ac.in

1 Introduction

Sexual Harassment is a behavior characterized by the making of unwelcome and inappropriate sexual remarks, physical advances in a workplace, other professional, or social situations. The anonymity that the Internet grants its users helps in sustaining predators. This is especially true for a social networking service like Twitter which does not require any form of authentication of a user to sign up and sells the idea of a virtual persona via pseudo or fake names and is an important feature for some of its users [1]. In this paper, we attempt to holistically study the ways of detecting sexual harassment on Twitter and related methods of social network analysis to aid in predicting common characteristics of both abusers and their targets on the platform. Detecting harassment has a varied number of approaches, with simple Word-List based methods to more complex Rule-Based and Machine Learning based approaches. We also study the nature of the Twitter network and examine the metrics that aid in understanding the significance of observed topological features in making deductions about a specific user or a group of users.

2 Literature Survey

Online harassment detection is in nascent stages on all major social media platforms, if not non-existent. There are three broad categories of online harassment detection, namely: wordlist-based approach, rule-based approach, and machine learning approach.

2.1 *Wordlist-Based Approaches*

Xu and Zhu [2] proposed an advanced lexicon-based approach to filter out offensive language in online content. They focus on an ideology that aligns with the freedom of speech: removing offensive parts of a sentence without removing the entire sentence. The challenge of this approach is maintaining readability while retaining semantic correctness. Chen et al. [3] proposed architecture to detect offensive content and identify potential offensive users. They introduce a qualitative element to this classification problem by determining the degree of offensiveness in online content. Bretschneider et al. [4] proposed a pattern-based approach, which optimizes a simple lexicon-based solution using a person identification module. This encompasses the necessity of the existence of a victim in order for an act to be termed harassment.

2.2 Rule-Based Approaches

The more advanced version of harassment detection is the rule-based technique which analyzes semantic relation in the text. This technique involves the process of integrating a rule engine and knowledge database with the classifier and investigating the effect on the performance. These methods can detect sexual harassment containing implicit knowledge. Despoina et al. [5] proposed a technique to extract text, user and network attributes to identify aggression and bullying online.

2.3 Machine Learning Based Approaches

Machine learning based approaches have come up in recent times. These approaches learn the rules of classification automatically by discerning patterns in sexual harassment messages. Machine learning approaches provide higher classification performance compared to wordlist-based approaches. Khatua et al. [6] propose a model that can address the lack of crime statistics available in order to design and implement stricter sexual harassment prevention policies. Their solution helps to identify and probe deeper into nature as well as the severity of risk associated with sexual violence. Their model was trained using a dataset that comprised of around 0.7 million tweets containing the #MeToo hashtag during the #MeToo movement that took various social networking platforms by storm around October 2017. The authors used 4 different deep learning techniques namely—Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (Bi-LSTM). Goenaga et al. [7] present an approach to automatically identify misogyny in tweets using neural networks. The task is one of the two sub-tasks organized by AMI-IberEval 2018 organization. AMI stands for Automatic Misogyny Identification. Also, a focus is on RNN approach using a Bidirectional Long Short-Term Memory (Bi-LSTM) with Conditional Random Fields (CRF) and it evaluates the proposed architecture on misogyny identification task (text classification) (Table 1).

Huang et al. [8] describe an approach that aims to improve the accuracy of methods that identify cyberbullying messages by integrating textual features that use keywords with social network features. They then characterized messages of harassment or bullying based on their content. The ‘SMOTE’ (Synthetic Minority Oversampling TTechnique) approach was then used to generate a balanced data set to train the model, before testing it in realistic imbalanced settings. Pitsilis et al. [9] proposed an RNN based solution for hate speech detection on Twitter. The proposed model employs LSTM, a class of Recurrent Neural Network, and was optimized by (1) including certain user-related features characterizing the users’ tendency to be racist or sexist and (2) combines the output by various LSTM classifiers. Yin et al. [10] developed a solution along three dimensions—content, sentiment, and context. These features are used for training an SVM.

Table 1 Comparison of various approaches and models

| Paper | Method | Precision | Recall | Accuracy | Other salient features |
|-------|---|---|---|---|---|
| [3] | Lexicon and Syntactic Based Approach (Advanced text mining and NLP approach) | Sentence offensive detection 0.982 User offensive detection 0.9434 | Sentence offensive detection 0.9434 User offensive detection 0.779 | | |
| [4] | Pattern Based Approach | | | | Pattern Based: F: 0.72 Wordlist Based: F: 57 ML Based: F: 0.63 |
| [5] | Rule Based Approach | Without Spam Removal 0.716 With spam removal 0.899 | Without Spam Removal 0.7332 With spam removal 0.917 | Without Spam Removal 0.7345 With spam removal 0.9108 | Without Spam Removal-Kappa: 0.4717 RMSE: 0.3086 |
| [6] | CNN LSTM Bi-LSTM MLP | | | CNN 0.83 LSTM 0.82 BiLSTM 0.81 MLP 0.77 | |
| [8] | J48, Naive Bayes, SMO, Bagging, Dagging | 0.9991 | 0.9972 | | F: 0.9981 |

(continued)

Table 1 (continued)

| Paper | Method | Precision | Recall | Accuracy | Other salient features |
|-------|---|--|--|--|---|
| [11] | Lexicon + Sentiment Analysis + NLP + Deep Neural Network (Bi-GRU) | | | Kaggle without sentiment 0.932 Kaggle with sentiment 0.937 Subversive Kaggle without sentiment 0.772 Subversive Kaggle with sentiment 0.801 Wikipedia without sentiment 0.881 Wikipedia with sentiment 0.885 Subversive Wikipedia without sentiment 0.814 Subversive Wikipedia with sentiment 0.820 Reddit without sentiment 0.942 Reddit with sentiment 0.943 Subversive Reddit without sentiment 0.830 Subversive Reddit with sentiment 0.839 | CNN F: 0.68 LSTM F: 0.70 BLSTM F: 0.70 SVM F: 0.41 LR F: 0.44 |
| [12] | LSTM Bi-LSTM CNN | CNN 0.75 LSTM 0.75 BLSTM 0.69 SVM 0.37 LR 0.39 | CNN 0.75 LSTM 0.71 BLSTM 0.73 SVM 0.47 LR 0.49 | CNN F: 0.68 LSTM F: 0.70 BLSTM F: 0.70 SVM F: 0.41 LR F: 0.44 | Kongregate F: 0.442 SlashDot F: 0.298 MySpace F: 0.313 |
| [10] | TF-IDF +SVM | Kongregate 0.352 SlashDot 0.321 MySpace 0.417 | Kongregate 0.595 SlashDot 0.277 MySpace 0.250 | | |

Brassard-Gourdeau et al. [11] propose a method to implement a toxicity detection tool that uses sentiment analysis to prevent subversion by users who circumvent the system by modifying toxic content to fool the system's filters. Marwa et al. [12] used a machine learning based approach to classify sexual harassment. Their proposed solution includes three different multi-layered neural networks, LSTM, BiLSTM, and CNN and compares their performances. Input is fed to the networks by making use of a concept known as word embedding. A specialized embedding technique, the Pennington GloVe models [13], and Word2vec [5] were used, to preserve the semantic and syntactic information of the input data.

Ibrahim et al. [14] present a method of identifying and classifying toxicity in online content. Their method involves building two separate classifiers. Firstly, a binary classifier determines whether a given comment has toxic content. After that, a second classifier identifies the toxicity class from the comments identified as toxic. The latter is a multi-class problem (with six classes). This dataset is highly imbalanced and to overcome this problem, the authors apply a data augmentation technique, wherein new samples of the rare class are generated using the existing data (Unique Word Augmentation, Random Mask, and Synonyms Replacement).

Pendar et al. [15], the authors work to identify the feasibility of using text categorization to identify sexual predators online. Machine learning models are used to distinguish between the child side and the pedophile side of text chat.

2.4 Social Network Analysis

Social Network analysis is the quantitative and qualitative study of social structures using graph theory that can be a useful approach to recognize patterns of users since every user generates a network on the platform.

Myers et al. [16] study the properties of twitter interactions to determine if Twitter is an information network or a social network, along with providing a characterization of Twitter's geometrical and spatial attributes. From the graph of vertices that had a degree of at least 1, the largest weakly connected component contained 99.94% of all vertices and the largest strongly connected component only contains 68.7% of all active users and more than 30% of active users did not have a single mutual edge. It was concluded that unreciprocated edges were a sign of an information network rather than a social network. Colombo et al. [17] study the patterns of communication and the connectivity of users on Twitter whose posts were annotated by people as a form of suicidal ideation. This paper is described to understand the approach to analyze social network characteristics of specific groups of users and can be extended to study the nature of both sexual harassment victims and abusers on Twitter. Chowdhury et al. [18] attempt to automate the process of collection of tweets containing recollections of sexual harassment by manual annotations for the tweets labeling as 1 for Recollections and 0 otherwise. They were explicitly a persons' recollection of an incident, which could be categorized into awareness related, tweets describing one's emotional experiences of an assault, or media reports.

3 Research Gap

There is no single comprehensive dataset that fully encompasses the vocabulary of an abuser. Lack of existing legitimate datasets results in the need to mine training data from Twitter. This is typically done by indexing a set of well-known keywords or hashtags associated with sexual harassment. The set of keywords is subjective to the miner and hence, no set can be exhaustive. The quality of the resulting dataset depends heavily on the suitability of the keywords selected to mine. Some researchers propose creating code-switched data artificially by recurrent neural network decoder [19]. Anchoring and monolingual Language ID are proposed techniques by [20] to collect code switched data. The task of analysis of such data starts with language identification, followed by a process called normalization [21]. Subsequently, non-English words must be translated into corresponding English words [22]. Multilingual sentiment classification approaches like proposed by Mihalcea et al. [23] require parallel corpora for each individual language separately. The most frequently observed source of data for researchers using Twitter data is what they mine using Twitter for Developer's APIs. Thus, the model that learns from crowdsourced labeling is heavily skewed towards the decision of the individual labeling the tweets. Such bias is intangible and hard to quantify because "more complex and controversial in multicultural environments where culturally-derived values and beliefs serve as norms that determine when certain behaviors and feelings are appropriate and when they are not" [24]. Much of the social studies that probe into sexual harassment and abuse on Twitter focus on Western countries and there is an alarming lack of such statistics from the rest of the world.

4 Conclusions

Overt sexual harassment in social media continues to be one of the most pervasive issues in our society. The three main approaches for solving this problem are the wordlist-based approach, the rule-based approach, and the machine learning approach. Word-list based approaches, in its primitive form, fail to give good results for a problem that distinguishes between crude profanity and intended aggressive behavior. Rule-based approaches give the flexibility to characterize observed patterns of behavior in users, which are found to be symptomatic of sexual harassers. These, when compounded with wordlists, achieve a good semantic understanding of sexual harassment. Machine learning leverages the power of computers to identify language features that define harassment. One of the important parameters influencing the accuracy of these techniques is the dataset used. The false-positive rate must be kept in check due to the grave consequences of falsely flagging out harassment. Research shows that these can be achieved by carefully adjusting the parameters and choosing the appropriate learning model.

References

1. S.T. Peddinti, K.W. Ross, J. Cappos, On the internet, nobody knows you're a dog, in *Proceedings of the Second Edition of the ACM Conference on Online Social Networks—COSN '14* (2014)
2. Z. Xu, S. Zhu, Filtering offensive language in online communities using grammatical relations, in *Proceedings of The Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS'10)* (2010)
3. Y. Chen, Y. Zhou, S. Zhu, H. Xu, Detecting offensive language in social media to protect adolescent online safety, in *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy Security Risk and Trust ser. SOCIALCOM-PASSAT '12* (2012), pp. 71–80
4. U. Bretschneider, T. Wöhner, R. Peters, Detecting online harassment in social networks (2014) (online). <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1003&context=icis2014>. Accessed 20 Mar 2015
5. D. Chatzakou, N. Kourtellis, J. Blackburn, E.D. Cristofaro, G. Stringhini, A. Vakali, Mean birds: detecting aggression and bullying on twitter (2017). CoRR, vol. abs/1702.06877
6. A. Khatua, E. Cambria, A. Khatua, Sounds of silence breakers: exploring sexual violence on twitter, in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2018)
7. I. Goenaga, A. Atutxa, K. Gojenola, A. Casillas, A.D. de Ilarrazo, N. Ezeiza, M. Oronoz, A. Pérez, O.P. de Vinaspre, Automatic misogyny identification using neural networks, in *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018). CEUR Workshop Proceedings*. CEUR-WS.org, Seville, Spain (2018)
8. Q. Huang, V.K. Singh, P.K. Atrey, Cyber bullying detection using social and textual analysis, in *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia* (2014), pp. 3–6
9. G.K. Pitsilis, H. Ramampiaro, H. Langseth, Detecting offensive language in tweets using deep learning (2018)
10. D. Yin, Z. Xue, L. Hong, B. Davison, Detection of harassment on Web 2.0, in *The Content Analysis in the Web 2.0 Workshop* (2009)
11. É. Brassard-Gourdeau, R. Khouri, Impact of sentiment detection to recognize toxic and subversive online comments (2019). arXiv preprint [arXiv:1812.01704](https://arxiv.org/abs/1812.01704)
12. T. Marwa, O. Salima, M. Souham, Deep learning for online harassment detection in tweets, in *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)* (2018)
13. J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation. *EMNLP* **12**, 1532–1543 (2014)
14. M. Ibrahim, M. Torki, N. El-Makky, Imbalanced toxic comments classification using data augmentation and deep learning, in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (2018)
15. N. Pendar, Toward spotting the pedophile telling victim from predator in text chats, in *Proceedings of the First IEEE International Conference on Semantic Computing* (2007), pp. 235–241
16. S.A. Myers, A. Sharma, P. Gupta, J. Lin, Information network or social network? The structure of the twitter follow graph, in *Proceedings of the 23rd International Conference on World Wide Web* (ACM, 2014), pp. 493–498
17. G.B. Colombo, P. Burnap, A. Hodorog, J. Scourfield, Analysing the connectivity and communication of suicidal users on twitter. *Comput. Commun.* **73**, 291–300 (2016)
18. A.G. Chowdhury et al., # YouToo? detection of personal recollections of sexual harassment on social media, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019)
19. H. Adel, N.T. Vu, K. Kirchhoff, D. Telaar, T. Schultz, Syntactic and semantic features for code-switching factored language models. *IEEE Trans. Audio Speech Lang. Process.* **23**(3), 431–440 (2015)

20. G. Mendels, V. Soto, A. Jaech, J. Hirschberg, Collecting code-switched data from social media, in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)* (2018)
21. F. Liu, F. Weng, X. Jiang, A broad-coverage normalization system for social media language, in *Proceedings of the 50th Annual Meeting Association for Computational Linguistics: Long Papers* (2012), pp. 1035–1044
22. P. Mathur, R. Shah, R. Sawhney, D. Mahata, Detecting offensive tweets in hindi-english code-switched language, in *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media* (2018), pp. 18–26
23. R. Mihalcea, C. Banea, J. Wiebe, Learning multilingual subjective language via cross-lingual projections, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague (2007), pp 976–983
24. R.S. Lazarus, S. Folkman, *Stress, appraisal, and coping* (Springer, New York, 1984), p. 165

Cyber Bullying Detection Based on Twitter Dataset



Debajyoti Mukhopadhyay , Kirti Mishra, Kriti Mishra, and Laxmi Tiwari

Abstract The acceleration of different social media platforms has alternated the way people communicate with each other it has also ensued in the rise of Cyber-bullying cases on social media that has various adverse effects on an individual's health. In this project, we aim to build a system that tackles Cyber bully by identifying the mean-spirited comments and also categorizing the comments into peculiar division. The target of developing such a system is to deal with Cyber bullying that has become a prevalent occurrence on various social media. The system uses two noticeable features—Convolutional Neural Network and Long Short-Term Memory which improves the efficiency of the system.

Keywords Cyber bullying detection · CNN algorithm · Twitter · Social media harassment · Online harassment · Long short-term memory · Word embedding

1 Introduction

Social media is the use of virtual platform for connecting, interacting, sharing of contents and opinion around the globe. Since the development of social platform, its usage by teens and adults across the globe has seen great upsurge. The most famous

D. Mukhopadhyay (✉) · K. Mishra · K. Mishra · L. Tiwari
Computer Science Department, Mumbai University, Mumbai, Maharashtra, India
e-mail: debajyoti.mukhopadhyay@gmail.com

K. Mishra
e-mail: mishrakirti2403@gmail.com

K. Mishra
e-mail: kritis mishra41@gmail.com

L. Tiwari
e-mail: laxmitiwari21998@gmail.com

D. Mukhopadhyay
WIDiCoReL Research Lab, Mumbai, Maharashtra, India

social media platforms are Facebook, YouTube, Twitter, Instagram. The paramountcy of it can be implied by the fact that out of 7.7 billion people in the world at least 3.5 billion people are online. Social media had a vital role in connecting people and eventually made the world a smaller well-connected and more tolerant place. But let's not blow the fact that social media is a double-edged sword and can have a various detrimental effects which can be of serious ramification like circulation of rampant information to cyber bullying.

Cyber bullying is a kind of bullying that occurs over digital devices that include phones, laptops, computers, tablets, netbook, hybrid through various SMS, apps, forums, gaming which are intended to hurt, humiliate, harass and induce various negative emotional responses to the victim, using text, images or videos and audios. It can cause more suffering than traditional bullying as the atrocious messages are perpetual and easy to prey on potential victims. Cyber bullying can result in low self-esteem in victims as constant mean messages can result in victims being more anxious and insecure about themselves. It may result in poor performance in school grades among teenagers. Teenagers and kids who are unable to cope up with bullying may result in social isolation by skipping school and interaction with friends and family, and also indulge in activities like drugs and alcohol. It also affects adults in prosaic day to day life activities. Victims of cyber bullying can have physical effects like headache, stomach problems and issues which are created due to stress like various conditions of skin and also ulcers of stomach. Victims may have eating disorders and various weight-related issues and sleeping disorders like insomnia. Apart from the physical effects, it may also have psychological effects like anger, frustration, sadness, behavioural problem like losing interest and in adverse condition may result in suicide intention. Cyber bullying has the ability to take the whole world into its grip by spreading false information regarding politics, diseases, laws and many more. Targeting a person based on appearance, different ideology, colour, chauvinism, sexual preference is a familiar occurrence. It often passes prejudiced and hatred towards targeted person or group. Cyber bullying has created a lot of hue and cry in the world and has created a compelling situation that has to be dealt with by recognizing such activities instantaneously and to develop stringent laws protecting people against search felony [1–8].

From Fig. 1, it can be observed that most cases of cyber bullying have been proclaimed by India. According to research conducted by Symantec, it is estimated that out of every 10 individuals nearly 8 individuals have been victims of Cyber bullying of some form in the nation accompanied by Brazil and the United States. Out of all the social media platforms, Twitter is one such means by which bullying occurs. It is an indispensable way for considerable socializing and to affix with like-minded people to prorate opinion. It also endows a platform to monitor brands and its eminence while keeping up with the voguish news around the globe and has seen a gradual rise in cyber bully associated cases. Hence, in this system, we have used Twitter dataset which focuses on bullying related to text.

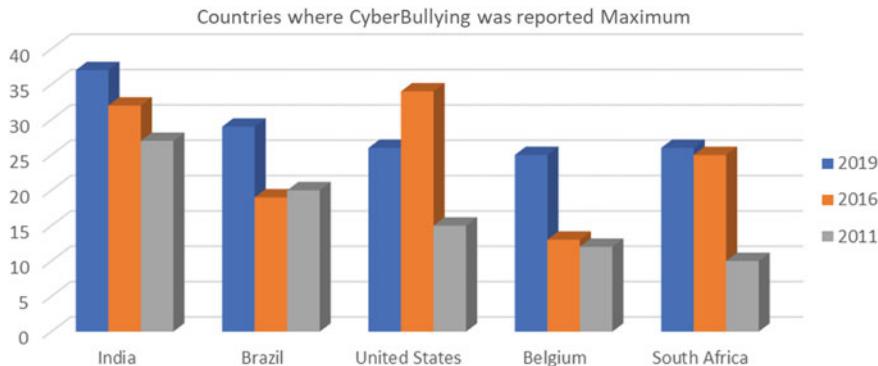


Fig. 1 Graphical representation of countries where Cyber bullying was reported maximum

2 Proposed Approach

The proposed approach using Convolutional Neural Network (CNN) algorithm for Cyber bullying Detection aims to improve the efficiency of the system that used long feature extraction and selection methods resulting in tedious and time-consuming process. CNN algorithm targets to improve the classification of tweets categorizing it into bullying or non-bullying divisions by using a word embedding an approach where words with analogous meaning have similar representation in the form of vectors which efficaciously saves the process of feature determination and extraction. As in the process of Feature determination and extraction method, the features can be entered manually or automatically that are considered to be relevant to the matter dealt in a text, in this case, bullying. With the exponential number of tweets abounding features are added which only makes things more conglomerate. It is then passed to the classifier and thus the use of word embedding saves the effort. For training of the system labelled data is used. After the training period, the System will detect cyber bully related tweets that have matching keywords from the trained database. In order to determine the cyber bully, we are focusing on the keywords posted by the users.

3 Methods

Figure 2 represents an overview of the proposed system. Initially, it starts with fetching of tweets from Twitter database or we can enter the tweets as an input using the keyboard too. Tweets are fetched and then pre-processed using a python code such that stop words, noisy and irrelevant data are removed and the processed words are then tokenized. A matrix-vector is formed to capture semantics and model words using word embedding (Fig. 3).

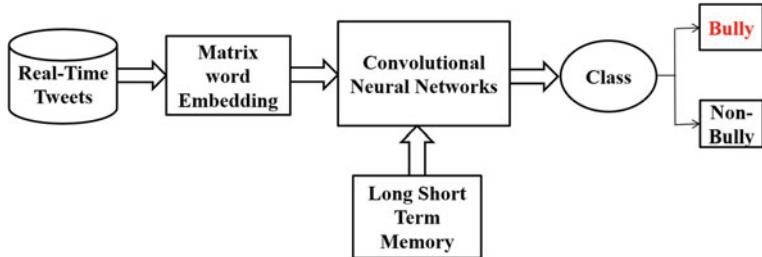


Fig. 2 Overview of the proposed approach

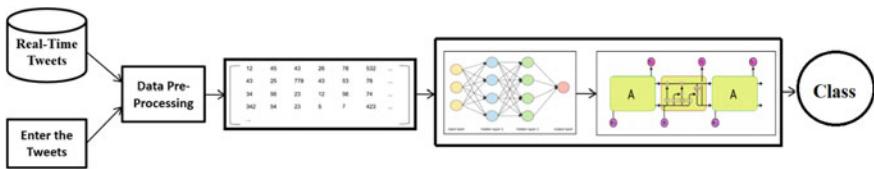


Fig. 3 Overall architecture

3.1 Data Pre-processing

Data pre-processing is a technique by which raw data is converted into useful information. In the data cleaning stage, we remove punctuation, emojis and removing of a numerical along with the lowercase conversion takes place. It also eliminates noisy and irrelevant data in the next stage tokenization take place in which sentence of the tweets are divided into certain characters, for example, the sentence “please give me three apples” would be divided into word “please, give, me, three, apples”. The stop words are removed after the tokenization the stop words are words like “be, can, is, the” which can be ignored there is no standard list for stop words. In the last step of data pre-processing, we reduced the inflected words, and hence words like happily, happiness and happy are reduced to happy as the word happy is lemma of all these words (Figs. 4 and 5).

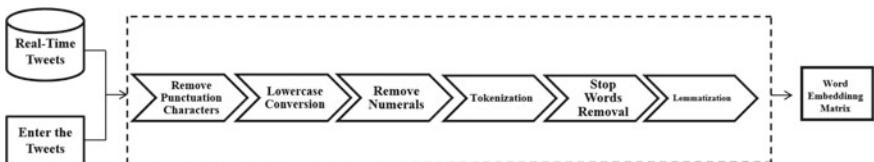


Fig. 4 Steps involved in data pre-processing

Fig. 5 Representation of data pre-processing

3.2 Convolutional Neural Network

CNN algorithm compartmentalizes tweets in more intelligent methodology than any other classification algorithm. CNN also jettisons the work required in the traditional cyber bullying detection system by adapting major principles instead of long machine learning classification techniques. This method of implementation also eliminates the few layers that were used in other traditional classification algorithms and acts as a remarkable aspect. Convolutional neural networks are inclusive of neurons called Covnets. Covnets in CNN share their parameters and is designed to meet the needs for classification. CNN algorithm is a multilayer perceptron and useful in natural language processing. The fact that they are useful for these fast-roaring growth areas is one of the main reasons they are so important in Deep Learning and Artificial Intelligence technology. Each of these layers contains neurons that are connected to neurons in the previous layer.

3.2.1 Input Layers

Number of neurons in input layer is equal to the total number of features in our data. It accepts input in different forms.

3.2.2 Hidden Layers

Input from the previous layer (input layer) is forwarded to further neurons that are present in between input and output layer and are called Hidden Layers. In these layers, number of neurons is greater than number of features. It performs calculations on the input.

3.2.3 Output Layers

Output of hidden layer is fed into logistic function which converts output of each class into probability score of each class. It delivers the outcome of the calculations and extractions. Each neuron has its own weight. Instead of neurons being connected to every neuron in pre-layer, they are instead only connected to neurons close to it and all have the same weight. This simplification in networks means the new network upholds the spatial aspect of the data set.

3.3 Long Short-Term Memory

A long short-term memory is networks which is a type of recurrent neural network and are capable of retaining the information for a long period of time. A recurrent neural network which is used for creating memory in neural network has a hidden state which helps to store information about the past. It also allows us to update the hidden state and solves the problem of sequence prediction partially. But fails because it is possible that the relevant information is not present at the site where it is needed because of many irrelevant data and it cannot predict the words stored in long memory but instead is used for recent information, for example, an apple has colour _____. The answer anticipated would be red. As in this context, the RNN has relevant information required to make prediction. Furthermore, if there is a statement I bought apples from the market. They are very delicious and its colour is _____ the network needs context apple from the previous statement and it is possible that the gap between two statements is more and the network will not be able to associate the information. And hence LSTM is used which has a property of selectively remembering patterns with the help of different memory blocks called cells. The ability to remember is not learnt but a default nature of the LSTM. The memory blocks are managed using the gates.

3.3.1 Forget Gate

The inputs which are given to the network are multiplied with the matrix and the addition of bias is done. It is then passed through activation function which gives output 0 or 1. If the output is zero then the pattern is not remembered similarly if the output is 1 then the pattern is remembered.

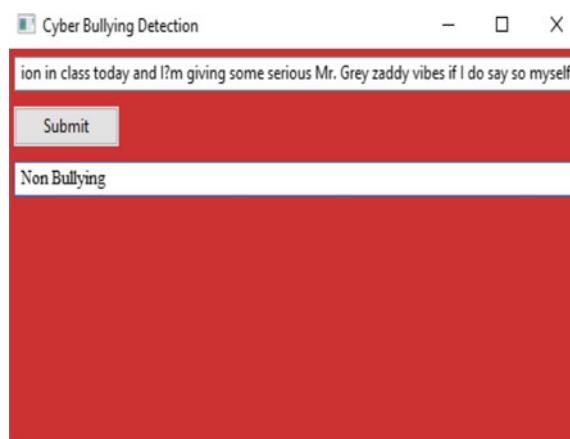
3.3.2 Input Gate

All the useful information in the network is passed using input gate.

Fig. 6 Output: Bullying detected



Fig. 7 Output: Non bullying



3.3.3 Output Gate

It determines the output to be generated.

4 Results

Figures 6 and 7 symbolize the end product of the proposed methodology where the system effectively distinguishes the text into its relevant category.

5 Future Work

Several possible optimizations for future works are as follows:

1. As sending image and video is becoming popular among adolescents. Hence, image and video processing would be another important area for cyber bullying detection.
2. Cyber bullying can also be further improved to take a variety of actions depending on the perceived seriousness of the post.
3. Detecting cyber bullying in streaming data.
4. Evaluating Annotation judgement.

6 Conclusion

This paper proposes a system to detect cyber bullying in real-time by using Twitter API. As social media is an emerging platform to connect worldwide and easy source to attack anyone in many forms of danger like cyber bullying. Automatic detection of cyber bully would enhance moderation and allow to respond quickly when necessary including different types of cyber bully covering posts from bullies and victims. We, therefore, intend to apply deep learning techniques to improve classifier performance.

References

1. R. Zhao, K. Mao, Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoders. *IEEE Trans. Affect. Comput.* (2015)
2. V. Nandakumar, B.C. Kovoor, MU Sreeja, Cyber-bullying revelation in twitter data using Naive-Bayes classifier algorithm. *Int. J. Adv. Res. Comput. Sci.* **9** (2018)
3. S. Bhoir, T. Ghorpade, V. Mane, Comparative analysis of different word embedding models (IEEE, 2017)
4. E. Raisis, B. Huang, Cyberbullying detection with weakly supervised machine learning, in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (2017)
5. E. Raisi, B. Huang, Weakly supervised cyberbullying detection with participant vocabulary consistency. *Soc. Netw. Anal. Min.* (2018)
6. P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Wei, B. Xu, Attention-based bi-directional long short-term memory network for relation classification, in *Proceedings of the 54th Annual Meeting of the Association For Computational Linguistics*, 12 Aug 2016, pp. 207–212
7. A. Conneau, H. Schwenk, Y.L. Cun, Very deep CNN for text classification, vol. 1. Association for Computational Linguistics, pp. 1107–1116 (2017)
8. <https://www.comparitech.com/internet-providers/cyberbullying-statistics/>

Soil pH Prediction Using Machine Learning Classifiers and Color Spaces



Tejas Wani, Neha Dhas, Sanskruti Sasane, Kalpesh Nikam, and Deepa Abin

Abstract The Indian economy is primarily dependent on agriculture. Successful production of crops is a need to ensure whether a particular crop will yield in a specific soil. Ph value, alkalinity, basicity directly affect the growth of the plant. Soil preparation is the most crucial process before plantation. All these factors of the soil can be determined by using the color image processing techniques. All farmers are interested in knowing how much yield can be expected. In the past days, yield prediction was performed by considering farmer's experience for a particular field. The crop yield prediction is a major issue that is unsolved based on available data with some limitations. If the crop is not yielding correctly, that means it must have some drawbacks. The proposed pH value prediction of 40 soil images is carried out. Using these color models, pH factor of each soil image is calculated. Different classifiers are applied to each color space model, and accuracy and RMSE values are obtained. So, the system primarily focuses on predicting the appropriate pH of a soil so that the crop will be predicted by using the pH values of the soil. The soil images are processed, and the pH values are gained.

Keywords pH · RGB · LUV · YcbCr · MLP · Naïve Bayes · Random forest · Random tree · J48 · SMO

T. Wani (✉) · N. Dhas · S. Sasane · K. Nikam · D. Abin

Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India
e-mail: tejaswani125@gmail.com

N. Dhas

e-mail: ndhas27@gmail.com

S. Sasane

e-mail: sanskruti192@gmail.com

K. Nikam

e-mail: kalpeshnikam1080@gmail.com

D. Abin

e-mail: deepaabbin@gmail.com

1 Introduction

India is one amongst the oldest countries that is still practicing agriculture. Farmers have a high range of variety of crops, and there is a drastic change in agricultural trends. Several factors have a great impact on the health of agriculture in India. Soil is recognized as the most valuable natural resource. The pH in the soil is important concerning part of the soil health [1]. The pH is defined as a scale to measure the parameters such as the acidity and basicity of the soil. Soil acidity and basicity directly affect the growth of the plants. Moreover, soil texture is the second important property that is related to the soil's electrical conductivity, water and nutrient content [2]. Soil texture indicates the sand, silt and clay fractions within the soil. Soil preparation is the most essential process before plantation. Excessive or lacking of the fertilizer use may have a significant effect on the crop yield. Soil testing is the way to know the amount of fertilizer that is necessary for the crop to grow. Soil testing will also help to overcome the soil problems. Most of the farmers do not rely on soil testing because the existing method consumes more time and more money. Very few farmers rely on soil testing done by the Government labs, which are not available near them. Soil which is sweet or sour gets difficulty in taking constituents like phosphorous, potassium and nitrogen [3]. Massive variety of possible soil colors is white, black, red, yellow and brown which are affected by the biological content and the amount of water. The existence of iron oxides is given by yellow and red soil. Organic matter existence is given by dark black or brown soil. The soil that has more water content appears to be darker than the soil having less water. Red and the brown colors of the soil are due to oxidation. Thus, the amount of the organic matter in the soil and the presence of water in it are the influenced factors of pH [4]. The camera receives the light in red, green, blue bands. These colors are the fundamental colors that are arranged in the band, which specifies the wavelength of electromagnetic radiation in a band of spectrum.

2 Literature Survey

The paper [1] states digital image processing techniques for determining the pH level of the soil. The author says that the degree of acidity or basicity is described by the pH of soil. The RGB digital values give a spectral signature of soil for wavelengths. The overall objective of the model is to determine the pH level of the soil. This model helps in the detection of color models values concerning soil color. In this paper [5], pH off soil is analysed using digital photographs of soil samples. The digital soil morph metrics are applications of tools techniques used for measuring and quantifying soil profile attributes. The soil attributes are widely used in classifying the soil and indicate many soil functions. The system in this work represents the concept of eigenvectors. Eigenvalues are compared to both the images, here matching is done of soil images, and the pH value is extracted.

The paper [2] is based on the fact that soil pH property is used to scale the acidic and basic contents in soil which affect plant growth. The digital image processing techniques such as histograms for density estimation, Bayer filter are used in the paper. For deep brown soil, pH ranges from 7.30 to 7.50. Similarly, for light yellowish soil, pH ranges from 6.80 to 7.04. For greenish soil, it ranges from 5.58 to 6.58. This paper [6] is based on the fact that the soil's pH is the most important factor in maximizing crop production. The paper introduces a technique for determining the crops that are suitable for the soil. The paper presents a system that helps in determining the pH level of soil by using artificial neural network (ANN). The pH value was found with the help of the color of soil and using the neural network. Four statistical indicators were used for performance evaluation. Mean absolute error (MAE) was 0.8, mean absolute percentage error (MAPE) was 0.2, root mean square error (RMSE) was calculated as 0.18, and determination coefficient (R^2) was 0.8.

Kamble Pravin, in the paper [7], explains the importance of soil testing. The amount of fertilizer and pH value are examined by the given paper. Eighty soil samples were collected, and their pH was tested and determined by using digital image processing techniques. The system works by providing the image file of a soil sample as input, and then, the RGB value of each image of soil sample was calculated by using digital image processing, and by using some constrained, they give soil pH and the constituents of soil as output. The pH was calculated by using the RGB function present in the MATLAB. The software gives the result of 60–70% in accuracy. In this paper [8], classifying the soil texture potential using RGB histograms was investigated. Linear patterns between slit content and histogram variables were shown by scatter plots. Root mean square error of calibration and prediction, the squared correlation coefficient was 2.2%, 6.3% and 0.96, respectively. Only 48% of samples were tested correctly. Future scope defined in the paper is to use different averaging techniques, and variable transformations could be applied. Another approach like Fourier transformation and AI techniques might be useful.

The pH level of the soil is determined using digital image processing techniques in paper [3]. The samples of the soil were segregated based on its color and the texture. The features are extracted using various techniques such as binarization, thresholding, normalizing and resizing. Images of the sample of soil are captured using digital cameras. Finally, the values taken by the lab and values acquired by the image processing are compared with the help of MATLAB.

3 Methodology

3.1 Dataset Discussion

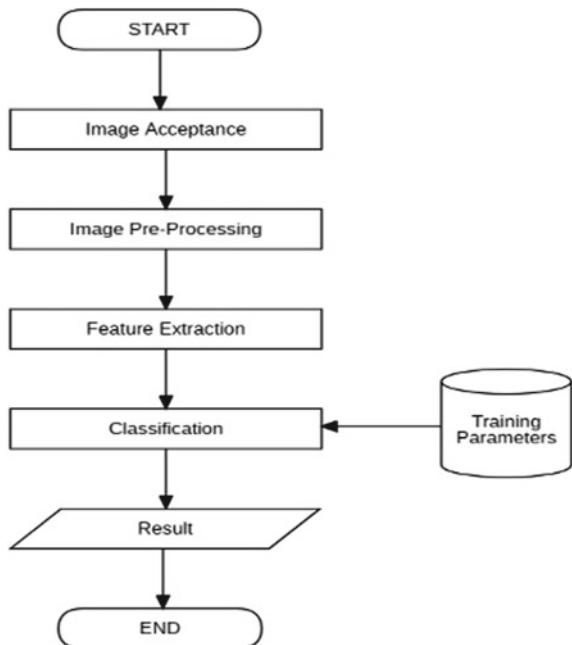
The dataset comprises the different soil images. The dataset is gathered from Kaggle and contains total of 40 soil images. The dataset has its Lab tested pH value as the name of the image file, for example, name of the file is ‘5.01.jpg’, here 5.01 is the pH

value of the soil. As the laboratory tested pH of each soil image is known, it will be helpful for the validation of pH calculated through various digital image processing techniques.

3.2 Proposed System

The process theme consists of acquiring image through scanner, camera or transportable. The image processing includes improving images and applying filters to the image to get rid of the noise, etc. The RGB, YCbCr and LUV values are extracted from the images in the feature extraction stage. The classification models like multi-layer perceptron, support vector machine, Naïve Bayes and J48 classifiers are used for the classification. The training parameters from the dataset are given to the classification model. The result is generated from the classification model with the training parameters provided (Fig. 1).

Fig. 1 Proposed system flowchart



3.3 Color Models and Classification Techniques

The survey of various methods, color models and classification technologies for prediction of pH value of soil image is discussed below:

3.3.1 Color Models

(a) ***RGB Color Space Model:***

RGB color space is used to find the pH factor based on RGB value of image. RGB color values are extracted from the soil image. This method is used for feature extraction of image, once the image is captured, and preprocessing is done on it. Soil pH factor can be calculated as

$$\text{soil pH Factor} = \frac{\text{avg}(R)/\text{avg}(G)}{\text{avg}(B)} \quad (1)$$

For new soil sample, to calculate pH value, we need to first capture the image of soil, then by using the soil pH factor formula, we need to calculate the new factor of new image. The new image can be taken as matrix of pixels related to combination of red, blue and green values.

The single soil pH factor value of each image can be calculated by taking average of each sector. We need to add +0.01 and subtract -0.01 as we have only few samples pH and their calculated pH factors (index) to get the approximation in the results.

Then, the new value is compared with the values already stored in the database which is in particular range returns the soil pH factor, and according to it, we return the soil pH value of new soil sample.

(b) ***YCbCr Color Space Model:***

The YCbCr space is used for the component digital video. YCbCr color space is a offset version of the YUV color space.

The following are simple equations to convert between RGB in the given range 0–255 and YCbCr [9].

$$Y = (0.257 * R_{\text{avg}}) + (0.504 * G_{\text{avg}}) + (0.098 * B_{\text{avg}}) + 16 \quad (2)$$

$$Cb = -(0.148 * R_{\text{avg}}) - (0.291 * G_{\text{avg}}) + (0.439 * B_{\text{avg}}) + 128 \quad (3)$$

$$Cr = (0.439 * R_{\text{avg}}) - (0.368 * G_{\text{avg}}) - (0.071 * B_{\text{avg}}) + 128 \quad (4)$$

$$\text{pH}_{\text{YCbCr}} = \frac{\text{double}(Cb)/\text{double}(Cr)}{\text{double}(Y)} \quad (5)$$

(c) **LUV Color Space Model:**

Dr. Kekre's LUV color space is an attempt to define an encoding with basic uniformity in the perceptibility of the color differences. Equations (6)–(8) give formula to calculate LUV component of soil image [10].

$$L = R + G + B \quad (6)$$

$$U = -(2 * R) + G + B \quad (7)$$

$$V = -G + B \quad (8)$$

$$\text{pH}_{\text{LUV}} = \frac{\text{double}(L)/\text{double}(U)}{\text{double}(V)} \quad (9)$$

3.3.2 Classification Techniques

(a) **Multilayer Perceptron (MLP):**

The multilayer perceptron is also defined as feedforward artificial neural network (FFNN). MLP is for the feedforward ANN and even for the networks which are composed of multiple layer of perceptrons. An multilayer perceptron consists of three layers nodes—input layer (one), hidden layer (one or more), output layer (one)

MLP makes use of the supervised learning technique also known as back propagation for training. The hidden layers make it possible for the network to exhibit the nonlinear behavior. When the hidden layers are increased, the complexity of the model increases. The perceptron is also known as the linear classifier.

Input is

$$y = w * x + b \quad (10)$$

where

x = Feature vector,

w = weights,

b = Bias (Fig. 2).

(b) **Support Vector Machine (SVM):**

SVM is supervised learning method that analyzes data for regression and classification. SVM is used in different purpose like text categorization, handwriting recognition, etc. SVM is also used to split the data in best way.

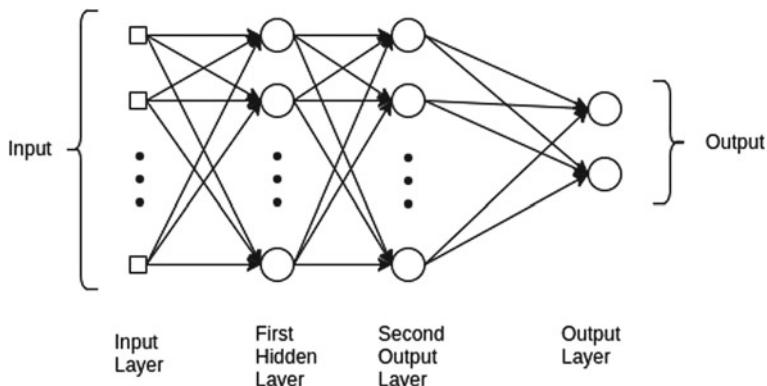


Fig. 2 Multilayer perceptron layers

SVM divides the mix data in separate cluster or class. If data mix with each other, then kernel method is used to separate out data using library function.

(c) **Naïve Bayes:**

Various objects are differentiated using Naïve Bayes based on particular feature. Naive Bayes works on the principle of probability, i.e., it is a probabilistic machine learning model used for classification work. It is simple and outperforms more complex classification methods. The core part of this classifier is based on Bayes theorem stated below:

$$P(x|y) = [p(y|x) * p(x)]/p(y) \quad (11)$$

By using this theorem, we can find that the probability of x happening, given y has occurred. The assumption made here is that the predictors/features are independent. Applicative use of Naïve Bayes can mostly be seen in sentiment analysis, spam filtering and recommendation systems, etc.

(d) **J48 Classifier:**

J48 is used to generate decision trees. In the WEKA, J48 is an open-source Java implementation of the C4.5 algorithm. C4.5 is an extension of earlier ID3 (Iterative Dichotomiser 3) algorithm where ID3 is an algorithm used to build a decision tree from a dataset. The limitations of ID3 include suffering from the problems of overfitting. J48 focuses to include accounting for missing values, decision trees pruning, derivation of rules, continuous attribute value ranges, etc.

4 Results and Discussions

The proposed pH value prediction of 40 soil image is carried out. The proposed system firstly accepts the images of soil sample. These images are preprocessed, and the features are extracted for three color spaces namely RGB, YCbCr and LUV color space model. Using these color models, pH factor of each soil image is calculated. The training parameters from the dataset are given to different classification models. The results are generated in the form of accuracy and RMSE values. The experimentations are stated in Table 1, showing the accuracy and error values for each classifier in every color model.

Table 1 shows the accuracy and RMSE values of different classifiers with respect to each color space model.

It shows that for each color space model, MLP shows the best accuracy as well as RMSE value. It is observed that for RGB color space model, highest accuracy obtained is 95% with RMSE value of 0.1271. For YCbCr model, the accuracy attained is 92.5% with RMSE value as 0.1515, and for LUV, accuracy is 95% with RMSE as 0.1315.

Figure 3 shows the graphical representation of comparative analysis of classifiers with each color model. It is observed that as the hidden layers of MLP are increased, the accuracy tends to increase at certain layer, after which the accuracy remains the same. Graphical representation in Fig. 3 clearly shows that SMO classifier gives poor accuracy with respect to other classifiers.

After obtaining results from Table 1, ensembling classifiers are formed, and accuracy and RMSE values are obtained for this ensembling classifiers.

Table 2 shows the accuracy and RMSE values of various ensembling models with respect to each color space. It is observed that the accuracy of 95% is obtained by using Naïve Bayes and MLP ensembling classifier for both RGB and YCbCr color space model, though the RMSE value is less for RGB as compared to YCbCr. For LUV, the accuracy is obtained as 95% for random forest and MLP as ensembling classifiers.

Figure 4 shows the comparative analysis of accuracy of ensembling classifiers for pH value prediction.

5 Conclusion

Agriculture is the backbone for a developing economy like India, and there is an enormous need to maintain the agricultural sustainability. One of the factors that are important for growth of particular crop is pH value; it decides whether the soil is acidic or alkaline in nature. This factor is very helpful to determine crops suitable for a particular environment as well as type of soil. The system proposes to classify the soil images based on pH values of soil. The experiment is carried on 40 soil images where RGB, YCbCr and LUV color model features are extracted from soil image

Table 1 Accuracy and RMSE values of different classifiers with respect to each color model

| Colour Models | Classifiers | | | |
|---------------|-------------|---------------------|----------|--------|
| | Sr. No. | Classifier | Accuracy | RMSE |
| RGB | 1 | MLP(Hidden Layer 1) | 77.5 | 0.3209 |
| | 2 | MLP(Hidden Layer 2) | 92.5 | 0.1804 |
| | 3 | MLP(Hidden Layer 3) | 92.5 | 0.1765 |
| | 4 | MLP(Hidden Layer 4) | 95 | 0.1271 |
| | 5 | MLP(Hidden Layer 5) | 95 | 0.1557 |
| | 6 | MLP(Hidden Layer a) | 95 | 0.1271 |
| | 7 | SMO | 72.5 | 0.3555 |
| | 8 | Random Forest | 90 | 0.2216 |
| | 9 | Random Tree | 87.5 | 0.25 |
| | 10 | Naïve Bayes | 92.5 | 0.1557 |
| | 11 | J48 | 92.5 | 0.1936 |
| YCbCr | Sr. No. | Classifier | Accuracy | RMSE |
| | 1 | MLP(Hidden Layer 1) | 82.5 | 0.3016 |
| | 2 | MLP(Hidden Layer 2) | 92.5 | 0.182 |
| | 3 | MLP(Hidden Layer 3) | 92.5 | 0.1556 |
| | 4 | MLP(Hidden Layer 4) | 92.5 | 0.1516 |
| | 5 | MLP(Hidden Layer 5) | 92.5 | 0.1571 |
| | 6 | MLP(Hidden Layer a) | 92.5 | 0.1515 |
| | 7 | SMO | 80 | 0.3466 |
| | 8 | Random Forest | 90.5 | 0.1763 |
| | 9 | Random Tree | 87.5 | 0.25 |
| | 10 | Naïve Bayes | 92.5 | 0.1795 |
| | 11 | J48 | 92.5 | 0.1936 |
| LUV | Sr. No. | Classifier | Accuracy | RMSE |
| | 1 | MLP(Hidden Layer 1) | 80 | 0.3127 |
| | 2 | MLP(Hidden Layer 2) | 92.5 | 0.1765 |
| | 3 | MLP(Hidden Layer 3) | 92.5 | 0.1768 |
| | 4 | MLP(Hidden Layer 4) | 92.5 | 0.1378 |
| | 5 | MLP(Hidden Layer 5) | 95 | 0.1315 |
| | 6 | MLP(Hidden Layer a) | 92.5 | 0.1378 |
| | 7 | SMO | 67.5 | 0.3613 |
| | 8 | Random Forest | 92.5 | 0.1637 |
| | 9 | Random Tree | 85 | 0.2739 |
| | 10 | Naïve Bayes | 90 | 0.219 |
| | 11 | J48 | 92.5 | 0.1936 |

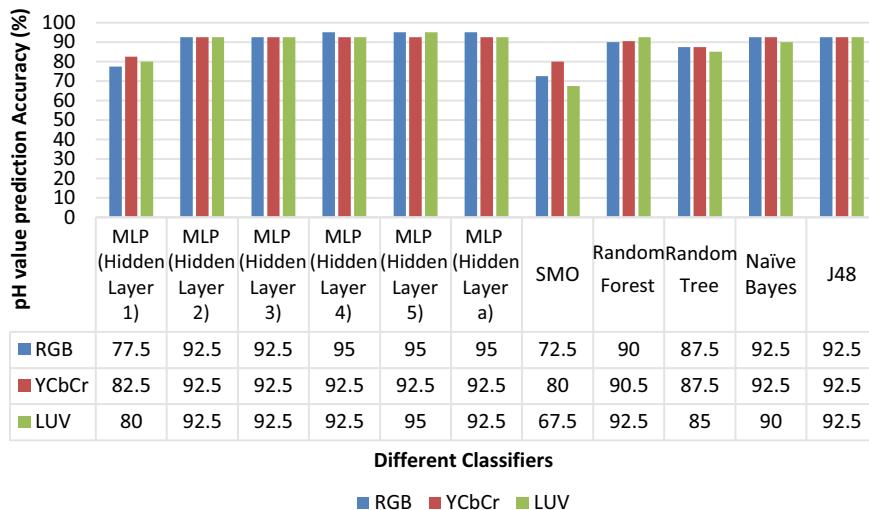


Fig. 3 Comparative analysis of accuracy of different classifiers in proposed pH value prediction

Table 2 Accuracy and RMSE values of ensembling classifiers with respect to each color space

| Colour Models | Classifiers | | Accuracy | RMSE |
|---------------|-------------|-------------------------------|-----------|---------------|
| | Sr. No. | Classifier | | |
| RGB | 1 | J48 and Random Forest | 92.5 | 0.1391 |
| | 2 | MLP and J48 | 92.5 | 0.1294 |
| | 3 | Random Forest and MLP | 95 | 0.1441 |
| | 4 | Naïve Bayes and MLP | 95 | 0.1325 |
| | 5 | Naïve Bayes and J48 | 90 | 0.1249 |
| | 6 | Random Forest and Naïve Bayes | 95 | 0.1477 |
| YCbCr | Sr. No. | Classifier | Accuracy | RMSE |
| | 1 | J48 and Random Forest | 92.5 | 0.1386 |
| | 2 | MLP and J48 | 92.5 | 0.1282 |
| | 3 | Random Forest and MLP | 92.5 | 0.1565 |
| | 4 | Naïve Bayes and MLP | 95 | 0.1454 |
| | 5 | Naïve Bayes and J48 | 92.5 | 0.1325 |
| LUV | 1 | J48 and Random Forest | 92.5 | 0.1363 |
| | 2 | MLP and J48 | 92.5 | 0.123 |
| | 3 | Random Forest and MLP | 95 | 0.1382 |
| | 4 | Naïve Bayes and MLP | 90 | 0.1594 |
| | 5 | Naïve Bayes and J48 | 92.5 | 0.1485 |
| | 6 | Random Forest and Naïve Bayes | 90 | 0.1765 |

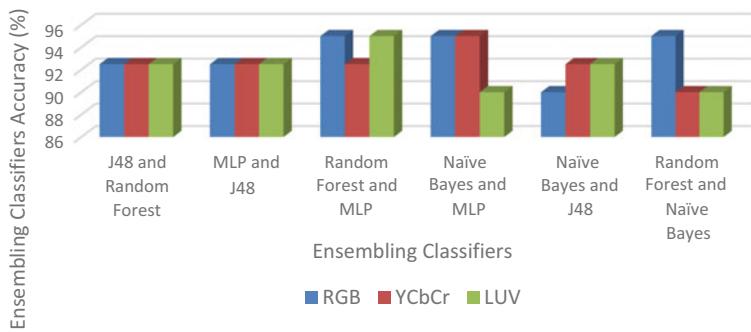


Fig. 4 Comparative analysis of accuracy of ensembling classifiers for pH value prediction

and pH factors are calculated. Various classifiers are applied, and it is observed that for RGB color space, the accuracy obtained is 95% with MLP classifier. For YCbCr color space model, accuracy obtained is 92.5%, and for LUV, accuracy is 95%. This system can be an aid for helping today's farmers.

References

1. V. Kumar, B.K. Vimal, R. Kumar, M. Kumar, Determination of soil pH by using digital image processing technique. *J. Appl. Nat. Sci.* **6**(1), 14–18 (2014)
2. R. Sudha, S. Aarti, S. Anitha, K. Nanthini, Determination of soil Ph and nutrient using image processing. *IJCTT* (2017)
3. S. Kshirsagar, P. Lendave, A. Vibhute, Soil nutrients analysis using color image processing. *IRJET* **5** (2018)
4. J.C. Puno, E. Sybingco, E. Dadios, I. Valenzuela, J. Cuello, Determination of soil nutrients and pH level using image processing and artificial neural network. *IEEE* (2017)
5. M.S. Gurubasava, S.D. Mahantesh, Analysis of agricultural soil pH using digital image processing. *Int. J. Res. Advent Technol.* **6**(8), 1812–1816 (2018)
6. M.M. Aziz, D.R. Ahmeed, B.F. Ibrahim, Determine the Ph of soil by using neural network based on soil's colour. *IJARCSSE* **6** (2016)
7. U. Kamble, P. Shingne, R. Kankrayane, S. Somkuwar, S. Kamble, Testing of agriculture soil by digital image processing. *IJSRD* **5** (2017)
8. S.-O. Chung, K.-H. Cho, J.-W. Cho, K.-Y. Jung, Texture classification algorithm using RGB characteristics of soil images. *Kyushu University Institutional Repository* **57** (2012)
9. E. Prathibha, A. Manjunath, R. Likitha, RGB to YCbCr color conversion using VHDL approach. *Int. J. Eng. Res. Dev.* (2012)
10. H.B. Kekre, S.D. Thepade, Image blending in vista creation using Kekre's LUV color space, in *Proceedings of SPIT-IEEE Colloquium and International Conference*, Mumbai, India
11. M.R. Dharwad, T.A. Badebade, M.M. Jain, A.R. Maigur, Estimation of moisture content in soil using image processing. *IJIRD* **3** (2014)
12. S. Krishna Prasad, B. Siva Sreedharan, S. Jaishanth, *Crop Monitoring And Recommendation System Using Machine Learning Techniques*. Madras Institute of Technology, Chennai (2017)

A Food Recommendation System Based on BMI, BMR, k -NN Algorithm, and a BPNN



Anilkumar Kothalil Gopalakrishnan

Abstract In this research, a novel food recommendation system is presented for recommending a proper calorie daily food for an overweighed person to gain a healthy body status by using his or her Body Mass Index (BMI), Basal Metabolic Rate (BMR), k -Nearest Neighbors (k -NN) algorithm, and a back-propagation neural network (BPNN). The system estimates the overweight status of a person by using the BMI value. By using the BMR value, the system calculates the Daily Needed Food calories (DNC) of a person. The k -NN algorithm selects a proper calorie daily food set from the food dataset by using the saturated value of the DNC as its test object. The system predicts the days required for a person to gain a healthy BMI status with the recommended food by using overweight and saturated DNC values. Finally, the system evaluates its user's satisfaction level based on the BPNN. The presented food recommendation system could be an effective way of propagating healthy weight awareness among common people.

Keywords BMI · BMR · Daily needed food calorie · Saturated daily food calorie · Neural network · k -NN algorithm · Overweight value · User satisfaction

1 Introduction

Individuals living with healthy life are assets of our society. Such personalities are always being recognized as top class social elements, and are privileged to be leaders of any socially related enterprises. In fact, in many aspects of human life, all of the unhealthy weight problems are mostly connected to their anomalous food habits. The tremendous growth of the fast food industry has contributed to anomalous eating habits among human beings that could surely be lead to the generation of unhealthy, overweighed, and socially sick citizens.

A. K. Gopalakrishnan (✉)

Department of Computer Science, Vincent Mary School of Science and Technology, Assumption University of Thailand, Bangkok, Thailand
e-mail: anil@scitech.au.edu

Therefore, this research aims to demonstrate a novel food recommendation system for overweighed persons of our society. The system uses the Body Mass Index (BMI) and Basal Metabolic Rate (BMR) values of individuals in order to estimate their overweight value and saturated value of the Daily Needed Calories (DNC). Actually this paper is just an extended research version of the previous research work which focused only on overweight and its management issues [1]. This paper focuses on the development of the novel ways to recommend a proper calorie daily food to an overweighed person with an abnormal BMI value. This means that this research meant to focus only to care the individuals with a high BMI value (which is above the “normal” BMI status). By using the BMI and the saturated DNC values of individuals, the system would be able to predict the days required for a person to achieve a healthy BMI status (or a “normal” healthy weight) with the recommended food by the system. Furthermore, the food selection process of the system is carried out by the application of the k -NN algorithm [2–4], which is one of the most popular instances based learning algorithms. The system also has a facility to evaluate its user’s satisfaction level of the system with its recommended food by using the BPNN [5–8]. The user evaluation of the system is based on the four rating attributes of its users. The initial dataset for the BPNN is developed based on these four user rate values.

The following sections of this paper would describe the literature review, the importance of the BMI and BMR in the system, application of the k -NN algorithm, application of the BPNN, the complete structure of the system, and the conclusion.

2 Review of Related Literature

The Body Mass Index (BMI) is an estimation of an individual’s body fat based on their height and weight values [9]. This measure can be used as a scale to predict whether an individual is at risk of coronary disease, cardiac arrest issue, high arterial cholesterol issue, pancreatic disorder, kidney disorder, and certain myeloma based on the large values of BMI. This research uses the BMI measure as a major key point for estimating the unhealthy weight status of individuals. From this, the system calculates their overweight values for predicting the number of days for them to attain a healthy BMI status with the recommended food.

From the restaurant food service system [10], it can be noticed that the k -NN algorithm was employed for the classification of food items as per the customer choices. But the system did not care about the food calories or customer’s overweight issues. Similarly, a combined structure of the k -NN algorithm with the Support Vector Machines (SVM) employed in a food image classification system is described in [11]. Somehow, it has shown that the application of the k -NN algorithm in an image recognition system. Even though research has successfully proved the usage of the k -NN in a food image recognition application, it did not show any attention toward the recognition of foods with healthy calorie values.

This food recommendation research is just an extended version of a previous research paper that has focused only on weight prediction related issues of overweight individuals based on their BMI and BMR, and the concept of food recommendation is not at all a part of it [1]. So far various types of BMR calculations have been developed; up to now there are 248 BMR equations available of researches [12, 13]. Therefore estimating a suitable BMR for individuals became crucial due to the complexity of selecting a proper BMR equation for health related computations. As per the 2020 edition of American Dietary Guidelines, women should consume a daily calorie of 2000 to maintain their weight, and 1500 calories daily to lose one pound of weight per week. Similarly, men should consume a daily calorie of 2500 to maintain his weight, and 2000 to lose one pound of weight per week [14].

3 Importance of BMI and BMR in the Food Recommendation System

The BMI status of an individual is the strength of this paper. The BMI indicates the overweight or lower weight status of a person by the application of his/her height, and weight values but it does not address how to change the body fat and muscle mass of the person [15]. The BMI value is just a scientific estimation of a person's body weight status, and it has nothing to do with food calorie, body fat, metabolic rate, etc. of the person. For this reason, this research has employed the BMR along with the BMI to estimate the number of days needed for an overweighed person to attain in a state of "normal" BMI. From this information, it can be concluded that the BMI is just an indirect indication of obesity and lower weight status of a person, and the BMI of a person can be estimated as [16]:

$$\text{BMI} = (\text{weight in Kilogram}) / (\text{height in meters})^2 \quad (1)$$

As per the above equation, various bodyweight statuses of individuals including their normal weight, overweight, underweight, and classes of obesities are estimated and are shown in Table 1. At this moment, the people with high body weight statuses including various classes of obesity cases are currently considered for this research.

The *overweight* value W of a person can be calculated from the height value (in meter) and the *normal weight* boundary values of the BMI (which are 18.5 and 24.9, see Table 1), called $\text{BMI}_{\text{normal}}$. The W (in kilogram) can be estimated as [1]:

$$W = \text{BMI}_{\text{normal}} * \text{height}^2 \quad (2)$$

The upper boundary value of W called W_{upper} , and it can be estimated from the upper most value of the $\text{BMI}_{\text{normal}}$ (24.9). The W_{upper} can be calculated as [1]:

$$W_{\text{upper}} = 24.9 * \text{height}^2 \quad (3)$$

Table 1 Body weight status by the BMI

| BMI | Body weight status |
|-----------------------|---------------------------|
| Less than 15 | Very severely underweight |
| Between 15.0 and 15.9 | Severely underweight |
| Between 16.0 and 18.4 | Underweight |
| Between 18.5 and 24.9 | Normal weight |
| Between 25.0 and 29.9 | Overweight |
| Between 30.0 and 34.9 | Class I obesity |
| Between 35.0 and 39.9 | Class II obesity |
| Above and equal to 40 | Class III obesity |

From (3), the overweight_{\min} of a person can be calculated as [1]:

$$\text{overweight}_{\min} = (\text{current weight} - W_{\text{upper}}) \quad (4)$$

Similarly, the overweight_{\max} value of a person can be calculated from the *lower* boundary value of W called W_{lower} , and the lower most boundary value of the $\text{BMI}_{\text{normal}}$ (18.5) as [1]:

$$W_{\text{lower}} = 18.5 * \text{height}^2 \quad (5)$$

From (5), the overweight_{\max} can be estimated as [1]:

$$\text{overweight}_{\max} = (\text{current weight} - W_{\text{lower}}) \quad (6)$$

Assume that a person with body weight 93 kg and height 1.68 m. As per (1), his/her BMI is 33.0 kg/m^2 . Similarly, from (3), the W_{upper} is 70.3 kg, and from (4), the overweight_{\min} is 22.7 kg. Similarly from (5), the W_{lower} is 52.24 kg, and from (6), the overweight_{\max} is 40.8 kg.

The BMR is the body calorie expenditure indication of a person based on his/her weight, height, age, and gender. It is one of the significant calorie attributes required for a person to lose or gain weight. From the values of BMR and BMI, it could generate a proper daily food calorie value for a person, and hence promoting healthy food habits among common people. This paper employs the Harris-Benedict version of BMR calculation for both men and women and is given as [17]:

$$\begin{aligned} \text{Men} &= 88.362 + (13.397 * \text{weight in kilogram}) \\ &\quad + (4.799 * \text{height in centimeter}) - (5.677 * \text{age in years}) \end{aligned} \quad (7)$$

$$\begin{aligned} \text{Women} &= 447.593 + (9.247 * \text{weight in kilogram}) \\ &\quad + (3.098 * \text{height in centimeter}) - (4.33 * \text{age in years}) \end{aligned} \quad (8)$$

Table 2 Estimated values of DNC by BMR

| | |
|----------------------------|-------------|
| Daily physical activity | DNC |
| Sedentary or little active | BMR * 1.2 |
| Lightly active | BMR * 1.375 |
| Moderately active | BMR * 1.55 |
| Very active | BMR * 1.725 |
| Extra active | BMR * 1.9 |

Table 2 shows the calculation of Daily Needed Calorie (DNC) of an individual from daily physical activity and the BMR value.

Based on Table 2, the equation for estimating a person's DNC can be given as [12]:

$$\text{DNC} = \text{BMR} * \text{Daily Physical Activity} \quad (9)$$

Assume that a sedentary life styled person living with very little physical activity has a BMR value of 1870.3 calories. From (9), the DNC of the person is 2244.36 ($1870.3 * 1.2$). It indicates that the person should need about 2244.36 calories to manage his/her daily body functions and life activities. In case, if the person would like to reduce weight, then he/she should loss 500 calories from his/her DNC value [15, 17, 18]. This means that the person should maintain a total of 1744.36 calories (2244.36–500) per day causes a lose weight of 1 lb (or 0.453592 kg) per week and is equivalent to losing a total of 3500 calories. That is, to reduce a body-weight of 1 lb per week requires a deficit of 3500 calories per week or deficit 500 calories from the DNC per day. The process of reducing 500 calories from a person's DNC value is called as DNC saturation (DNC_{sat}) and it can be given as:

$$\text{DNC}_{\text{sat}} = \text{DNC} - 500 \quad (10)$$

From (10), it can be noticed that the consumption of the food with DNC_{sat} will cause a weight reduction of 0.06485 kg per day and with $\text{overweight}_{\text{min}}$ value from (4), the Minimum Number of Days Needed (NDN_{min}) for an overweighed individual to reach the *upper normal level* of BMI (which is 24.9) can be given as:

$$\text{NDN}_{\text{min}} = \text{overweight}_{\text{min}} / 0.06485 \text{ kg} \quad (11)$$

Similarly, with $\text{overweight}_{\text{max}}$ from (6), the Maximum Number of Days Needed (NDN_{max}) for an overweighed individual to attain the *lower normal level* of BMI (which is 18.5) can be given as:

$$\text{NDN}_{\text{max}} = \text{overweight}_{\text{max}} / 0.06485 \text{ kg} \quad (12)$$

Table 3 shows the values of age in years, gender, BMI in kg/m^2 , BMR, DNC_{sat} , NDN_{min} , and NDN_{max} of eight overweighed persons.

Table 3 Values of age in years, gender, BMI, BMR, DNC_{sat}, NDN_{min}, and NDN_{max} eight persons

| Age | Gender | BMI | BMR | DNC _{sat} | NDN _{min} | NDN _{max} |
|-----|--------|------|--------|--------------------|--------------------|--------------------|
| 35 | M | 25.8 | 1788.0 | 2271.0 | 43 | 344 |
| 43 | F | 30.8 | 1599.0 | 1419.0 | 260 | 542 |
| 33 | M | 35.7 | 1852.0 | 2046.0 | 414 | 661 |
| 28 | F | 26.8 | 1513.0 | 1845.0 | 203 | 440 |
| 33 | M | 28.7 | 1993.0 | 1892.0 | 188 | 505 |
| 44 | M | 32.8 | 1945.0 | 2515.0 | 358 | 646 |
| 24 | M | 33.2 | 1822.0 | 2005.0 | 321 | 568 |
| 30 | F | 29.4 | 1667.0 | 1500.0 | 208 | 503 |

4 Application of the *k*-NN Algorithm

The *k*-NN is one of the simplest forms of machine learning algorithms, and it is a type of instance-based learning technique [2]. It could be the best choice for a simple data classification purpose when the available knowledge is not enough to manipulate the data [3]. It is one of the best options for the identification of features from an untrained data entry [2, 4]. The value *k* is a *problem dependent positive integer* which determines the number of nearest neighbors of the test object (test variable) of the problem based on its distance function [3]. The Euclidean distance function is one of the common distance metrics used with *k*-NN algorithm for a continuous variable application.

The food recommendation system employs the *k*-NN algorithm for its food selection application. The recommendation system has provided a food dataset with various food items including their calorie values. A popular food dataset called *MyPyramid-Food-Raw-Data* is selected for this research [19]. The food dataset has undergone a process of data normalization by combining the various individual food items along with their calorie values in order to reduce the size and complexity of the original dataset, prior to applying it to the system. The employed normalization process has resulted in the generation of various small units of food sets instead of many small calorie valued foods in the food dataset. This aforementioned normalization process on the food dataset has significantly reduced the number of rows in the food dataset, and it could fasten the process of finding the proper calorie food sets from the dataset by the *k*-NN algorithm as part of its food selection process. Again, it could always enable the system for selecting proper calorie food sets to its users by using the DNC_{sat} value as the *test object* for the *k*-NN algorithm. The description of the *k*-NN algorithm that is applied in this research is shown below:

The *k*-NN algorithm for the food system:

- i. Set the *k* value (*k* = 5).
- ii. Set the DNC_{sat} as the test object.
- iii. Estimate the Euclidean Distance between the test object and each calorie value of the food set.

Table 4 Calories of various food sets selected by the k -NN based on the DNC_{sat}

| DNC_{sat} | $k = 1$ | $k = 2$ | $k = 3$ | $k = 4$ | $k = 5$ |
|---------------------------|---------|---------|---------|---------|---------|
| 2271.0 | 2269.0 | 2279.0 | 2208.0 | 2318.0 | 2329.0 |
| 1419.0 | 1420.0 | 1410.0 | 1430.0 | 1452.0 | 1460.5 |
| 2046.0 | 2049.5 | 2060.0 | 2081.5 | 2086.0 | 2106.0 |
| 1845.0 | 1844.0 | 1848.0 | 1865.5 | 1805.0 | 1799.5 |
| 1892.0 | 1893.5 | 1886.5 | 1880.0 | 1910.0 | 1915.5 |
| 2515.0 | 2517.5 | 2523.5 | 2503.0 | 2534.5 | 2020.5 |
| 2005.0 | 2007.5 | 2012.0 | 2016.5 | 2018.0 | 2020.0 |
| 1500.0 | 1498.0 | 1509.0 | 1513.5 | 1526.5 | 1540.0 |

- iv. Sort the estimated distances together with calories and the food sets in their ascending order.
- v. Select the food sets (with their calories) based on the k .
- vi. Save the food sets together with their calories.
- vii. Go to the next search.

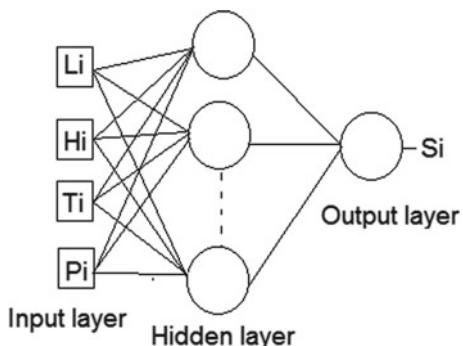
This food recommendation system does not support any individual with a *normal* or *underweight* BMI value. Table 4 shows the calories of various food sets resulted by the k -NN algorithm based on the DNC_{sat} from Table 1.

5 Application of the BPNN

Typically, a BPNN is a multilayer perceptron (MLP) with three or four layers including its input and output layers [6]. The MLP is formed by connecting every perception in each layer is connected to every other perceptron in the adjacent forward layer of the network [7]. The training of perceptrons in each layer of the network is by adjusting their numerical weights and biases by a gradient descent algorithm called back-propagation algorithm [8].

This food recommendation system has a user satisfaction level estimation section based on its recommended foods by using a BPNN. This feedback facility could be used to measure the healthy weight consciousness nature of its users. In the future, this information can be used to implement an auto correction facility to the system based on its user's suggestions. It means that the value of the user satisfaction level will be used to recommend various alternative food sets with sufficient user satisfaction levels if the currently recommended food set is indicated as unsatisfied. To estimate the user satisfaction level of the recommended food, the system should collect the values of the following four attributes from its users: (1) L_i , *level of interest toward the recommended food* by user i , (2) H_i , *health consciousness* of user i , (3) T_i , *trust towards gadget based health computation* by user i , and (4) P_i , *patience in controlling hunger* of user i . Based on the values of these attributes, the BPNN

Fig. 1 Structure of the 3-layer BPNN



predicts the satisfaction level S_i of its user i . From the values of the input and output attributes, the system generates the *initial* dataset pattern (also called *seen* dataset) for the BPNN. For this research, a 3-layer BPNN with topology 4 20 1 (four data inputs, twenty hidden neurons, and one output neuron) has selected. The BPNN has trained with a learning rate of 0.4. The structure of the selected BPNN is shown in Fig. 1.

In order to generate the initial training dataset for the BPNN, there are five numerical values with their linguistic terms are assigned to each of the four input attributes and their output attributes of the BPNN. The assigned numerical values with their linguistic meanings of the input and output attributes of the BPNN are given below:

- L_i , level of interest toward the recommended food of i has values: 0.1 (*very low*), 0.3 (*low*), 0.5 (*not low*), and 0.7 (*high*), and 0.9 (*very high*).
- H_i , health consciousness level of i has values: 0.1 (*very low*), 0.3 (*low*), 0.5 (*not low*), 0.7 (*high*), and 0.9 (*very high*).
- T_i , trust toward gadget based health computation of i has values: 0.1 (*very low*), 0.3 (*low*), 0.5 (*not low*), 0.7 (*high*), and 0.9 (*very high*).
- P_i , patients in hunger control of i has values: 0.1 (*very low*), 0.3 (*low*), 0.5 (*not low*), 0.7 (*high*), 0.9 (*very high*).

Similarly, the values of the output attribute can be given as; S_i has values varying from 0.01 (*negligible*) to 0.99 (*very high*).

From the aforementioned values of the input and output attributes of the BPNN, there is an input–output relationship schema has been employed for creating the initial training dataset (*seen data*) for the BPNN and the details of each relationship schema are given as:

- A person with a *very high/high* value of L , a *very high/high* value of H , a *very high/high* value of T , and a *very high/high* value of P would always holds a *very high/high* value of S .
- A person with a *very low/low* value of L , a *very low/low* value of H , a *very low/low* value of T , and a *very low/low* value of P would always holds a *very low/low* value of S .

- A person with a *low* value of L , a *very high/high* value of H , a *high* value of T , and a *high* value of P would always holds a *high* value of S .
- A person with a *not low* value of L , a *not low* value of H , a *not low* value of T , and a *not low* value of P would always holds a *not low* value of S .
- A person with a *high* value of L , a *low* value of H , a *low* value of T , and a *low* value of P would always holds a *low* value of S .
- A person with a *low* value of L , a *high* value of H , a *high* value of T , and a *high* value of P would always holds a *high* value of S .
- A person with a *very high/high* value of L , a *very low/low* value of H , a *very high/high* value of T , and a *very low/low* value of P would always holds a *very low/low* value of S .
- A person with a *not low* value of L , a *very low* value of H , a *high* value of T , and a *very low* value of P would always holds a *very low* value of S .
- A person with a *high* value of L , a *not low* value of H , a *not low* value of T , and a *not low* value of P would always holds a *not low* value of S .
- A person with a *low* value of L , a *not low* value of H , a *high* value of T , and a *very low* value of P would always holds a *low* value of S .

Based on the mentioned input-output relationship schema of the attribute of the BPNN, there are seventy five input–output data patterns are generated for the seen dataset. The over fitting issue of the BPNN is cleared with the correlation based detection algorithm [20].

Table 5 shows the satisfaction test results generated for 20 persons with their unseen input values (called *unseen* dataset) by the trained BPNN. The people are identified with the value of S more than 0.6, are considered as of “satisfied” status. From Table 5, it can be noticed that there are 10 persons (identified as 2, 3, 4, 6, 9, 10, 12, 15, 17, and 18) who are satisfied with the food recommendation system.

The prediction accuracy of the BPNN is 98.6% and is calculated based on the results generated by its seen and unseen datasets by using an evaluation matrix [5].

The initial version of this research [1] was simulated as an android application and its efficiency in estimating both overweight and daily needed calorie values of people with abnormal BMI has been successfully verified from its users. One of the aims of this research is to develop and test the performance of a novel food recommendation scheme as part of a gadget based health computation system. Therefore, the C++ language is used to simulate the various sections of this research to estimate their performance and accuracy individually. From the results of various carried out simulations, it is confirmed that the concept of the presented food recommendation system is an appropriate one for any health apps. It is due to the issue of the page size limitation of the conference, all the carried out simulations are not shown in this paper.

Table 5 Satisfaction values of twenty persons by the BPNN

| Person | <i>L</i> | <i>H</i> | <i>T</i> | <i>P</i> | <i>S</i> |
|--------|----------|----------|----------|----------|----------|
| 1 | 0.37 | 0.21 | 0.28 | 0.18 | 0.19 |
| 2 | 0.70 | 0.83 | 0.86 | 0.68 | 0.74 |
| 3 | 0.83 | 0.93 | 0.75 | 0.89 | 0.87 |
| 4 | 0.43 | 0.75 | 0.49 | 0.86 | 0.74 |
| 5 | 0.56 | 0.39 | 0.41 | 0.41 | 0.38 |
| 6 | 0.78 | 0.68 | 0.75 | 0.73 | 0.72 |
| 7 | 0.64 | 0.45 | 0.63 | 0.44 | 0.43 |
| 8 | 0.28 | 0.26 | 0.37 | 0.24 | 0.25 |
| 9 | 0.97 | 0.87 | 0.79 | 0.93 | 0.87 |
| 10 | 0.98 | 0.84 | 0.87 | 0.89 | 0.88 |
| 11 | 0.33 | 0.26 | 0.18 | 0.28 | 0.26 |
| 12 | 0.76 | 0.67 | 0.73 | 0.66 | 0.65 |
| 13 | 0.56 | 0.53 | 0.61 | 0.51 | 0.52 |
| 14 | 0.37 | 0.23 | 0.28 | 0.27 | 0.26 |
| 15 | 0.98 | 0.99 | 0.97 | 0.98 | 0.98 |
| 16 | 0.72 | 0.44 | 0.35 | 0.51 | 0.46 |
| 17 | 0.97 | 0.78 | 0.84 | 0.77 | 0.79 |
| 18 | 0.58 | 0.63 | 0.71 | 0.63 | 0.62 |
| 19 | 0.53 | 0.16 | 0.54 | 0.17 | 0.15 |
| 20 | 0.48 | 0.54 | 0.57 | 0.57 | 0.56 |

6 The Complete Structure of the Food Recommendation System

The complete structure of the food recommendation system is shown in Fig. 2 as a flowchart.

7 Conclusion

The food recommendation system has adopted widely accepted scientific measurements such as BMI, and BMR for estimating its users' overweight, and DNC_{sat} values. The employed k -NN algorithm has shown a great performance in its food sets selection process based on the DNC_{sat} values. The facility of providing the number of days for a person to reach in a state of healthy BMI level along with the recommended food has made the system more reliable to its users. The BPNN section of the system could be able to show success in predicting users' satisfaction with the recommended food set. The food recommendation system is one of the

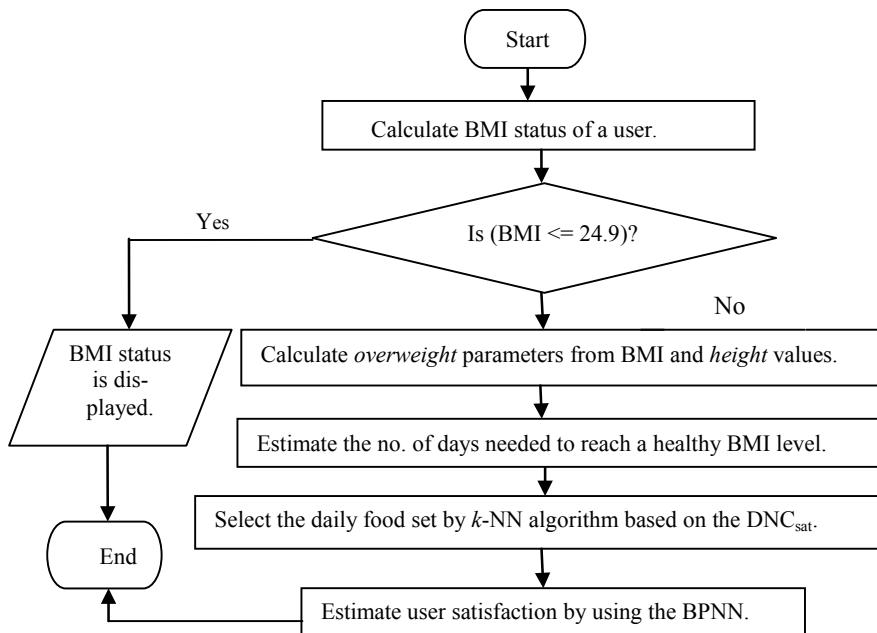


Fig. 2 Complete structure of the food recommendation system

mandatory life systems for supporting people living with anomalous body weight and food habits. At this point, the system does not categorize its user's food interests into a vegetarian and non-vegetarian class. In the future, this research will surely be enhanced with users' food interests, and it will make use of the value of the user satisfaction level for that.

References

1. K.G. Anilkumar, Recommended weight prediction system based on BMI, BMR, food calorie and a neural network, in *2nd International Conference on Intelligent Informatics and BioMedical Sciences (ICIIBMS 2017)* (IEEE, Okinawa, Japan, 2017), pp. 15–22
2. <https://www.geeksforgeeks.org/k-nearest-neighbours> Last accessed 30 Jan 2020
3. [https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn.](https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/) Last accessed 3 Feb 2020
4. B. Wolfgang, N. Peter, E.K. Robert, D.F Frank, *Genetic Programming An Introduction* (Morgan Kaufmann Publishers, San Francisco, 1998)
5. L.P. David, A.K. Mackworth, *Artificial Intelligence Foundations of Computational Agents*, 2nd edn. (Cambridge University Press, Cambridge, 2018)
6. M. Negnevitsky, *Artificial Intelligence-A Guide to Intelligent Systems*, 2nd edn. (Addison Wesley, Boston, 2005)
7. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edn. (Pearson Education, New Jersey, 2004)

8. V.B. Rao, H.V. Rao, *Neural Networks & Fuzzy Logic* (BPB Publications, New Delhi, 1996)
9. https://www.nhlbi.nih.gov/health/educational/lose_wt/BMI/bmicalc.htm. Last accessed 4 Feb 2020
10. <https://medium.com/analytics-vidhya/how-to-build-a-restaurant-recommendation-engine-part-2-71e2d0721084>. Last accessed on 8 Feb 2020
11. C. Pathanjali, E.S. Vimuktha, G. Jalaja, A. Latha, A comparative study of Indian food image classification using k -NN and SVMs. *Int. J. Eng. Technol.* **7**(3.12), 521–525 (2018)
12. N.S. Sabounchi, H. Rahmandad, A. Ammerman, Best fitting prediction equations for basal metabolic rate: informing obesity interventions in diverse populations. *Int. J. Obes. (London)* **37**(10), 1364–1370 (2013)
13. T.K. Abdel-Hamid, Modeling the dynamics of human energy regulations and its implications for obesity treatment. *Syst. Dyn. Rev.* **18**(4), 431–471 (2002)
14. <https://www.healthline.com/nutrition/how-many-calories-per-day#intake-averages>. Last accessed 13 Feb 2020
15. K.J. Rothman, BMI-related errors in the measurement of obesity. *Int. J. Obes. (London)* **32**, 56–59 (2008)
16. <https://www.healthstatus.com/calculate/body-mass-index/>. Last accessed on 2 Feb 2020
17. <https://www.sharecare.com/group/sharecare-fitness>. Last accessed on 9 Feb 2020
18. https://www.sciencedaily.com/terms/basal_metabolic_rate.htm. Last accessed on 1 Feb 2020
19. <https://catalog.data.gov/dataset/mypyramid-food-raw-data-f9ed6>. Last accessed on 30 Jan 2020
20. K.G. Anilkumar, A subjective job scheduler based on backpropagation neural network. *Hum. Centric Comput. Inf. Sci. (HCIS) (A Springer Open Journal)* **3**(1), 17 (2013). <https://doi.org/10.1186/2192-1962-3-17>

Complexity Reduced Bi-channel CNN for Image Classification



Nivea Kesav and M. G. Jibukumar

Abstract Convolutional neural networks (CNN) with deep architectures have paved way for immense opportunities where any complex mechanism can be undertaken and analyzed. The main drawback of all such architectures is their high number of trainable parameters. This increases its complexity and makes it difficult for real-time processing. We suggest in this paper a multi-channel-based design with shallow layers that can efficiently be trained and tested with less complexity as compared to the existing deep architectures. The performance is achieved by using the concept of a side channel with a main channel. The main concentration is to reduce the parameters to be trained as much as possible with slight compromise in the accuracy. Different values of filter sizes are given, and the output accuracy was observed for different cases. The proposed network was tested on a brain tumor-type database, and it successfully classified the images comprising of meningioma and pituitary tumor. The entire network performance is evaluated by comparing it with two deep architectures known as AlexNet and VGG16. The results show a huge drop in the number of parameters to be trained with much less execution time and comparable accuracy.

Keywords Deep learning · Multi-channel CNN · Complexity reduction

1 Introduction

Machine learning is a new emerging era which solves almost all human day-to-day issues where the problems are learned by the machine itself by several iterations and conclusions. It has found magnificent applications in various engineering fields. The process of making a computer program learn a task and analyze a situation

N. Kesav (✉) · M. G. Jibukumar
Cochin University of Science and Technology, Cochin, Kerala, India
e-mail: nivea11093@gmail.com

M. G. Jibukumar
e-mail: jibuthattakuzha@gmail.com

has immensely reduced the amount of human intervention required. It has done remarkable changes in the areas like image, video and signal processing, computing, networking, communication, finance, etc. Another major advantage that prevails the machine learning era is that in the case of image processing it is often seen that the images need not be heavily preprocessed as in the conventional image processing techniques. Deep learning is a part of machine learning paradigm that has a deep architecture with a higher number of layers. Deep learning is termed as a form of representation learning [1] in which a system automatically recognizes several patterns and representations from the input data when the data flows from initial to deeper layers. It can extract and represent various features both high dimensional and low dimensional. Then, we also have the multi-layer architectures where the data can flow through more than one path simultaneously. In spite of the high computational complexity, deep learning paves way for a much user-friendly platform that enables us to adapt with the working environment easily. Our research work is mainly focused on the multi-channel architecture. We have used the brain tumor-type database extensively to analyze the performance of our architecture. Brain tumor is the unusual growth of tissue in the brain which results in issues related to the normal functioning of the human body.

There are many researches existing in the area of deep learning that helps in classifying and segmenting the images, and excellent results have been obtained [2–5]. But most of the studies rely mainly on transfer learned networks of existing architectures which are highly complex and hard to compute [2]. In [3] authors have also included a fine-tuning to the transfer learned network of existing architectures to improve the overall accuracy. Such complex networks have several disadvantages including increased number of parameters, unavailability of storage space to store them, high execution timings, and difficulty in implementation. However, it is found that these networks may contain a lot of redundant data in [6]. The performance of a CNN can be improved by exploiting this redundancy among filters and excluding them so as to optimize the architecture complexity. The method of removing highly sparse filters helps to prune the network. Wang et al. proposed a set of correlative filters in [4] for shallow layers and translational filters for deeper layers. Their architecture efficiently decreased the amount of filter parameters to be trained as they aimed at pre-setting the filters before execution. Later, it has been found that pruning just the parameters of a filter will not help in reducing the computational cost in [5]. Instead, the entire filter should be pruned to have a serious impact on the complex nature. Instead of using the sparsity concept, they have focused on getting the l_1 norm of the kernel weights to decide which filter to be pruned. The authors in [7] discussed about reducing the size of a deep CNN so as to accommodate it into a local devices. CNN can perform well with local computing than cloud-based scenario due to its low latency problems. The redundant kernel removal helps to increase the computational speed at a higher rate. Later, the effects of different filter dimensions have been discussed in [8] and parameters like size of the filter, stride, and number of filters have been analyzed. The pattern of decreasing accuracy with increasing filter size has been also observed by them. The authors in [9] have found out that the accuracy increases for increase in input window size, reaches a saturation level, and then decreases.

CNN has excellent capabilities to classify images with very less preprocessing. Multi-channel CNN has been extensively used in various areas of image processing. It gives a separate independent path for information flow. The authors in [10] describe an idea of using double-channel CNN to classify multi-spectral RGB-NIR images. Classification of such type of images is highly challenging and can be done efficiently using two-channel CNN having identical filters and layers. But the input is RGB (visible) data for first channel and NIR (near infrared) data for the second channel. A multi-channel CNN can efficiently identify different hand postures [11]. Here, the three channels have identical layers but the input to the three channels is differently processed outputs of same images. This method can also be accomplished to recognize facial expressions using two dissimilar channels [12] where the first channel is a CNN-based channel and the other is made up of auto-encoders. As mentioned earlier, we have considered the brain tumor-type dataset for evaluation of our architecture. In our proposed architecture, we have made use of the concept of multi-channel CNN to give a side channel information. Compared to the earlier works [10–12], we are having two dissimilar channels and we focus on sending the same information through both the channels which have different filter sizes. We have evaluated the effect of different filter sizes on the performance of the model. Our simulation results have proved to increase the overall accuracy of the system with very few parameters. So even if we decrease the parameters, the addition of a side channel will help to maintain the accuracy levels at a better rate as the side channel can give additional information that will help to train the network better than a single-channel network. In this paper, Sect. 2 consists of the methodology adopted, Sect. 3 considers the simulation results, and Sect. 4 concludes the work.

2 Methodology

2.1 Multi-channel Convolutional Neural Network

CNN is a class of deep learning techniques which can efficiently work on images. The CNN constitutes of a convolutional layer, a pooling layer, and a ReLU layer as an activation function. The max pooling or average pooling layer calculates the maximum or average of a given subset and helps greatly in the dimensionality reduction of the feature map, and the ReLU activation function gives an output for max (0, input). The important study is conducted on the number of parameters (P) to be trained for a particular layer in the system, and it is given by the equation:

$$P = ((a \times b \times g) + 1) \times k \quad (1)$$

where $a \times b$ is the dimension of the filter, g is the number of feature maps from the previous layer as input, and k is the number of feature maps to the next layer as output. Final layer is the fully connected layer which is connected to the softmax

layer which gives the classification output. Multi-channel CNN has two or more independent separate paths for information flow. It can either have same input for all the channels or different inputs for each channel; similarly, it can have same channels or different channels [12]. Each path works as a single independent CNN, and final outputs are combined in the fully connected layer. The loss function and weight updating schemes are same as in the single-channel CNN. A general multi-channel CNN consists of ‘ N ’ independent paths that constitute an N -channel CNN. The data will pass through the given N paths separately. Each path has got M layers. Each path will function on its own when given an input. Later, all the N paths are merged in a fully connected layer. The fully connected layer combines all the different output feature maps as a one hot vector and is further processed.

2.2 Proposed Architecture

The suggested architecture makes use of a multi-channel CNN. Multi-channel CNN has more than one path for forward data flow as shown in Fig. 1. The idea used here is to give an effect of a side channel information through the secondary channel along with the main channel. So in effect our architecture comprises of two channels which are a main channel and a side channel. The image processed through the side channel passes through a filter and acts as a side channel information to aid the accuracy of the entire system at the same time reducing the complexity compared to the revolutionary architectures like the AlexNet [13] and VGG16 [14]. The efficiency is brought about by a low complexity system accompanied with side channel information.

In our model, the input image has a dimension of $227 \times 227 \times 3$ pixels. The main channel consists of two convolutional layers with a single max pooling layer in between. The first layer gives 16 feature maps as output and the second layer gives 64 feature maps as output. The pooling layer has a filter of size 3×3 . The

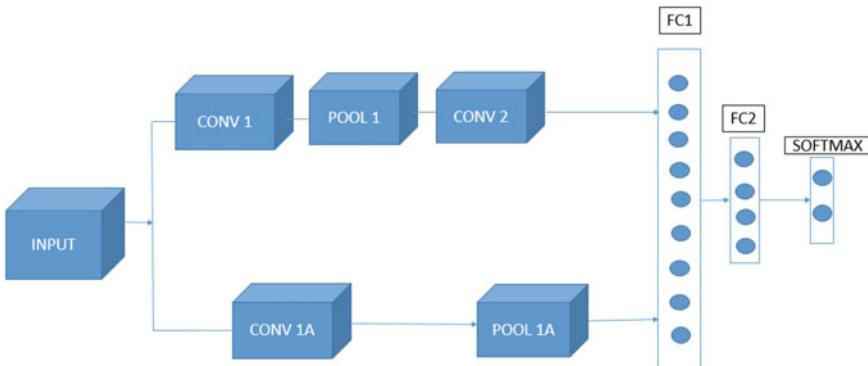


Fig. 1 Proposed architecture

side channel consists of a single convolutional layer and a max pooling layer of the filter dimension 3×3 . This layer gives an output of 64 feature maps. We have only one convolutional layer in side channel as we wish to give only a single coarse side information. Both the outputs are connected to fully connected (FC) layers. The first part of the FC layer consists of 100 neurons as output, and the second part reduces it into 50 neurons as output. Finally, we have a softmax classification layer that classifies the output into meningioma or pituitary tumor. Activation functions (ReLU) are also associated with different convolutional layers. In our architecture, the dimensions of the filters of convolutional layers are varied for several different cases to analyze the performance outcome. The optimization technique for gradient updating mechanism is done by adaptive moment estimation (ADAM). It is a combination of momentum and root mean square propagation. It replaces the general technique of stochastic gradient descent whose learning rate is constant for all weight updates and has given an improved performance compared with conventional existing technology.

3 Experiments and Results

3.1 Software Requirements

The entire research was conducted using MATLAB 2018a. The system specifications include 8 GB RAM single CPU processing under the clock frequency of 2.6 GHz. The aim of the proposed architecture was to obtain maximum efficiency with minimum software so that it can be best suitable for real-time processing scenario without wastage of resources.

3.2 Dataset Collection and Pre-processing

The dataset used for the purpose of research is the commonly and freely available brain tumor type dataset from figshare [15]. Brain tumor is the abnormal growth of tissue in the brain, and it can lead to dysfunctioning of proper brain activities. The three major tumor types in existence is glioma, meningioma, and pituitary tumor. In this research, we are concentrating on meningioma and pituitary tumor as we have a two-class classification problem. Meningioma is a slow-growing benign tumor, and it is mostly found at the outer coverings of brain just under the skull. Pituitary gland is located in the base part of the brain, and the tumor associated with it is known as the pituitary tumor. The main criterion that helps us to classify them is the location of the tumor as shown in Fig. 2. The dataset is composed of 3064 brain MRI slices of 233 patients identified with meningioma, glioma, and pituitary tumor. These images belong to T1-contrast enhanced MRI modality and include all the three views coronal, axial, and sagittal. From that, 1638 images are used for our two-class

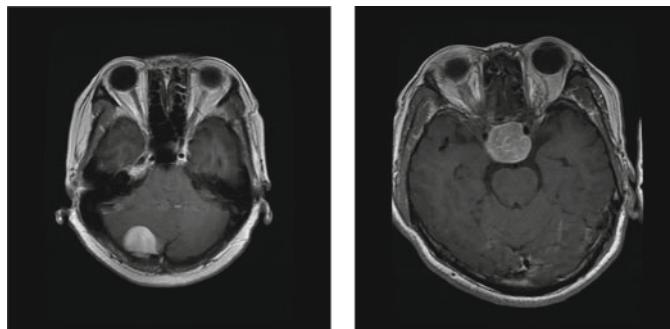


Fig. 2 MRI images of meningioma and pituitary tumor, respectively

classification problem. There are 708 images of 82 patients in meningioma category and 930 images of 62 patients in pituitary tumor category.

From a total of 708 images of meningioma patients, 80% are chosen for training and 20% for testing the system. Similarly, out of 930 images of pituitary tumor patients same division is done for training and testing. We have followed the design of AlexNet just for the input layer, which requires an image of size $227 \times 227 \times 3$. The images that we acquired from the given dataset reference are all grayscaled images of dimension 512×512 pixels. We had to resize the images into 227×227 pixels, and we repeated it three times to get the three channels as the input.

3.3 Simulation and Results

The performance of the system is calculated from the confusion chart by analyzing the following parameters:

- Sensitivity (Recall) = $\frac{TP(\text{True Positive})}{TP+FN(\text{False Negative})}$
- Specificity = $\frac{TN(\text{True Negative})}{TN+FP(\text{False Positive})}$
- Precision = $\frac{TP}{TP+FP}$
- Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$
- F1 score = $\frac{2*\text{recall}*\text{precision}}{\text{recall}+\text{precision}}$

The learning algorithm is chosen to be ADAM, mini batch size is 30, initial learning rate is fixed at 0.0003, iterations per epoch is 35, and the system is trained for a total of 10 epochs. Data augmentation is also applied for enhancing the accuracy. The analysis is carried out in four different cases, and the factors like accuracy, training parameters, and execution time are analyzed. The four cases are explained briefly in the next following paragraphs.

In the first method (case 1), we will be keeping the main channel filter size constant and will be training the side channel with an increasing pattern of filter sizes. The

side channel variation is tested for three cases of main channel 3×3 , 5×5 and 7×7 . The results are then observed as shown in Fig. 3. We see that the accuracy increases at first until the filter size is 13×13 . There it reaches a maximum limit and then the accuracy starts to deplete. So a very high filter size beyond a limit is not much useful for the side channel. The pattern is same for all the three values of constant main channel filter size. Therefore, we come to a conclusion that the highest accuracy is obtained when the side channel filter size is 13×13 and this is chosen as the optimum filter size for the side channel. From Table 1, we can observe that for all the three values of constant main channel filter size, 3×3 outperforms with an overall accuracy of 94.86% and with a total number of training parameters 10,082,692, which is very less when compared to the conventional highly complex structures like AlexNet (58,289,794) and VGG16 (138 million). So this proves that

Fig. 3 Accuracy versus side channel filter size for fixed main channel

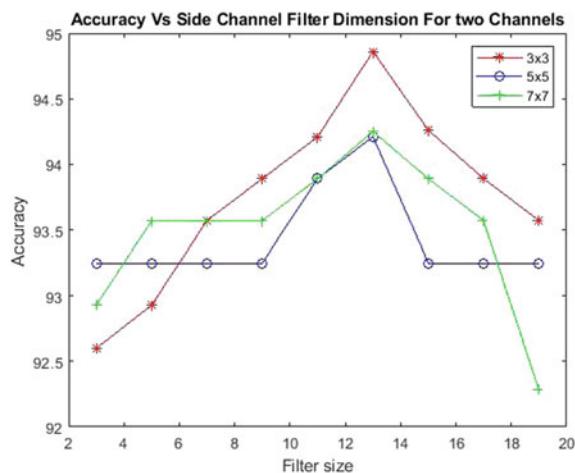


Table 1 Variation in accuracy and training parameters for side channel filter sizes

| Side channel filter size | Accuracy (%) for main channel 3×3 | Accuracy (%) for main channel 5×5 | Accuracy (%) for main channel 7×7 | No. of parameters for main channel filter size 3×3 |
|--------------------------|--|--|--|---|
| 3×3 | 92.60 | 93.25 | 92.93 | 10,051,972 |
| 5×5 | 92.93 | 93.25 | 93.57 | 10,055,044 |
| 7×7 | 93.57 | 93.25 | 93.57 | 10,059,652 |
| 9×9 | 93.89 | 93.25 | 93.57 | 10,065,796 |
| 11×11 | 94.21 | 93.89 | 93.89 | 10,073,476 |
| 13×13 | 94.86 | 94.21 | 94.25 | 10,082,692 |
| 15×15 | 94.26 | 93.25 | 93.89 | 10,093,447 |
| 17×17 | 93.89 | 93.25 | 93.57 | 10,105,732 |
| 19×19 | 93.57 | 93.25 | 92.28 | 10,119,556 |

the use of such a low complex design with a side channel can compensate for decrease in parameters.

In case 2, we vary the main channel by keeping the side channel fixed. The side channel dimension is chosen to be 13×13 as we got a highest accuracy for the above-mentioned filter size in the previous case. The filter size of the main channel is varied in an increasing fashion starting from 3×3 , and the accuracy is found to be considerably decreasing as shown in Fig. 4. It is seen that when we increase the size from 3×3 to 11×11 , the number of parameters are also increasing which is not advisable as our main aim is to reduce the parameters of the design. The Table 2 gives the accuracy and number of parameters observed in this experiment. We got a maximum accuracy of 94.86% for the main channel filter size 3×3 with the side channel filter having dimension 13×13 . So we fix the main channel filter size as 3×3 . This observation gives assurance to our previous result from Fig. 3.

In the third case, we concentrate on analyzing the performance for a single channel CNN. Single channels usually use more number of layers as it is clear from famous architectures like AlexNet, VGG Net, etc. Still we checked our shallow architecture by just considering the main channel only. The results show that there was not any much improvement in the accuracy levels for a single-channel case as shown in Fig. 5. Table 3 gives the number of parameters and accuracy values, and a maximum

Fig. 4 Accuracy versus main channel filter size for fixed side channel

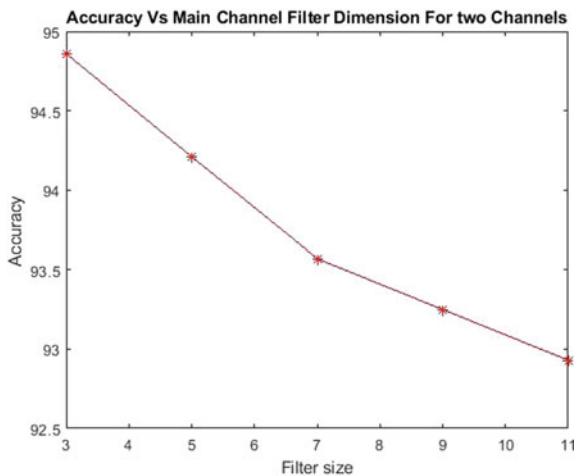


Table 2 Variation of accuracy and training parameters for change in main channel filter sizes

| Main channel filter size | Accuracy (%) | No. of parameters |
|--------------------------|--------------|-------------------|
| 3×3 | 94.86 | 10,082,692 |
| 5×5 | 94.21 | 10,099,844 |
| 7×7 | 93.57 | 10,125,572 |
| 9×9 | 93.25 | 10,159,876 |
| 11×11 | 92.93 | 10,202,756 |

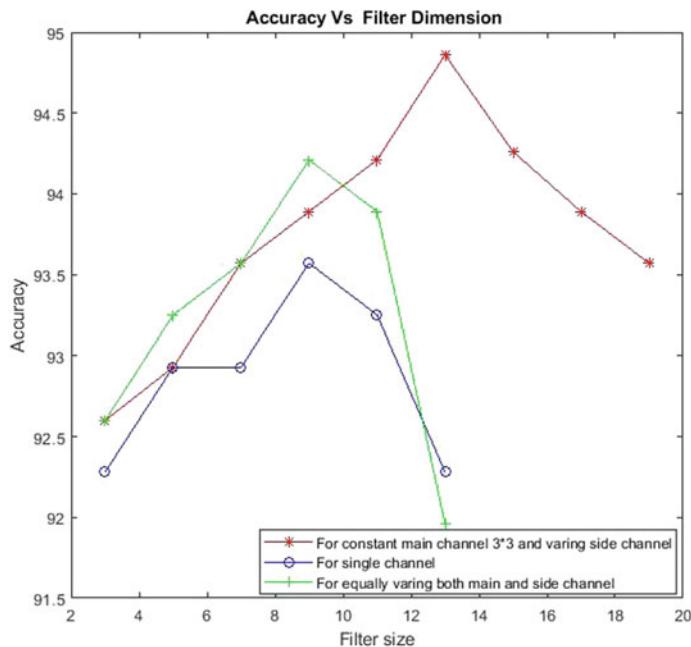


Fig. 5 Accuracy versus filter size comparison of cases 1, 3, and 4

Table 3 Variation of accuracy and training parameters for change in single channel filter sizes

| Main channel filter size | Accuracy (%) | No. of parameters |
|--------------------------|--------------|-------------------|
| 3×3 | 92.28 | 5,032,580 |
| 5×5 | 92.93 | 5,049,732 |
| 7×7 | 92.93 | 5,075,460 |
| 9×9 | 93.57 | 5,109,764 |
| 11×11 | 93.25 | 5,152,644 |
| 13×13 | 92.28 | 5,204,100 |

accuracy of only 93.57% is observed. But our first case gave a performance of 94.86% with two channels. So the single channel with shallow layers was proved to be not a good choice for the classification. This again helps to prove our result that multi-channels can perform well with shallow architectures than a single channel.

In case 4, filter dimensions of both the channels are varied in an increasing manner. Whatever filter size is chosen for the main channel, the same size is again chosen for the side channel. The results show that the accuracy increases, reaches a maximum saturation point, and later shows a decreasing nature. From Table 4, the maximum accuracy is at 9×9 filter size for both the main and side channels. But while we compare it with the case 1 of main channel size 3×3 and side channel size 13×13 , this case has less accuracy of just 94.21% whereas the above-mentioned case has an

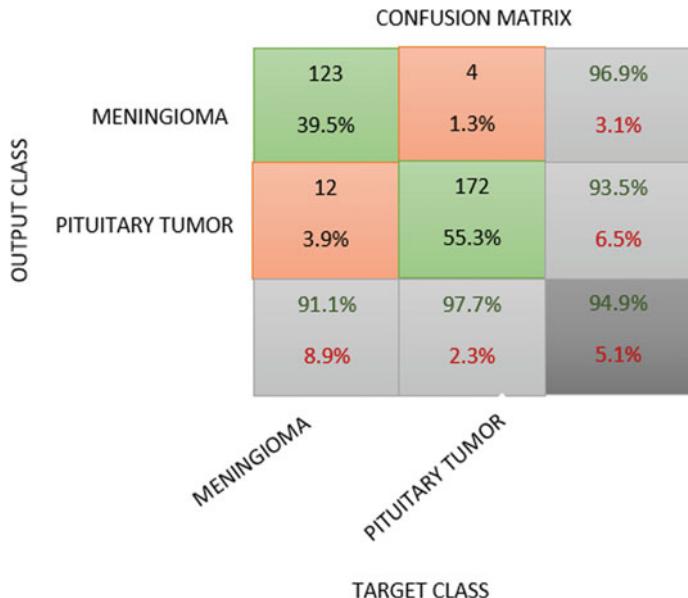
Table 4 Variation of accuracy and training parameters for equally varying filter sizes of two channels

| Main channel filter size | Accuracy (%) | No. of parameters |
|--------------------------|--------------|-------------------|
| 3×3 | 92.60 | 10,051,972 |
| 5×5 | 93.25 | 10,072,196 |
| 7×7 | 93.57 | 10,102,532 |
| 9×9 | 94.21 | 10,142,980 |
| 11×11 | 93.89 | 10,193,540 |
| 13×13 | 91.96 | 10,254,212 |

accuracy of 94.86%. While we compare the number of parameters, we see that case 4 has more number of parameters than case 1. So again we can strongly conclude that case 1 is better than this. Cases 1, 3 and 4 are compared in Fig. 5 from which we can easily identify that maximum accuracy is for case 1 with main channel filter size 3×3 and side channel size 13×13 .

From the above given results, we finally came to a conclusion that giving a high dimensional filter as a side channel will indeed enhance the performance of the overall system. The effective architectural design is chosen to be having main channel filter size of 3×3 and side channel filter size of 13×13 with an overall accuracy of 94.86%. The performance improvement using these filter sizes may be due to the fact that main channel requires a lower filter size to acquire huge information with local complex features and side channel requires only a coarse view of the image with basic components resulting in the use of a higher dimensional filter. The different performance parameters are calculated from the confusion chart shown in Fig. 6. Out of the total number of testing samples, 123 samples are correctly classified for meningioma and 172 are correctly classified for pituitary tumor. A comparison study of our proposed architecture was done with the AlexNet and VGG16. In Table 5, we observe that all the different performance parameters have comparable values even though the proposed architecture has very less complexity.

The main contribution of our work is that this proposed architecture has got very less parameters when compared to the complex AlexNet and VGG16. The total trainable parameters have reduced from 58,289,794 to 10,082,692 resulting in a percentage decrease of 82.702% in the case of AlexNet. Also while comparing with VGG16 the two channel system with side information has achieved a huge decrease in parameters. At the same time, there is no much compromise in the performance parameters when compared to AlexNet and VGG16. The tradeoff is found here in the case of accuracy when compared with both architectures. An enormous decrease in the execution time is also observed. The receiver operator characteristics (ROC) is used to analyze the performance of the system which gives the relationship between true positive and false positive rate as shown in Fig. 7. The area under the curve (AUC) is then computed, and the AUC for both the classes are above 90% which implies that the classifier has a good performance.

**Fig. 6** Confusion matrix**Table 5** Comparison of performance parameters of proposed network with AlexNet and VGG16

| | AlexNet | VGG16 | Bi-Channel CNN |
|--------------------|------------|-------------|----------------|
| Sensitivity | 0.94776 | 0.9419 | 0.9685 |
| Specificity | 0.9548 | 0.9657 | 0.9347 |
| Precision | 0.94074 | 0.9558 | 0.9111 |
| Accuracy | 0.9517 | 0.9550 | 0.9486 |
| F1 score | 0.94423 | 0.94879 | 0.9386 |
| AUC meningioma | 0.9993 | 0.9906 | 0.9837 |
| AUC pituitary | 0.9993 | 0.9906 | 0.9837 |
| Execution time (s) | 1047.3799 | 10783.8849 | 265.396706 |
| Total parameters | 58,289,794 | 138,000,000 | 10,082,692 |

4 Conclusion

We have suggested in this paper a new low complex architecture, bi-channel CNN. The results show that the network has reduced computational cost and less execution time but with a tradeoff of slightly decreased accuracy as compared to the existing architectures. In our analysis, the system with 3×3 filters in main channel and 13×13 filter in the side channel is found to be the best architecture. The test input images are classified with an accuracy of 94.86% as compared to AlexNet which

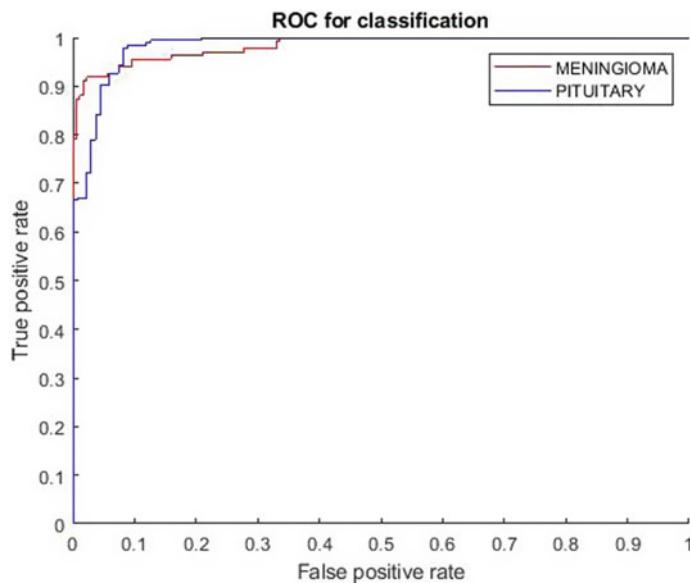


Fig. 7 ROC curve

has an accuracy of 95.17% and VGG16 having 95.50%. In spite of this reduction in accuracy, we are able to achieve an 82.702% decrease in the total trainable parameters as compared to AlexNet and 92.693% decrease when compared to VGG16. Here, we have only considered a two-channel system for the analysis. The effect of the number of channels and the information acquired by both the channels will be dealt with in the future research.

References

- Y. Lecun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**(7553), 436–444 (2015)
- S. Deepak, P.M. Ameer, Brain tumor classification using deep CNN features via transfer learning. *Comput. Biol. Med.* **111**, 103345 (2019)
- Z.N.K. Swati et al., Brain tumor classification for MR images using transfer learning and fine-tuning. *Comput. Med. Imaging Graph.* **75**, 34–46 (2019)
- H. Wang, P. Chen, S. Kwong, Building correlations between filters in convolutional neural networks. *IEEE Trans. Cybern.* **47**(10), 3218–3229 (2017)
- H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, *Pruning Filters for Efficient ConvNets* (2016), no. 2016, pp. 1–13
- C.T. Liu et al., Computation-performance optimization of convolutional neural networks with redundant filter removal. *IEEE Trans. Circ. Syst. I Regul. Pap.* **66**(5), 1908–1921 (2019)
- C.F. Chen, G.G. Lee, V.Sritapan, C.Y. Lin, Deep convolutional neural network on iOS mobile devices (Invited Paper), in *International Workshop on Signal Processing Systems SiPS Design Implement.* (2016), pp. 130–135

8. M. R. Islam and N. Rishad, “Effects of filter on the classification of brain MRI image using convolutional neural network, in *4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEICT) 2018*, pp. 489–494 (2019)
9. T. Sinha, B. Verma, A. Haidar, Optimization of convolutional neural network parameters for image classification, in *IEEE Symposium Series on Computational Intelligence SSCI 2017—Proc.*, vol. 2018 (2018), pp. 1–7
10. J. Jiang, X. Feng, F. Liu, Y. Xu, H. Huang, Multi-spectral RGB-NIR image classification using double-channel CNN. *IEEE Access* **7**, 20607–20613 (2019)
11. P. Barros, S. Magg, C. Weber, S. Wermter, A multichannel convolutional neural network for hand posture recognition, in *Lecture Notes Computer Science (including Subseries Lecture Notes Artificial Intelligence Lecture Notes Bioinformatics)*, vol. 8681 LNCS, no. Icann, pp. 403–410 (2014)
12. D. Hamester, P. Barros, S. Wermter, Face expression recognition with a 2-channel convolutional neural network, in *Proceedings of International Joint Conference Neural Networks* (2015)
13. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in *Handbook Approximation Algorithms Metaheuristics* (2007), pp. 45-1–45-16
14. S. Tammina, Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *Int. J. Sci. Res. Publ.* **9**(10), 143–150 (2019)
15. Brain tumor dataset. Accessed date July 2019. https://figshare.com/articles/brain_tumor_dataset/1512427

An Approach to Mitigate the Risk of Customer Churn Using Machine Learning Algorithms



Debajyoti Mukhopadhyay , Aarati Malusare, Anagha Nandanwar, and Shriya Sakshi

Abstract In various service-based industries such as telecom industry, life insurance, hospitality, banking, and gaming, Churn Prediction plays an important role. Companies are trying to establish means for predicting potential clients to turnover in the telecom sector. Therefore, it is crucial to identify the factors that rising the churn of customers and take the appropriate steps and reduce the churn. Hence the purpose of our research is to establish the model of churn prediction. The cycle where one user leaves one company and enters another is called churn. This paper would explore how to identify customers who could churn, using machine learning techniques to forecast, and helping to represent large datasets in graph form.

Keywords Customer churn · Machine learning · Classification · Support vector machine · Random forest · KNN · Prediction · CRM

1 Introduction

There are indeed a variety of telecommunications industries available, yet according to our requirements, we can choose either. Churn Estimation is a crucial feature of the serviced industry. Churn can be identified as the customer who moves from one

D. Mukhopadhyay (✉) · A. Malusare · A. Nandanwar · S. Sakshi
Computer Science Department, Mumbai University, Mumbai, Maharashtra, India
e-mail: debajyoti.mukhopadhyay@gmail.com

A. Malusare
e-mail: aaratipmalusare1998@gmail.com

A. Nandanwar
e-mail: anaghanandanwar16@gmail.com

S. Sakshi
e-mail: shriya.sakshi03@gmail.com

D. Mukhopadhyay
WIDiCoReL Research Lab, Mumbai, Maharashtra, India

supplier to the other. Holding the existing customer is always important to the firms, rather than searching for a future one.

Predictive models in the near future offer the correct idea about churners which enables to provide a retention solution. This paper introduces a new prediction model based on strategies and techniques on Machine Learning (ML). The current proposal consists of seven phases which are: problem domain recognition, data collection, data set analysis, classification, clustering, use of information and data representation. Decision Tree, Support Vector Machine and Random Forest are the algorithms employed in this framework.

It is therefore not easy to keep the customers because of a competitive climate. Industries are now using innovative technology to give their consumers improved services so that they can maintain them. In doing so, it is important to recognise such customers who are expected to leave the business in the immediate future as losing them will lead to the company's substantial profit loss. It is called the Churn Prediction method.

Machine learning approaches are found to become much more accurate in predicting client churn from a past number of years of study. Creating an appropriate model for churn estimation is an essential activity involving the selection of an appropriate machine learning algorithm with lots of research right type recognition of features forming large dataset.

We got a thought about this topic by searching various reference papers. The contribution of work starts with data analysis and then making the data perfect to pass through the model by performing action for missing data, feature scaling, categorical values. We are going to find which algorithm is best for our dataset by making the model for a various algorithm.

2 Literature Review

Wang et al. [1], is Churn Prediction Model as a cost-sensitive question of classification. They used the word cost-sensitive because the manner in which the model was built to identify the customers into churn and non-churn category would determine the company's overall benefit. The authors had shown the classification performance and the rate of misclassification in the suggested study.

Lu et al. [2] Churn Prediction System proposed to use Boosting Algorithm. Based on the weight given by the Boosting algorithm, customers were separated into two groups in this paper. On each Base Classifier, the Logistic Regression is being used to estimate churn customers. The Boosting outcome was a good classifier for analysis of the Churn Potential.

Peng et al. [3–9] In these papers the customer predicts churning using different R packages and builds a classification model and trains by giving it a database and after training, they can divide the records into churns or non-churns. For this, they use a type of logistic regression.

Dahiya and Bhatia [10] Customer churn plays a significant role in customer relationship management (CRM), using a number of machine learning algorithms to forecast customer churn and discovering ensemble learning is the best way to forecast customer churn.

Praveen et al [11] According to the study based on the real results, it is shown that the approach not only produces a good result in classification but it also effectively decreases the overall costs of misclassification.

3 Dataset Overview

Each row represents a customer, each column contains customers attributes described on column Metadata. The data set contains 7043 rows (Customer) and 21 columns (features) (Figs. 1 and 2).

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | | |
|----|--------|---------------|---------|------------|--------|--------------|--------------|-----------------|----------------|--------------|------------------|-------------|-------------|-----------------|-----------|------------------|---------------|----------------|-------------|------|------------|
| 1 | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLine | InternetService | OnlineSecurity | OnlineBackup | DeviceProtection | TechSupport | StreamingTV | StreamingMovies | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalChurn | | |
| 2 | Female | 0 | Yes | No | 1 | No | No | phone | DSL | No | Yes | No | No | No | Month-to- | Yes | Electronic | 29.85 | 29.85 No | | |
| 3 | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | No | Yes | No | No | No | One year | No | Mailed chi | 56.95 | 1889.5 No | | |
| 4 | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | Yes | No | No | No | No | Month-to- | Yes | Mailed chi | 53.85 | 108.15 Yes | | |
| 5 | Male | 0 | No | No | 45 | No | No | phone | DSL | Yes | No | Yes | Yes | No | No | No | One year | No | Bank trans | 42.3 | 1840.75 No |
| 6 | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | No | No | No | No | No | Month-to- | Yes | Electronic | 70.7 | 151.65 Yes | | |
| 7 | Female | 0 | No | No | 8 | Yes | Yes | Fiber optic | No | No | Yes | No | Yes | Yes | Month-to- | Yes | Electronic | 99.65 | 820.5 Yes | | |
| 8 | Male | 0 | No | Yes | 22 | Yes | Yes | Fiber optic | No | Yes | No | Yes | No | Yes | Month-to- | Yes | Credit card | 89.1 | 1949.4 No | | |
| 9 | Female | 0 | No | No | 10 | No | No | phone | DSL | Yes | No | No | No | No | Month-to- | No | Mailed chi | 29.75 | 301.9 No | | |
| 10 | Female | 0 | Yes | No | 28 | Yes | Yes | Fiber optic | No | No | Yes | Yes | Yes | Yes | Month-to- | Yes | Electronic | 104.8 | 3046.05 Yes | | |
| 11 | Male | 0 | No | Yes | 62 | Yes | No | DSL | Yes | Yes | No | No | No | No | One year | No | Bank trans | 56.15 | 3487.95 No | | |
| 12 | Male | 0 | Yes | Yes | 13 | Yes | No | DSL | Yes | No | Yes | No | No | No | Month-to- | Yes | Mailed chi | 49.95 | 587.45 No | | |
| 13 | Male | 0 | No | No | 16 | Yes | No | No | No | No | No | No | No | No | Two year | No | Credit card | 18.95 | 326.8 No | | |
| 14 | Male | 0 | Yes | No | 58 | Yes | Yes | Fiber optic | No | No | Yes | No | Yes | Yes | One year | No | Credit card | 100.35 | 5681.1 No | | |
| 15 | Male | 0 | No | No | 49 | Yes | Yes | Fiber optic | No | Yes | Yes | No | Yes | Yes | Month-to- | Yes | Bank trans | 103.7 | 5036.3 Yes | | |
| 16 | Male | 0 | No | No | 25 | Yes | No | Fiber optic | Yes | No | Yes | Yes | Yes | Yes | Month-to- | Yes | Electronic | 105.5 | 2686.05 No | | |
| 17 | Female | 0 | Yes | Yes | 69 | Yes | Yes | Fiber optic | Yes | Yes | Yes | Yes | Yes | Yes | Two year | No | Credit card | 113.25 | 7895.15 No | | |
| 18 | Female | 0 | No | No | 52 | Yes | No | No | No | No | No | No | No | No | One year | No | Mailed chi | 20.65 | 1022.95 No | | |

Fig. 1 Dataset

```

print ("Rows      : ",dataset.shape[0])
print ("Columns   : ",dataset.shape[1])
print ("\nFeatures : \n",dataset.columns.tolist())

Rows      : 7043
Columns   : 20

Features :
['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLine',
 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
 'TotalCharges', 'Churn']

```

Fig. 2 Row columns and features of dataset

3.1 Visualization of Dataset

See Figs. 3, 4, and 5.

4 Proposed Work

The proposed framework will use Python programming to create the churn prediction model.



Fig. 3 Gender distribution in customer attribution

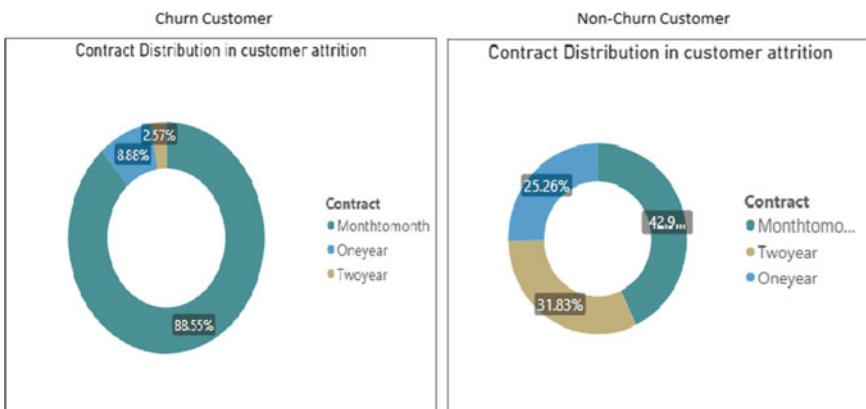
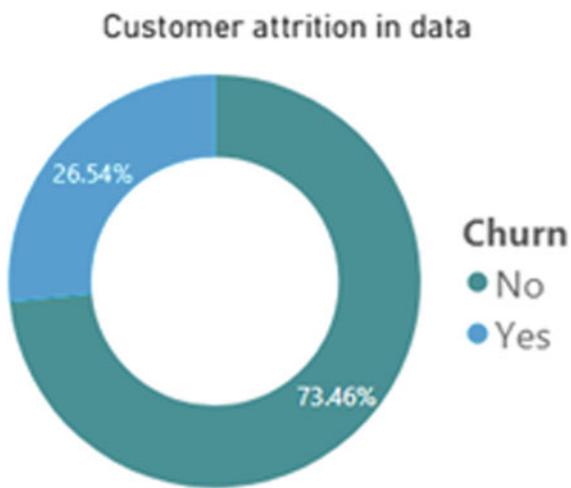


Fig. 4 Contract distribution in customer attribution

Fig. 5 Customer attribution in data



4.1 Data Collection

The telecom database is used for churn estimation, and estimation was made for the same.

4.2 Data Preparation

We must first clean the data for analysis and keep it ready so that the desired results can be obtained from it. A Churn Analysis has been applied to Telecommunications data in this paper; here the objective is to get information from the current service provider about potential customers who can churn from. The used database has 20 available variables. These concern gender, monthly fee, telephone service, etc. The database has access to more than 7000 customer-specific information (Fig. 6).

4.3 Prediction

The company is interested in the end product and it is very important that the outcome is represented in a “visual image” in such a way that it is identifiable and the outcome helps the organization to make the appropriate decisions that will in turn bring revenue (Fig. 7).

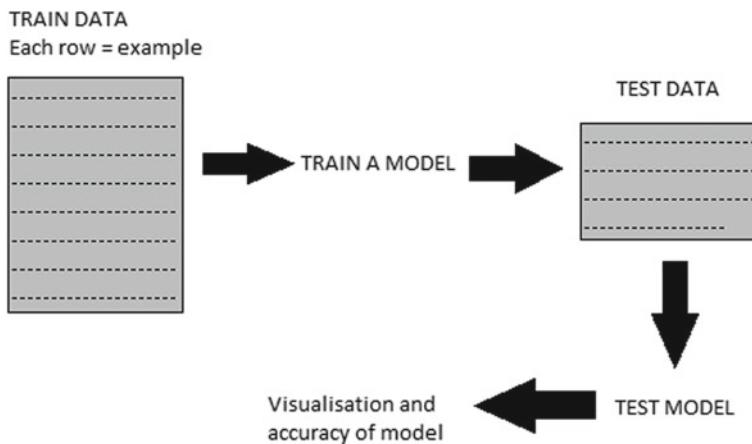


Fig. 6 Proposed system

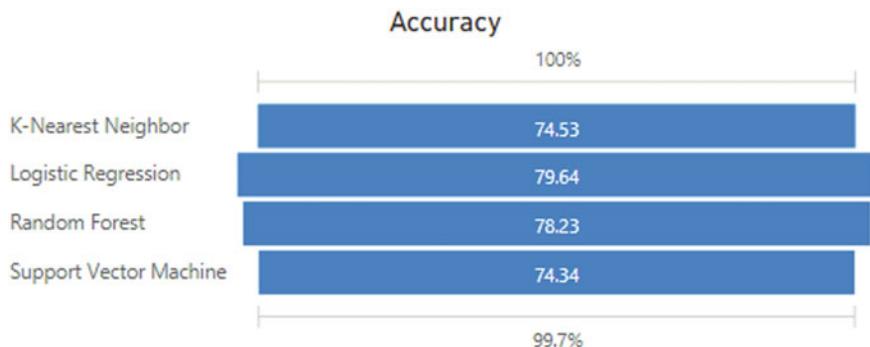


Fig. 7 Accuracy

4.4 Data Visualization Tools

Matplotlib Python library is used in this system which is a graph plotting library which was developed by John D. Hunter. It is the most popular data visualization module for python which helps in representing the information in an efficient and understandable way. Power BI tool is also used for clear visualization of dataset.

At the end, from the graph churn value is represented and concludes that those are the potential clients that the telecom service provider will be churning out (Fig. 8).

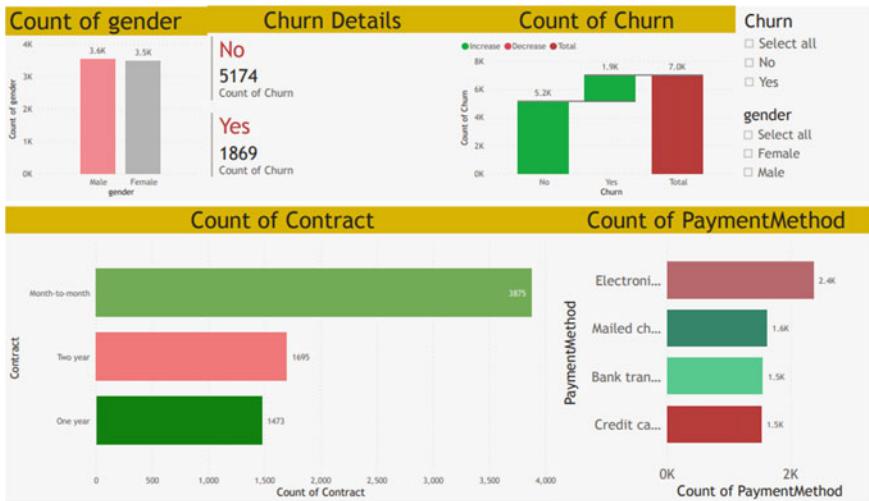


Fig. 8 Visualization of dataset

5 Architecture

A model churn prediction system consists of five phases:

- Collection of datasets and identifying the problem domain
- Extracting the features that are required for developing churn models
- Constructing the model using different classifiers and cross-validate the models
- Clustering of churning customers
- Prediction in the form of GUI and visualizing the result (Fig. 9).

In this model, different algorithms are used such as SVM, Decision Tree, Random Forest.

5.1 Support Vector Machine

SVM maps all of the rows of dataset to a higher dimensional plane to separate the data linearly. The plane dividing data is called hyperplane. To divides the dataset into two parts churn and the non-churn customer it plots the datapoint in n-dimensional plane and divides based on maximum marginal hyperplane.

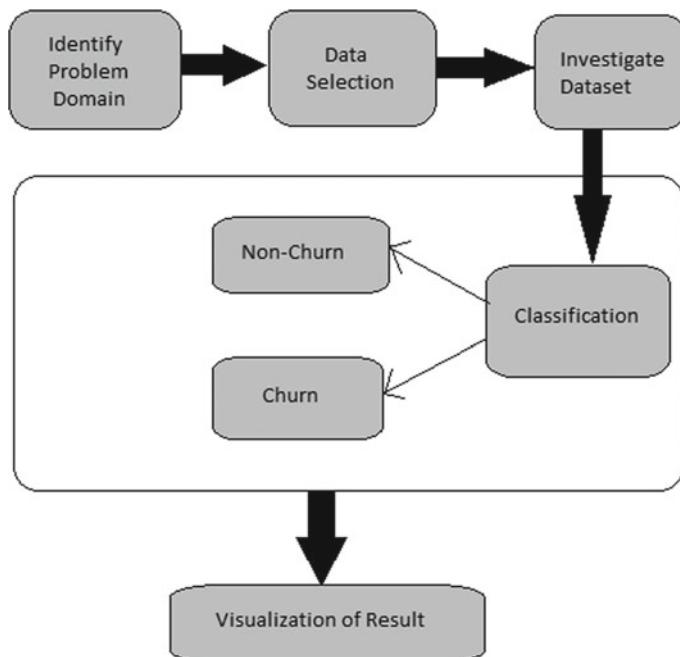


Fig. 9 Process flow

5.2 Random Forest

The theory has been confirmed by many studies that perhaps the random forest algorithm has a strong predictive accuracy with good openness for anomalous value and noise.

Random forests are used not only for classification but also for regression. It's mostly used in the classification issues though. As we know, trees form a forest and more trees make sense to say a more strong forest. Similarly, decision trees on data samples are developed by a random forest algorithm and one that gives each of them the prediction and selects the best alternative.

5.3 KNN

KNN is a very simple algorithm. The algorithm searches over the database when a specific customer is introduced for customers who are most identical to the target customer. This then estimates whether or not the client would churn depending on whether or not other similar customers have churned. KNN deals only with mathematical variables, so we'll exclude all non-numerical variables for this model.

5.4 *Logistic Regression*

Logistic regression is a statistical analysis that can be used when the dependent variable is dichotomous (binary). Dependent variable (here churn 1 or churn 0) is categorical and independent variables may be categorical or numerical.

6 Future Scope

- The proposed system can be used in banking and financial institution. Customers tend to move from one company to other company due to interest rate, fixed deposit rate and other services provided by the bank.
- The proposed system can be used in the Telecom industry. System helps to find the reason for why customer is leaving them and join to their competitors.
- The proposed system can be used in Online Gaming Industries.
- The proposed system can be used in Hospitality.

7 Conclusion

Till date, many churn prediction models are introduced. But companies require a simple and robust model to differentiate between non-churns and churns then clustering the resulting churners for providing retention solutions. In this prediction model, ML algorithms are introduced to help a CRM (Customer Relationship Management) department of various service-based companies to keep track of its customers and their behaviour against churn.

Initially, we solved the churn prediction problem by applying machine learning algorithms to overcome the issue of customer churn. Previous works are thoroughly studied and summarized the present customer churn prediction model. Unlike other existing systems, which primarily focused only on the prediction models and the accuracy of churn prediction, this system presents the characteristics of the existing publicly available churn prediction datasets as well as visualized them. Further, we focused on different variables related to a customer that are used for churn prediction and categorized them. Lastly, we surveyed the list of commonly used metrics proposed in the previous system to assess the efficiency of different churn predictive models. Also, we visualized the end result for better understanding using python programming.

References

1. C. Wang, R. Li, P. Wang, Z. Chen, Partition cost sensitive CART based on customer value for telecom customer churn prediction, in *2017 36th Chinese Control Conference (CCC)*, September 2017
2. N. Lu, H. Lin, J. Lu, G. Zhang, A customer churn prediction in telecom industry using boosting. Customer behaviour in telecommunications. *IEEE Trans. Ind. Inform.* **10**(2) (2014)
3. P. Li, S. Li, T. Bi, Y. Liu, Telecom customer churn prediction method based on cluster stratified sampling logistic regression. *IEEE* (2014)
4. A. Idris, A. Khan, Ensemble based Efficient churn prediction model for telecom, in *International Conference on Frontiers of Information Technology (FIT)* (2015), pp. 5680–5684
5. G. Xia, H. Wang, Y. Jiang, Application of customer churn prediction based on weighted selective ensembles, in *International Conference on Systems and Informatics (ICSAI 2016)*, November 2016, pp. 513–519
6. M. Rohini, P. Devaki, Analysis of customer churn by big data clustering. *Int. J. Innovative Res. Computer Commun. Eng.* **5**(3) (2017)
7. N. Saini, Churn prediction in telecommunication industry using decision tree. *Streamed Info. Ocean.* **1**(1) (2016)
8. A.A.Q. Ahmed, D. Maheswari, Churn prediction on huge telecom data using hybrid firefly based classification Churn prediction on huge telecom data. *Egypt. Inform. J.* **18**(3), 215–220 (2017)
9. M. Akmal, *Factors Causing Customer Churn: A Qualitative Explanation of Customer Churns In Pakistan Telecom Industry* (2017)
10. K. Dahiya, S. Bhatia, *Customer Churn Analysis in Telecom Industry* (IEEE, 2015). 978-1-4673-7231-2/15
11. Praveen et al., Churn prediction in telecom industry using R. *IJETR.* **3**(5) (2015). ISSN 2321-0869

Frequency Detection and Variation with Smart-Sensor Data Analysis Using Artificial Neural Network and Cloud Computing



Arabinda Rath, Dillip K. Mishra, S. Q. Baig, and Gayatri Devi

Abstract When an electrical machine operates with load current that does not exceed its current rating, the temperature rise of any part will never exceed the permissible limits, thus providing continuous and reliable operation. IoT-based protection system is proposed for the identification of any runtime current/voltage conditions that leads to breakdown. IoT implementation with sensor data analysis in using artificial neural networks and cloud computing is presented in this paper. Here, we use LoRa network to transmit the hall effect sensor with raw data that measures current to another node which connects with the system. In the system, we use C# program to process the raw data and extract meaning from it using artificial neural networks that learn to filter the raw data. The data is then sent to the cloud server. The simulation shows that we have to maintain a threshold condition to improve the quality of the electrical machine.

Keywords Hall effect sensor · C# · Arduino · Fluctuating-Frequency · LoRa · ANN · Cloud server

A. Rath · D. K. Mishra · S. Q. Baig · G. Devi (✉)

Department of Computer Science and Engineering, ABIT, Cuttack, Odisha, India
e-mail: gayatridevi@abit.edu.in

A. Rath
e-mail: Aurobinda.ratha@rediffmail.com

D. K. Mishra
e-mail: mishra.dillip88@gmail.com

S. Q. Baig
e-mail: baig.sarosh@gmail.com

1 Introduction

Increasing reliability of the production process is inherently tied to the ability of controlling breakdown time of machines and moving from a scenario of breakdown maintenance to preventive maintenance. Preventive maintenance can be best executed by having predictive methods that are based on previous data of operating conditions that lead to machine failure. More specifically, the predictive mechanism involves identifying breakdown possibilities ahead of time by correlating with machine characteristics such as current flow, vibrations, thermal signature, etc.

In the first part of this paper, we consider the current flow consumption of electrical machine for identifying breakdown possibility through the hall effect sensor. Second, with widespread adoption of IoT, such thought to be used in a scenario of multiple machines used together in a production process to improve reliability and reduce damaged goods in the entire production process. Hall effect sensor will take the reading of current passing through a device and the raw data is then transmitted through LoRa Gateway to a system containing C# application. In the third part, Artificial Neural Network program is applied to filter the raw data. When the device starts consuming current abnormally then ANN program can detect those changes. The program differentiates normal fluctuations from abnormal fluctuations.

The ANN having hidden layer of two nodes is created in the application:

1. Calibration factor
2. Fluctuation time when the power changes.

From the experimental results, the novelty of the technique indicates that the proposed method achieves detection of high energy consumption and maximizes the accuracy in detecting fault of machine in an early stage.

2 Methods

In the commercial production industries, the subject of machine condition with current consumption monitoring and fault diagnosis has gained a lot of interest due to reduction of maintenance budgets. Sensor data analysis through ANN using IoT smart technology is a successful method of machine condition monitoring and fault diagnosis. Several researchers had developed smart technology using IoT [7–9].

In 2019, Xu et al. had proposed a fault diagnosis method based on deep convolutional neural network [4]. In 2017, Zhang et al. established the fault diagnosis model on deep neural networks [3]. However, the effectiveness of these methods have some difficulties in learning representative features from the raw data. In 2014, the low displacement or high-frequency mechanical waves are transformed to electronic signals. The signal strength can be increased by using preamplifier before the acoustic emission data analysis [1, 2]. In 2012, Aguiar et al. mentioned digital signal processing for acoustic emission using ANN [6]. In 2018, Ali introduced AI

application in machine condition monitoring and fault diagnosis [5]. However, most existing methods still have difficulties in learning features from the raw data.

2.1 Artificial Neural Network

ANN is an information processing approach. In this architecture, the number of nodes in hidden layers, initial weight assignments, and activation functions played a key role. When designing a neural network, there are a number of different parameters that must be decided. Some of these are the number of neurons per layer and the transfer functions and so on.

This technique was found to be very useful as it can be used in the industrial health diagnosis of machines. In 2018, Yasir Hasan Ali's work is mentioned in this paper because it contains AI application in Machine Condition Monitoring but still need more encouragement using smart technology for raw data analysis from sensor.

2.2 C# Application with ANN

Artificial neural network having hidden layer of 2 nodes is created in the application and is shown in Fig. 1. The two nodes of the hidden layer are

- Calibration factor (value)
- Fluctuation time when the power changes (time).

Calibration Factor (value) This is the ratio of response from the detector to the analyte concentration.

The calibration factor, K , relates the incident power to the DC power substituted by self-balancing bridge, It is defined as a ratio of DC substituted power, P^{DC} , to total incident RF power P^{RF}

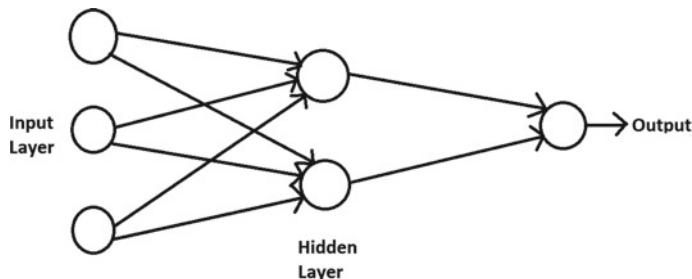


Fig. 1 ANN with two nodes in hidden layer

$$K = P^{\text{DC}}/P^{\text{RF}} = (1 - |\Gamma|)\eta$$

Γ is the input coefficient and η is the mount effective efficiency. Here in our paper, it is used to calculate the response values of pick area to **long spike** and the down area to **dwarf spike** (with and without load).

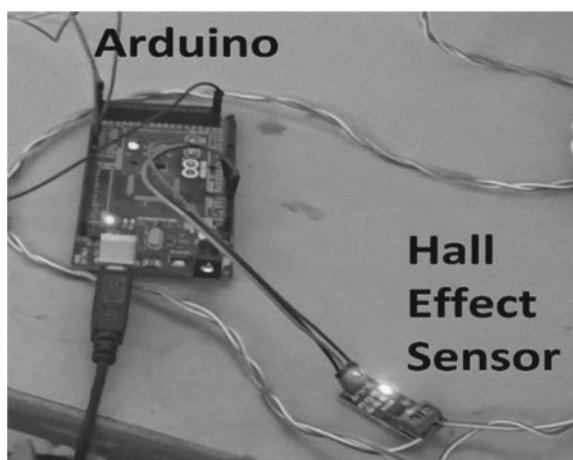
Artificial neurons try to imitate the biological neurons of the human brain. The neurons interact with each other via some links that connect them. The nodes take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs learn by altering their weight values.

If the ANN generates the desired output then there is no need to adjust the weights. However, if the network generates undesired output or an error is generated, then the system needs to make changes in the weights in order to improve the output. The application tries to extract meaning from the raw data. It eliminates the minor calibration error and the sensor-related fluctuations. It performs all the calculations to change the weight of the neuron that determines there is an actual fluctuation in power or not.

2.3 Hall Effect Sensor Data Input with IoT Implementation

We are using ACS712 Hall effect sensor module to measure the current consumption of a machine (example: a fan). The sensor data is collected through Arduino and it transmits the sensor data to the node connected to the application through LoRa model (Dorgi DRF 1276 DM) at frequency 433 MHz. Receiving node interacts with the C# application. The data is extracted from the node in C# application which is shown in Fig. 2.

Fig. 2 Arduino connected to hall effect sensor (In our research LAB)



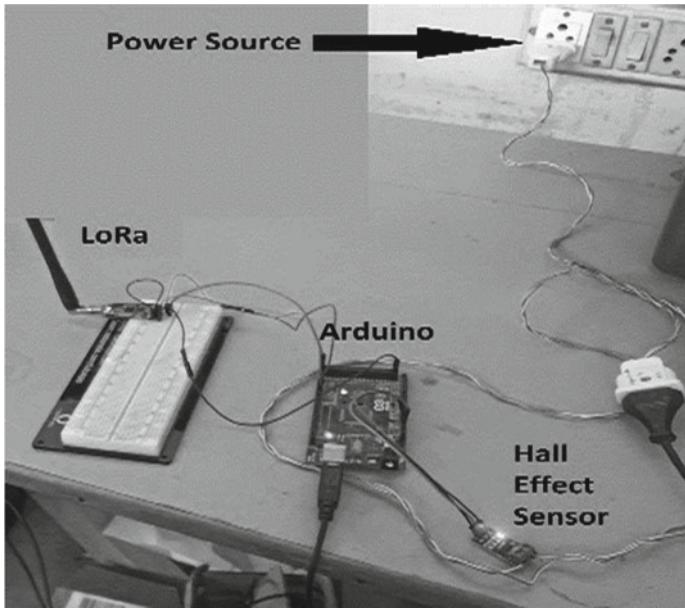


Fig. 3 LoRa transmits the sensor data

2.4 *Transmission of Data Through Lora*

Arduino transmits the sensor data to the node that is connected with the application through LoRa Module (DorgiDRF1276DM) at Frequency: 433 MHz and is shown in Fig. 3. LoRa offers radio coverage to a large area with very low energy consumption in the end devices [10].

2.5 *Cloud to Application Interaction*

In Fig. 4, C# Application interacts with the cloud server in real-time and the processed data can be viewed on a website. A web server is set up to store and host the data. The server is linked and authenticated in the system we have the C# application so that the application can interact with the website directly. The C# application can manipulate the php files of the websites directly or it may alter a database that is accessed by the website.

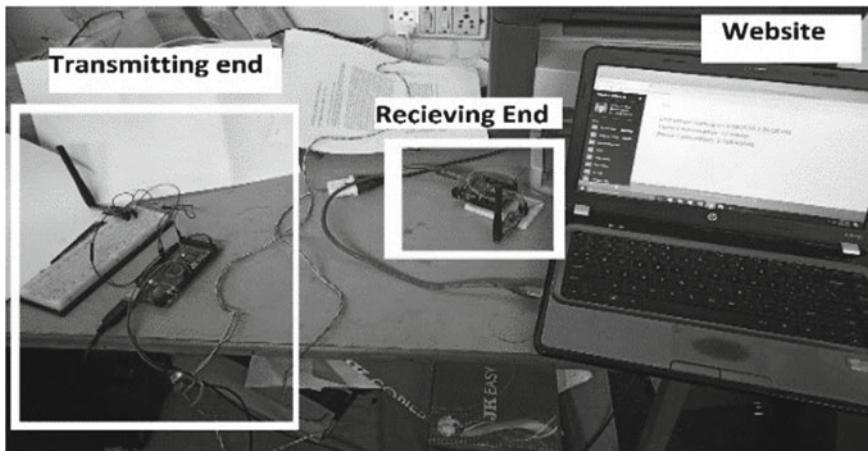


Fig. 4 Two ends of LoRa and created website

3 Data Set and Simulation Result

3.1 Data Set

See Fig. 5.

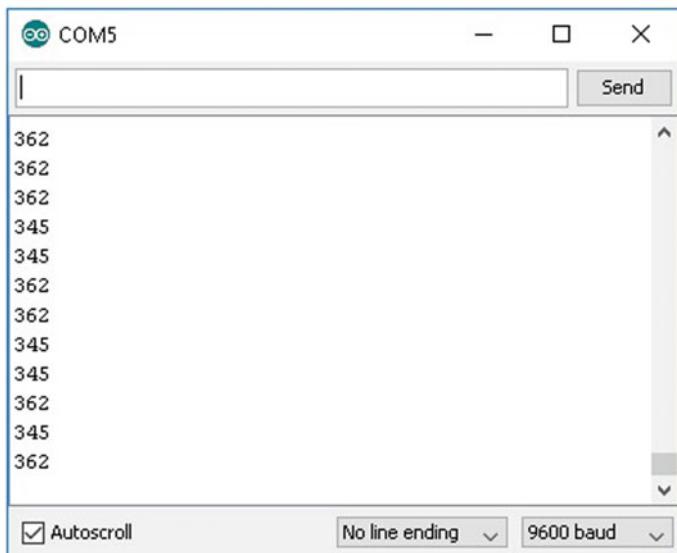


Fig. 5 Raw sensor data received by Arduino connected to the LoRa module

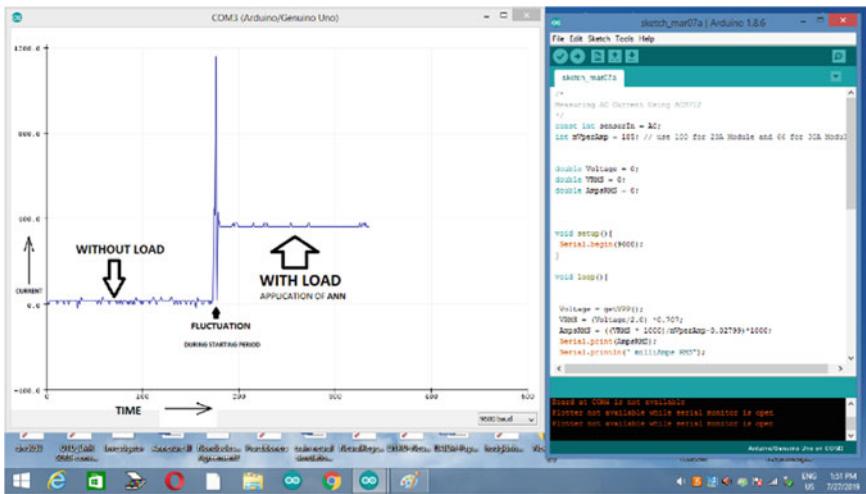


Fig. 6 Simulation result

3.2 Simulation

Here in Fig. 6 (from simulation result) **with load** shows that the microcontroller is connected with the load. When there is no load the dwarf spikes are visible and long spikes are visible if load is connected. From **without load** to **with load** there is a fluctuation seen in the simulation graph. After fluctuation, there is a continuous sequence of dwarf spicks which shows that the machine is working in a flexible manner.

The website shows the result of all the processes executed on different pages.

Figure 7 shows the Date and time of the last sensor reading and the extracted meaning from the raw data which is current consumption and power consumption. This result is driven through the simulation graph.

The manipulated data page in Fig. 8 shows the extracted meaning from the raw data as well as the fluctuation of current in seconds. The program differentiates the normal fluctuation from abnormal fluctuation and the filtered value is then passed to the website to display.

Figure 9 shows all the filtered meaningful data that is recorded in the cloud server. The meaningful data recorded are the spikes in current, abnormal fluctuations in current, and abnormal change in current consumption.

Apart from these, there are pages that display us the fluctuations, all the recorded sensor raw data, etc. there is also an alert page that contains the information that needs urgent notice like heavy fluctuation or heavy diversion from normal current consumption.

Fig. 7 Sensor raw data

| | | | | | | | | | | | | | | | | |
|--|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|
|  <p>Sarosh Baig Administrator : 8093135350</p> | <p>Manipulated Data:</p> <p>Last sensor reading on: 4/24/2019 2:09:10 PM</p> <p>new Base Value: 345</p> <p>Fluctuations for seconds: 0</p> <p>Current consumption: 345 mAmp Power Consumption: 107.3226 Watt</p> <p>Original Data:</p> <table><tbody><tr><td>1: 345</td></tr><tr><td>2: 345</td></tr><tr><td>3: 345</td></tr><tr><td>4: 345</td></tr><tr><td>5: 345</td></tr><tr><td>6: 345</td></tr><tr><td>7: 345</td></tr><tr><td>8: 345</td></tr><tr><td>9: 345</td></tr><tr><td>10: 345</td></tr><tr><td>11: 345</td></tr><tr><td>12: 345</td></tr><tr><td>13: 345</td></tr><tr><td>14: 345</td></tr><tr><td>15: 345</td></tr></tbody></table> | 1: 345 | 2: 345 | 3: 345 | 4: 345 | 5: 345 | 6: 345 | 7: 345 | 8: 345 | 9: 345 | 10: 345 | 11: 345 | 12: 345 | 13: 345 | 14: 345 | 15: 345 |
| 1: 345 | | | | | | | | | | | | | | | | |
| 2: 345 | | | | | | | | | | | | | | | | |
| 3: 345 | | | | | | | | | | | | | | | | |
| 4: 345 | | | | | | | | | | | | | | | | |
| 5: 345 | | | | | | | | | | | | | | | | |
| 6: 345 | | | | | | | | | | | | | | | | |
| 7: 345 | | | | | | | | | | | | | | | | |
| 8: 345 | | | | | | | | | | | | | | | | |
| 9: 345 | | | | | | | | | | | | | | | | |
| 10: 345 | | | | | | | | | | | | | | | | |
| 11: 345 | | | | | | | | | | | | | | | | |
| 12: 345 | | | | | | | | | | | | | | | | |
| 13: 345 | | | | | | | | | | | | | | | | |
| 14: 345 | | | | | | | | | | | | | | | | |
| 15: 345 | | | | | | | | | | | | | | | | |
| <p>Data</p> <p> Sensor data Raw Data</p> <p> Program Status status</p> <p> Manipulated Data</p> <p> Alert</p> <p> Data History</p> <p> Fluctuations</p> <p> All Data</p> <p> Detailed Data</p> | | | | | | | | | | | | | | | | |

Fig. 8 Manipulated data after ANN

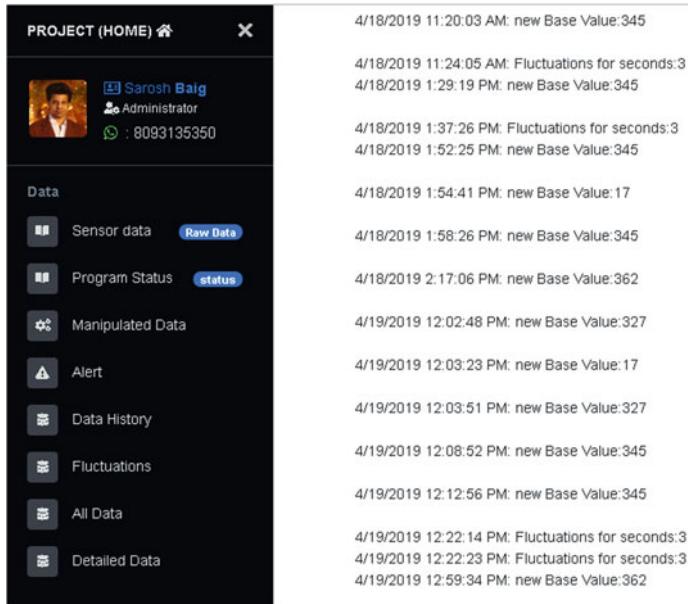


Fig. 9 Filtered meaningful data (Stored at cloud server)

4 Conclusion and Future Work

When the devices start consuming current abnormally, then our ANN program can detect those changes. The program differentiates normal fluctuations from abnormal fluctuations. Hence any damage or malfunction in the device can be detected in the early stage. Apart from this, IoT is implemented using LoRa gateway and cloud server. The result can be viewed on a website that is created to display real-time data on the server. We are expecting that the future scope work in this is to implement multi-level neural network for complexity.

References

- Y.H. Ali, M.H Omar, R.A Rahman, Acoustic emission technique in condition monitoring and fault diagnosis of gears and bearings. *Int. J. Acad. Res.* **6**(5) 2014
- Y.H. Ali, R.A. Rahmen, R. Hamzah, Acoustic emission signal analysis and artificial intelligence techniques in machine condition monitoring and fault diagnosis. *Rev. J. Tecnol.* **69**(2) (2014)
- R. Zhang, Z. Peng, L. Wu, B. Yao, Y. Guan, Fault diagnosis from raw sensor data using deep neural networks considering temporal coherence. *Sensors* **17**(549) (2017)
- G. Xu, M. Liu, Z. Jiang, D. Soffker, W. Shen, Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. *Sensors* **19**(5) (2019)
- Y.H. Ali, *Artificial Intelligence Application in Machine Condition Monitoring and Fault Diagnosis* (Intech Open, 2018), p. 74932, Chapter 14

6. P.R. Aguiar, C.H. Martins, M. Marchiand, E.C. Bianchi, *Digital Signal Processing for Acoustic Emission* (INTECH, 2012), Chapter 12
7. F. Sheikh, X. Li, Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications. *Procedia Comput. Sci.* **34**, 103–110 (2014)
8. A. Zaslavsky, C. Perera, D. Georgakopoulos, Sensing as a service and big data, in *Proceedings of the International Conference on Advances in Cloud Computing (ACC)*, Bangalore, India, July 2012
9. G. Jayawardhana, R. Buuyaab, S. Marusic, M. Palaniswami, Internet of things (IoT): a vision, architectural elements and future directions. *Future Gener. Comput. Syst.* **29**(7), 1645–1660 (2013)
10. A. Augustin, J. Yi, T. Clausen, W.M. Townsley, A study of LoRa: long range and low power networks for the internet of things. *Sensors* (2016)

Comprehensive Study of Fetal Monitoring Methods for Detection of Fetal Compromise



Vidya Sujit Kurtadikar and Himangi Milind Pande

Abstract Fetal monitoring usually refers to monitoring fetal heart rate (FHR) for detection fetal well-being. This is important activity carried out by doctors during prepartum, intrapartum phase as per health requirement of patient. Fetal monitoring is required to reduce chances of fetal to become hypoxic (when fetal is deprived from sufficient oxygen) that can cause fetal brain injury and even fetal death. Fetal monitoring plays important role in reducing mortality and morbidity rate. The most common noninvasive fetal monitoring device is cardiotocograph (CTG). CTG captures FHR based on Doppler ultrasound principle and uterine contractions (UC) based on pressure transducers. The present study highlights accuracy limitations of CTG and proposes more accurate noninvasive fetal ECG (NIfECG) as data acquisition methods to acquire FHR and electrohysterogram (EHG) to capture UC. CTG interpretation is one of the decision-making parameters used by doctors for early intervention like caesarian section. CTG interpretation often suffers from inter-observer and intra-observer agreement of CTG patterns which are non-reassuring. To overcome this limitation, computerized analysis can be useful. The present study also discusses usage of machine learning to detect fetal compromise. Further, in addition to FHR and UC analysis, we also propose to use ST waveform analysis to improve results.

Keywords Fetal monitoring · FHR · CTG · NIfECG · EHG · Machine learning · Fetal compromise · ST analysis

V. S. Kurtadikar () · H. M. Pande
School of Computer Engineering and Technology, MITWPU, Pune, India
e-mail: vidya.kurtadikar@mitwpu.edu.in

H. M. Pande
e-mail: himangi.pande@mitwpu.edu.in

1 Introduction

According to WHO health statistics 2018, 2.5 million neonatal deaths occurred in 2018 globally. Despite of substantial progress in child survival since 1990, the decline in mortality rate for neonatal from 1990 to 2018 has been slower than that of post-neonatal mortality. As compared to high income country, there are 10 times more chances of a child born in sub-Saharan Africa or in Southern Asia to die in the first month. It has been observed that, among all neonatal deaths, 75% deaths occur during the first week of life, and within first 24 h, about 1 million newborns die. Major causes of these deaths are preterm birth, intrapartum-related complications (such as birth asphyxia) [1]. Some measures like high quality antenatal care with good coverage, availability of skilled staff during birth, proper care for mother and baby after birth, and special care of low weight and sick newborns will help to improve survival of mother and newborns which will reduce stillbirths ratio [2].

1.1 *Fetal Monitoring*

Fetal monitoring is commonly used during prepartum and antepartum phase to study relationship between Fetal Heart Rate (FHR) and fetal health to prevent fetal compromise. Various devices are used to monitor and understand the baby's heartbeat pattern. The interpretation of fetal heart patterns during labor is major factor that helps doctors to decide whether or not to intervene in the birth process.

The remainder of this paper is organized as Sect. 2 describing related work, Sect. 3 explains the proposed methodology, and Sect. 4 concludes paper.

2 Related Work

This section compares commonly used devices used in fetal monitoring considering various parameters including accuracy. We have also compared previous literature based on various machine learning algorithms used.

2.1 *Comparison Between Common Fetal Monitoring Devices*

Table 1 summarizes commonly used fetal monitoring devices based on its purpose, advantages and disadvantages.

Table 1 Comparison of fetal monitoring devices

| S. No. | Device name | Invasive/NonInvasive | Purpose | Advantages | Disadvantages |
|--------|---|----------------------|--|--|---|
| 1 | Intrauterine pressure catheter (IUPC) | Invasive | Used to measure strength of uterine contractions. Placed into the amniotic space | <ul style="list-style-type: none"> It can measure strength of uterine contractions More precise and accurate as compared to noninvasive method [4] | <ul style="list-style-type: none"> Requires ruptured membranes and dilated cervix Small risk of infections |
| 2 | Fetal scalp electrode (FSE) | Invasive | Used to monitor fetus electrocardiogram during labor | <ul style="list-style-type: none"> More precise and accurate as compared to noninvasive method [3] | <ul style="list-style-type: none"> Requires ruptured membranes and dilated cervix Small risk of infections |
| 3 | Cardiotocography (CTG) | Noninvasive | Multipurpose device to monitor: <ul style="list-style-type: none"> FHR UC and its frequency Based on doppler ultrasound | <ul style="list-style-type: none"> Noninvasive method Can be used antepartum, intrapartum | <ul style="list-style-type: none"> Less sensitive and less accurate as compared to invasive methods [3] Many times, reposition required |
| 4 | Abdominal electrocardio gram (aECG) OR NIHECG | Noninvasive | Multipurpose device to monitor: <ul style="list-style-type: none"> FHR Fetal ECG Uterine contractions Based on electrolystrogram (EHG) | <ul style="list-style-type: none"> Noninvasive method Can be used antepartum, intrapartum More reliable and accurate as compared to CTG [3] | <ul style="list-style-type: none"> Less sensitive and less accurate as compared to invasive methods [3] Need of separation of mECG and fECG |

2.2 Study of CTG and Electrocardiogram (ECG) Based Devices for FHR and Uterine Contractions (UC) Accuracy

Accuracy of FHR and UC affects decision making during labor. Considering scalp Electrode as gold standard for FHR measurement, it has been found that Noninvasive Fetal ECG is more accurate as compared to CTG which is based on Doppler method. Accuracy is maintained even if body mass index (BMI) is more [3].

For uterine activity measurement, considering IUPC as gold standard, Electro hysteroscopy (EHG) which capture uterus electrical activity using electrodes placed on the maternal abdomen is found to be more accurate as compared to CTG by tocodynamometry based on toco (strain gauge technology) which measures UC frequency and approximate duration. Furthermore, signal dropouts is also one of the limitation of CTG. Accuracy is maintained in EHG regardless of body habitus [4]. CTG devices accuracy is not sufficient for reliable quantitative evaluation of short term FHR variability [5].

Without considering Direct Fetal ECG signal it has been proved that, abdominal fetal electrocardiography provides more reliable description of instantaneous FHR variability, as compared to ultrasound method. Moreover, additional advantage of abdominal fetal electrocardiography is lower signal loss as compared to ultrasound method [6]. Fetal Lite is a fetal monitoring device developed by an Indian company Sattva MedTech based on NIffECG and EHG is compared with state of the art CTG devices GE Coro-metrics 170 series. Fetal Lite was found to be accurate [7].

Major concern while using noninvasive electrocardiogram-based devices is to separate fetal ECG (fECG) from maternal ECG (mECG). An approach is presented for extracting fECG noninvasively, using a single electrode. Physionet dataset was used and validated using gold standard which is scalp fetal ECG [8].

2.3 Study of CTG Only and CTG and ST Waveform Analysis Methods

Cardiotocography alone is inadequate for fetal distress detection [9]. In a normal CTG pattern reassures well-being of fetus, while an abnormal pattern does not ensure that fetal is hypoxic. CTG often has high sensitivity at the same time low specificity. Adjuncts methods are required to increase the sensitivity. One of adjunct method is fetal ECG component ST waveform analysis or STAN [10]. In fetal ECG analysis, ST segment elevation and T wave and QRS amplitudes ratio (T/QRS), may identify anaerobic myocardial metabolism caused by insufficient supply of oxygen to fetal. Intrapartum phase use of fetal ECG with cardiotocography (CTG) labor along with strategies and references concerning CTG and ST classification and interpretation are mentioned [11]. STAN methodology and derivation of ratio (T/QRS) is explained [12].

Previous analysis of CTG and STAN meta-analyses contained errors with a conclusion that meta-analysis did not contain whole and related data from all Randomized Controlled Trials (RCT) [13]. Combined approach of CTG and STAN did not reduce total caesarian section deliveries, but there was substantial decrease in operative vaginal deliveries. There is noteworthy decrease in total blood sampling for confirmation of fetal acidosis in the STAN group [14]. In term high-risk deliveries in which there is a need for internal fetal monitoring, statistical analysis proved that combined approach of CTG and STAN is cost effective [15]. Usage of CTG STAN increased from 26 to 69%. There is substantial reduction of cord metabolic acidosis rate from 0.72 to 0.06% [16]. Considering largest RCT from the United States (US) there is significant reduction of metabolic acidosis rates by 36% and reduction in operative vaginal delivery rates by 8% [17].

2.4 *Study of CTG Interpretation Methods*

This section reviews CTG interpretation methods.

Initial system describes proper baseline derivation which is used to measure accelerations or decelerations [18] and describes a program that recognizes accelerations and decelerations from automated baseline [19]. Very preliminary stage computerized system that provides a numerical description reactive fetal as per fetal gestational age and which is independent of presence of accelerations [20]. Hidden Markov Models were used to classify FHR patterns into hypoxic and normal newborns [21]. A model based on supervised artificial neural network (ANN) was developed for CTG data classification system into Normal, Suspicious and Pathologic condition with very good accuracy [22]. Different machine learning algorithms were used to fetal into normal, suspicious and pathological class. Result showed Decision Tree is the best algorithm on RStudio, as compared Support Vector Machine (SVM) and Naïve Bayes [23]. SVM and Random Forests resulted in 96% accuracy for prediction of fetal outcome, SVM performance was somewhat better for suspect cases [24]. Among different techniques like SVM, ANN, Radial Basis Function Network, Extreme Learning Machine, and Random Forest, according to experimental results, ANN was found to be best of all [25]. A CTG Open Access Software (CTG-OAS) which used CTU-UHB dataset, and which characterizes FHR patterns considering various features. CTG-OAS also has tools preprocessing, feature selection, feature extraction, classification [26]. SVM based classifier was used with Genetic Algorithm (GA) for selection of optimal feature subset to maximize performance of classification. Performance is tested on extensive clinical CTG data, which is categorized by experts. SVM with GA proved to be better when compared to Adaptive Network-based Fuzzy Inference System (ANFIS) and ANN [27]. Different Feature Selection (FS) models were used and evaluated to determine influence on naïve Bayes performance while classifying fetal states. ReliefF yielded better performance for fetal state classification [28]. Adaptive boosting is used with decision trees and different ML techniques were used for fetal classification [29]. Considering perinatal database

consisting of normal and pathological classes, SVM classifiers used for classification. Detection of 50% of pathological cases was done with a little false positive rate. This detection was done well in advance needing clinical intervention [30].

Binary decision tree and least squares support vector machine (LS-SVM) used for fetal state classification. Particle swarm was used as optimization method. Experimental results demonstrated that a good classification accuracy rate [31]. GA was used as a feature selection method. Classification performance is good, when different features were integrated [32]. Automated method for FHR diagnostic analysis during labor that uses ANN was used introduced [33]. To overcome limitations of SMOTE in case of nonlinear complications, Weighted Kernel based SMOTE (WK-SMOTE) was proposed which when used with SVM resulted in performance improvement as compared to rest of baseline methods which stands as benchmarking on imbalanced datasets. Multiclass imbalanced issues were addressed by developing hierarchical framework [34].

Sparse features subset selection was done automatically which resulted in reasonable classification [35]. This paper present proof-of-concept demonstrating use of machine learning in determining need cesarean section objectively that will be helpful in avoiding antenatal deaths [36]. Inter-observer and intra-observer agreement quantification on non-reassuring CTG patterns was done, resulting into conclusion that poor inter-observer and intra-observer agreement of CTG classification [37]. A computer-aided diagnosis (CAD) scheme combined Deep Learning (DL) method was proposed. Recurrent plot was used to transform preprocessed one-dimensional FHR signal into two-dimensional image to capture the nonlinear characteristics. After experimentations on CNN optimization, better results were obtained. There was no need of complex feature engineering [38].

Two issues were addressed: 1. Equal Class Distribution; which was resolved by using window approach by splitting CTG time series signals and 2. Automatic Feature Extraction using a one-dimensional CNN (1DCNN) and MLP ensemble. Training and evaluation of 1DCNN-MLP models was done with several windowing strategies which showed better results with size of window was 200. Comparison with a SVM, a Random Forest (RF) and a Fishers Linear Discriminant Analysis (FLDA) classifier proved that 1DCNN strategy is best among rest of methods used [39]. Multimodal CNN (MCNN) and Stacked MCNN models were used as classification methods. These methods were used to analyze three CTG datasets: Oxford archive, SPaM 17, CTU-UHB. Signal quality effect on MCNN performance was also accessed. Two external datasets were used for comparison; MCNN showed better results compared to existing feature extraction-based techniques and claimed to be initial attempt to evaluate the CTG by assessing likelihood of fetal compromise at specific point, considering likelihood estimates at an earlier point [40].

3 Proposed Methodology

After conducting comprehensive Literature Review in 2.1, 2.2, 2.3 and 2.4 following observations can be drawn:

- Fetal monitoring using CTG is less accurate and less reliable as compared to NIECG method.
- Fetal monitoring using CTG also suffers from signal loss as compared to NIECG method.
- Fetal monitoring using CTG alone is not sufficient. There must be adjunct method like Fetal Blood Sampling method or STAN.
- CTG manual interpretation suffers from Inter-observer and Intra-observer variability.

To address these issues, we are proposing to adopt following methodology in our future study:

- Consider NIIfECG as data acquisition method for FHR and Uterine activity.
- Consider ST analysis in adjunct with CTG/NIIfECG.
- Use of AI/Machine learning methods for interpretation of fetal monitoring patterns that will support in decision making to Gynecologist/Obstetrics/Doctor.

Refer Fig. 1 for proposed block diagram.

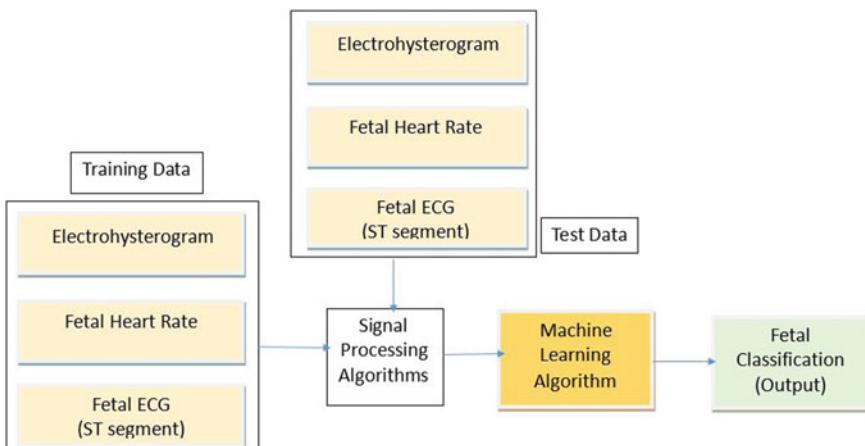


Fig. 1 Proposed block diagram

4 Conclusion

We conducted a comprehensive study of fetal monitoring devices for FHR and UC accuracy and other parameters. We propose to use NIIfECG as data acquisition method for FHR, ST segment and EHG for UC. CTG alone is inadequate for fetal distress detection as it often suffers from high sensitivity at the same time low specificity. Adjuncts methods such as STAN can be used to increase sensitivity. These two fetal monitoring approaches one that uses only CTG and second combined approach that uses CTG and STAN (as an adjunct method) are discussed. CTG manual interpretation suffers from Inter-observer and Intra-observer variability. Computerized analysis and AI and Machine Learning based methods can be used to overcome this limitation. We propose to use Machine learning for detection of fetal compromise with adjunct to ST segment analysis.

References

1. World Health Statistics (2018). <https://apps.who.int/iris/bitstream/handle/10665/272596/9789241565585-eng.pdf>. Last accessed 2020/02/09
2. J.E. Lawn, H. Blencowe, P. Waiswa, A. Amouzou, C. Mathers, D. Hogan, V. Flenady, J.F. Frøen, Z.U. Qureshi, C. Calderwood, S. Shiekh, F.B. Jassir, D. You, E.M. McClure, M. Mathai, S. Cousens, For the lancet ending preventable stillbirths series study group* with the lancet stillbirth epidemiology investigator group, ending preventable stillbirths 2 stillbirths: rates, risk factors, and acceleration towards 2030. *Lancet* **387**, 587–603 (2016)
3. T.Y. Euliano, S. Darmanjian, M.T. Nguyen, J.D. Busowski, N. Euliano, A.R. Gregg, Monitoring fetal heart rate during labor: a comparison of three methods. *Hindawi J. Pregnancy* **2017**, 5 (2017). Article ID 8529816
4. T.Y. Euliano, M.T. Nguyen, S. Darmanjian et al., Monitoring uterine activity during labor: a comparison of 3 methods. *Am. J. Obstet Gynecol.* **208**:66, e1–e6 (2013)
5. J. Jezewski, J. Wróbel, K. Horoba, Comparison of doppler ultrasound and direct electrocardiography acquisition techniques for quantification of fetal heart rate variability. *IEEE Trans. Biomed. Eng.* **53**(5) (2006)
6. J. Jezewski, J. Wróbel, A. Matonia, K. Horoba, R. Martinek, T. Kupka, M. Jezewski, Is abdominal fetal electrocardiography an alternative to doppler ultrasound for FHR variability evaluation? *Front. Physiol.* **8**, 305 (2017). <https://doi.org/10.3389/fphys.2017.00305>
7. B.K. Subramanian, S. Kaulgud, V.R. Joshi, G. Godbole, Comparative study to determine the reliability and accuracy of the fetal lite electronic fetal monitor when compared with conventional cardiotocography. 671–676 (2018). <https://doi.org/10.1109/comsnets.2018.8328293>
8. R. Mujumdar, P. Nadar, A. Bondre, A. Kulkarni, S. Pathak, Principal component analysis (PCA) based single-channel, non-invasive fetal ECG extraction (2019). <https://fetosense.com/assets/publications/PCA.pdf>
9. S. Parveen, Umbilical cord arterial blood base excess as gold standard for fetal wellbeing screening test validity at term delivery. *J. Pak. Med. Assoc.* **60**(5), 347–350 (2010)
10. A. Sacco, J. Muglu, R. Navaratnarajah, M. Hogg, ST analysis for intrapartum fetal monitoring. *Obstetrician Gynaecologist* **17**, 5–12 (2015)
11. I. Amer-Wahlin, S. Arulkumaran, H. Hagberg, K. Maršál, G.H.A. Vissere, Fetal electrocardiogram: ST waveform analysis in intrapartum surveillance. *BJOG* **114**, 1191–1193 (2007)

12. L.D. Devoe, Fetal ECG analysis for intrapartum electronic fetal monitoring: a review. *Clin. Obstet. Gynecol.* **54**(1), 56–65 (2011)
13. P. Olofsson, D. Ayres-de-Campos, J. Kessler, B. Tendal, B.M. Yli, L. Devoe, A critical appraisal of the evidence for using cardiotocography plus ECG ST interval analysis for fetal surveillance in labor. Part II: the metaanalyses. *Acta Obstet. Gynecol. Scand.* **93**(6), 571–586. discussion 587–8 (2014). <https://doi.org/10.1111/aogs.12412>
14. E. Blix, K.G. Brurberg, E. Reierth, L.M. Reinars, P. Øian, ST waveform analysis versus cardiotocography alone for intrapartum fetal monitoring: a systematic review and meta-analysis of randomized trials. *Acta Obstet. Gynecol. Scand.* **95**(1), 16–27 (2016). <https://doi.org/10.1111/aogs.12828>
15. E. Heintz, T. Brodtkorb, N. Nelson, L. Levin, The long-term cost-effectiveness of fetal monitoring during labour: a comparison of cardiotocography complemented with ST analysis versus cardiotocography alone. *BJOG* **2008**(115), 1676–1687 (2008)
16. H. Norén, A. Carlsson, Reduced prevalence of metabolic acidosis at birth: an analysis of established STAN usage in the total population of deliveries in a Swedish district hospital. *Am. J. Obstet. Gynecol.* **202**(6), 546.e1–546.e7 (2010). <https://doi.org/10.1016/j.ajog.2009.11.033>
17. I. Amer-Wählin, A. Ugwuamadu, B.M. Yli, A. Kwee, S. Timonen, V. Cole, D. Ayres-de-Campos, G.E. Roth, C. Schwarz, L.A. Ramenghi, T. Todros, V. Ehlinger, C. Vayssiére, Study group of intrapartum fetal monitoring (European association of perinatal medicine).: fetal electrocardiography ST-segment analysis for intrapartum monitoring: a critical appraisal of conflicting evidence and a way forward. *Am. J. Obstet. Gynecol.* **221**(6), 577–601.e11 (2019). <https://doi.org/10.1016/j.ajog.2019.04.003>
18. G.S. Dawes, C.R.S. Houghton, C.W.G. Redman, Baseline in human fetal heart-rate records. *Br. J. Obstet. Gynaecol.* **89**(4), 270–5 (1982)
19. R. Mantel, H.P. van Geijn, F.J. Caron, J.M. Swartjes, E.E. van Woerden, H.W. Jongsma, Computer analysis of antepartum fetal heart rate: 2. Detection of accelerations and decelerations. *Int. J. Biomed. Comput.* **25**(4), 273–86 (1990)
20. J. Pardey, M. Moulden, C.W.G. Redman, A computer system for the numerical analysis of nonstress tests. *Am. J. Obstet. Gynecol.* **186**(5), 1095–1103 (2002)
21. G.G. Georgoulas, C.D. Stylios, G. Nokas, P.P. Groumpas, Classification of fetal heart rate during labour using hidden Markov models, in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, Budapest, vol. 3 (2004), pp. 2471–2475
22. C. Sundar, M. Chitradevi, G. Geetharamani, Classification of cardiotocogram data using neural network based machine learning technique. *Int. J. Comput. Appl.* **47**(14) (2012). ISSN 0975-888
23. K. Agrawal, H. Mohan, Cardiotocography analysis for fetal state classification using machine learning algorithms, in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, Tamil Nadu, India (2019), pp. 1–6
24. V. Nagendra, H.G. Divya, S.S. Corns, S. Long, Evaluation of support vector machines and forest classifiers in a real-time fetal monitoring system based on cardiotocography data, in *2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Manchester (2017), pp. 1–6
25. Z. Cömerta, A.F. Kocamaz, Comparison of machine learning techniques for fetal heart rate classification, in *Special issue of the 3rd International Conference on Computational and Experimental Science and Engineering (ICCESEN 2016)* (2016)
26. Z. Cömert, A.F. Kocamaz, A novel software for comprehensive analysis of cardiotocography signals “CTG-OAS”, in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya (2017), pp. 1–6
27. H. Ocak, A medical decision support system based on support vector machines and the genetic algorithm for the evaluation of fetal well-being. *J. Med. Syst.* **37**, 9913 (2013). <https://doi.org/10.1007/s10916-012-9913-4>
28. M.E.B. Menai, F.J. Mohder, F. Al-mutairi, Influence of feature selection on Naïve Bayes classifier for recognizing patterns in cardiotocograms. *J. Med. Bioeng.* **2**(1), 66–70 (2013). <https://doi.org/10.12720/jomb.2.1.66-70>

29. E.M. Karabulut, T. Ibrikci, Analysis of cardiotocogram data for fetal distress determination by decision tree based adaptive boosting approach. *J. Comput. Commun.* **2**, 32–37 (2014). <https://doi.org/10.4236/jcc.2014.29005>
30. P.A. Warrick, E.F. Hamilton, D. Precup, R.E. Kearney, Classification of normal and hypoxic fetuses from systems modeling of intrapartum cardiotocography. *IEEE Trans. Biomed. Eng.* **57**(4), 771–779 (2010). <https://doi.org/10.1109/tbme.2009.2035818>. PMID 20659819
31. E.Y. Jilmaz, Ç. Kılıkçer, Determination of fetal state from cardiotocogram using LS-SVM with particle swarm optimization and binary decision tree. *Comput. Math Method. Med.* **2013**, 487179 (2013). <https://doi.org/10.1155/2013/487179>
32. L. Xu, C.W.G. Redman, S.J. Payne, A. Georgieva, Feature selection using genetic algorithms for fetal heart rate analysis. *Physiol. Meas.* **35**(7), 1357–1371 (2014). <https://doi.org/10.1088/0967-3334/35/7/1357>
33. A. Georgieva, S.J. Payne, M. Moulden et al., Artificial neural networks applied to fetal monitoring in labour. *Neural Comput. Appl.* **22**, 85–93 (2013). <https://doi.org/10.1007/s00521-011-0743-y>
34. J. Mathew, C.K. Pang, M. Luo, W.H. Leong, Classification of Imbalanced data by oversampling in Kernel space of support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(9), 4065–4076 (2018)
35. J. Spilka, J. Frecon, R. Leonarduzzi, N. Pustelnik, P. Abry, M. Doret, Sparse support vector machine for intrapartum fetal heart rate classification. *IEEE J. Biomed. Health Inform.* **21**(3), 664–671 (2017)
36. P. Fergus, A. Hussain, D. Al-Jumeily, D.-S. Huang, N. Bouguila, Classification of caesarean section and normal vaginal deliveries using foetal heart rate signals and advanced machine learning algorithms. *BioMed. Eng. OnLine* **16**, 89 (2017). <https://doi.org/10.1186/s12938-017-0378-z>
37. S. Rhöse, A.M.F. Heinis, F. Vandenbussche, J. Van Drongelen, J. Van Dillen, Inter-observer and intra-observer agreement of non-reassuring cardiotocography analysis and subsequent clinical management. *Acta Obstet. Gynecol. Scand* **93**, 596–602 (2014)
38. Z. Zhao, Y. Zhang, Z. Comert, Y. Deng, Computer-aided diagnosis system of fetal hypoxia incorporating recurrence plot with convolutional neural network. *Front. Physiol.* **10**, 255 (2019). <https://doi.org/10.3389/fphys.2019.00255>
39. P. Fergus, C. Chalmers, C.C. Montanez, D. Reilly, P. Lisboa, B. Pineles, Modelling segmented cardiotocography time-series signals using one-dimensional convolutional neural networks for the early detection of abnormal birth outcomes. *IEEE Trans.* (in press) (2019). <https://arxiv.org/abs/1908.02338>
40. A. Petrozziello, C.W.G. Redman, A.T. Papageorghiou, I. Jordanov, A. Georgieva, Multimodal convolutional neural networks to detect fetal compromise during labor and delivery. *IEEE Access* **7**, 112026–112036 (2019)
41. D. Urmull, A. Salter, B. Simpson et al., Comparing the effect of STan (cardiotocographic electronic fetal monitoring (CTG) plus analysis of the ST segment of the fetal electrocardiogram) with CTG alone on emergency caesarean section rates: study protocol for the STan Australian randomised controlled trial (START). *Trials* **20**, 539 (2019). <https://doi.org/10.1186/s13063-019-3640-9>

Enhanced Flower Pollination Algorithm for Task Scheduling in Cloud Computing Environment



Timea Bezdan, Miodrag Zivkovic, Milos Antonijevic, Tamara Zivkovic, and Nebojsa Bacanin

Abstract Cloud computing technology refers to on-demand access to services, applications, and infrastructure that runs on a distributed network utilizing virtualized resources. In the cloud model, an efficient task scheduling algorithm plays an important role in order to achieve better functioning in general and resource utilization of the cloud. The end-users submit tasks, and the scheduling algorithm needs to allocate them to the available resources on time. Task scheduling issue is considered as NP-hard problems, and metaheuristics algorithms demonstrate high efficiency in solving such problems, thus, in this work, we propose enhanced flower pollination algorithm for the task scheduling. The major focus of this study is to reduce the makespan. We compared the results of the proposed method to other similar approaches, such as PBACO, ACO, Min-Min, and FCFS allocation strategies. The obtained results from the experiment show that the proposed EEFPA scheduler has the potential to allocate submitted tasks by the user to the available resources on the cloud.

Keywords Flower pollination algorithm · Optimization · Metaheuristics · Cloud computing · Task scheduling · Makespan · NP hard

T. Bezdan (✉) · M. Zivkovic · M. Antonijevic · N. Bacanin
Singidunum University, Belgrade, Serbia
e-mail: tbezdan@singidunum.ac.rs

M. Zivkovic
e-mail: mzivkovic@singidunum.ac.rs

M. Antonijevic
e-mail: mantonijevic@singidunum.ac.rs

N. Bacanin
e-mail: nbacanin@singidunum.ac.rs

T. Zivkovic
School of Electrical Engineering, Belgrade, Serbia
e-mail: zt125040p@student.etf.bg.ac.rs

1 Introduction

Cloud computing technology enables on-demand access to an elastic pool of shareable virtual or physical resources to the cloud user via a network (Internet). It provides dynamic, flexible, scalable, on-demand services to the cloud users on the basis of the flexible pay-as-you-go payment plan. One party (the user) submit requests to the cloud, and another party, the cloud service provider, execute the submitted task. To improve performance, an efficient task scheduling algorithm is required. Task scheduling is classified as an NP hard problem. Deterministic algorithms fail in solving complex optimization problems; it can get trapped in the local minima and does not guarantee to find the global optimum. However, we can use another approach to find an approximate solution, which may not be the best solution, but it is close to the best. Metaheuristics algorithms are stochastic approximation algorithms, and they are proven to be competent in addressing complex optimization problems, such as task scheduling; thus, in this paper, a swarm-based metaheuristic scheduler algorithm is proposed to schedule tasks in the cloud model. Metaheuristic algorithms are characterized by randomization; in any metaheuristic algorithm, the initial population is generated randomly. The main processes in metaheuristics algorithms are exploration and exploitation, and they refer to the way of search area exploration. The exploitation process indicates to the local search; on the other hand, the exploration process refers to the global search space exploration. To establish the right balance between exploration and exploitation is essential in all metaheuristic optimization algorithms. Swarm-intelligent based algorithms were tested on benchmark problems [1], and also have a lot of successful application in solving different real-life optimization problems, such as portfolio optimization [2], computer vision [3], clustering [4], scheduling problem [5], wireless sensor node localization [6, 7], radio frequency identification network planning [8].

According to previously conducted research, swarm intelligence can be successfully applied for task scheduling in cloud computing [9–11]. The basic assumption behind this research is that tackling cloud scheduling challenges can be further enhanced by using swarm algorithms. We improved and adapted the flower pollination algorithm (FPA) [12] for the cloud computing task scheduling problem. We note that the basic version of the FPA was previously applied for this problem; however, different models, as well as data set, was used [13].

The rest of the paper is organized as follows: Cloud computing concept and architecture are described in Sect. 2, Sect. 3 gives the description of the proposed model, Sect. 4 explains the basic, as well as improved flower pollination algorithm, that was proposed in this paper. Section 5 shows the simulation result and comparative analysis, and Sect. 6 summarizes the conclusion of the paper and future work.

2 Concept and Architecture of Cloud Computing

Cloud computing technology allows access to computing power services in a virtualized environment to the user. According to how the cloud services are made available to the end-users, the service models in cloud computing are categorized as Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). SaaS provides applications to the end-user, which are running on the provider's cloud infrastructure. IaaS provides computing resources to the consumer, such as storage, network, CPU, memory, etc. PaaS provides an environment for developers to create and deploy applications, using programming languages, services, libraries, and tools supported by the cloud service provider.

Comparing the architecture of cloud computing to traditional IT architecture, it has a few key differences, such as elasticity, security, utilization, scalability, etc. The architecture of cloud computing has four layers; the data center, infrastructure, platform, and the fourth layer is the application layer. The first layer is the datacenter layer or hardware layer, which manages physical resources, such as servers, switches, routers, CPU, memory, storage, etc. The next layer is the infrastructure layer; in this layer, by the utilization of different virtualization technologies, the physical resources are partitioned into a set of virtual resources. The third layer, the platform layer, consists of an operating system and application software. Finally, the application layer is on the top, which includes various cloud services used by the end-user.

3 Proposed Task Scheduling Model

In this section, we describe the model that is proposed for scheduling tasks in cloud computing. The objective of this work is to minimize the makespan, which is the total required time to execute all submitted tasks. The cloud infrastructure consists of n physical machines, each physical machine has j virtual machines, and it can be described as follows: $P_n = \{\text{VM}_1, \text{VM}_2, \dots, \text{VM}_j\}$, where P_n denotes the n -th physical machine and VM denotes the virtual machines. Each virtual machine (VM) consists of the set with two properties: $\text{VM}_j = \{\text{IDVM}_j, \text{ESVM}_j\}$, where IDVM_j denotes the identification number, and ESVM denotes the j th virtual machine's execution speed, expressed in million instructions per second (MIPS). The end-users submit tasks, which should be allocated to the available VMs, the set of a task is described as follows: $T = \{T_1, T_2, \dots, T_i\}$, where the total number of submitted tasks is i . The algorithm needs to map i tasks to j virtual machines. Each task (T) can be represented as the set of: $T_i = \{\text{IDT}_i, L_i, \text{ETC}_i, P_i\}$, where the identification number (ID) of the i -th task is denoted by IDT, L represents the length, which is expressed in million instructions, ETC denotes the expected time for completing a task, and P denotes the priority of the i th task.

The aim of this paper is to reduce the makespan (MS). Initially, the expected time for completing each task is calculated as follows:

$$\text{ETC}_{i,j} = L_i / \text{ESVM}_j \quad (1)$$

which generates a matrix with dimensions $i \times j$.

The execution time of all VMs is calculated by using the following formula:

$$\text{ET}_{i,j} = x_{j,i} \times \text{ETC}_{i,j} \quad (2)$$

where x denotes the decision variable, which has value 1, if the task is allocated to the virtual machine; otherwise, its value is 0.

The resulted makespan of the proposed algorithm is calculated as follows:

$$\begin{aligned} \text{MS} &= \max(\text{CT}(T_i, \text{VM}_j)) \\ \text{CT}_{ij} &= \text{ET}_{ij} + \text{VM}_j \end{aligned} \quad (3)$$

where the makespan is denoted by MS, CT denotes the time of completion of a virtual machine. The time of the execution is represented by ET, and VM_j represents the j th the virtual machine.

4 Flower Pollination Algorithm Overview

Flower pollination algorithm (FPA) [12] is a successful swarm intelligence (SI) algorithm. FPA is inspired by the transfer of the flower's pollen. The FPA is tested on ten different standard unconstrained functions, and on one mechanical engineering benchmark, on the pressure vessel design problem. The simulation results are compared to other metaheuristic algorithms, to the popular particle swarm optimization (PSO) [14] and genetic algorithm (GA) [15]. The results show that FPA outperformed both algorithms, GA and PSO, respectively. The pollination process is needed in order to reproduce flower plants. In the FPA algorithm, the pollination process is represented by four rules: Rule 1: Cross-pollination and biotic pollination represent the process of global pollination, and pollen transfer is performed by Lévy flights [16]; Rule 2: Self-pollination and abiotic pollination represent the process of local pollination; Rule 3: The probability of the reproduction represents the flower constancy behavior, and it is proportional to two involved flowers similarity; Rule 4: The switch probability has a value between 0 and 1, and it controls the global as well as the process of local pollination.

For the sake of simplicity, in the algorithm, a plant only has one flower (solution), and each flower produces one pollen gamete. The population of solutions consists of flowers, and each flower represents a solution x_i in the solution space. The algorithm has two key processes, which are the local pollination and global pollination. These two processes represent the exploration and the exploitation process, respectively.

The first rule and the third rule (flower constancy) represents the global pollination process and it can be mathematically formulated as follows:

$$x_i^{t+1} = x_i^t + L(g_* - x_i^t) \quad (4)$$

where the i -th individual (solution) at iteration t is denoted by x_i^t , g_* represents the current best solution. L denotes the step size, and this characteristic is simulated by the Lévy flight.

The step size, $L > 0$ and it is calculated as:

$$L \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0) \quad (5)$$

where $\Gamma(\lambda)$ indicates to the gamma function, and its distribution is valid for large steps $s > 0$. The recommended value for λ is 1.5.

The local pollination process is defined by the second and the third rule (flower constancy), which is formulated as follows:

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t) \quad (6)$$

where the i -th solution in the population at iteration t is denoted by x_i^t ; x_j^t and x_k^t represents the pollen from various flowers of the same type of plant, and in case that the value of ϵ is in the range $[0, 1]$, then the equation is local random walk; thus the newly generated solution will be close to the current solution.

The switch probability is the fourth rule in the algorithm, which determines whether the solution will be updated by the procedure of the local or global pollination. The starting value of the switch probability p can be 0.5; for most applications $p = 0.8$ is more efficient [12].

4.1 Our Modification

By performing simulation on standard unconstrained benchmark functions, we observed that the original FPA approach has some shortcomings. At the beginning of execution, in some runs, because of the lack of exploration power, the algorithm is not able to discover the right section of the search area. In this scenario, the algorithm performs search in the neighborhood of already discovered solutions, and new domains of the search space are not explored. As a consequence, the algorithm gets stuck in some of the sub-optimal domains, and its convergence is premature.

To avoid this, in our proposed implementation, in the first 30% of iterations, we remove from the population the worst individuals, and they are replaced by a new random solution. The random solutions are created in an identical way as in the phase of population initialization. We named our approached exploration-enhanced FPA (EEFPA). Simulations of the EEFPA on standard test functions show that the EEFPA results with a better quality of solutions along with better convergence speed.

5 Experimental Results and Discussion

The experimental results of the proposed EEFPA task scheduler are described in this section. The experiments are performed by the CloudSim toolkit. CloudSim is a framework for cloud infrastructure simulation and modeling. The control parameter $\lambda = 1.5$ and the value of the switch probability is 0.5, according to the suggestion in [12]. The model of the system is in reference to the work [17], and the experiments are performed with a different number of tasks (user request), 100, 200, 300, 400, 500, and 600, similarly like in [17].

For the purpose of performance estimation of the introduced approach, more than one run with independent random population initialization should be made. Accordingly, in this study, the results are collected in 100 independent experiments. In the final results, the average makespan value of 100 runs is reported. For experimental purposes, an artificial dataset is used, which is generated by the CloudSim tool. In every run, the random cloud infrastructure consists of 10 VMs. Each VM's processing power from 1860 to 2660 MIPS, the length of the tasks are formed 400 to 1000, and the memory of all VMs is 4GB of RAM. The randomly generated tasks by the CloudSim have different requirements in terms of memory, CPU, etc. The arrival rate of the tasks is set to 10 requests/second in all conducted experiments. The experimental outcome of the proposed technique is compared to other comparable approaches, such as PBACO (performance budget ACO), ACO (Ant Colony Optimization), Min-Min,

Algorithm 1 EEFPA pseudocode

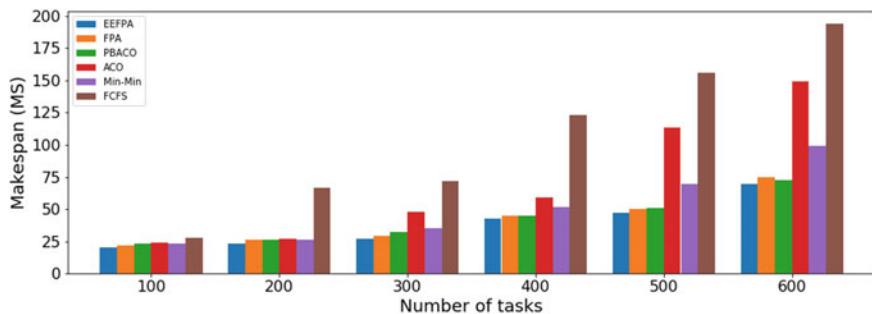
```

Initialize randomly the population of  $N$  flowers (individuals)
Evaluate the fitness and save the fittest solution  $g_*$ 
Determine the value of the switch probability
Set the iteration counter  $t$  to 1 and define the number of maximum iteration  $MaxIter$ 
while  $t < MaxIter$  do
    for  $i = 1$  to  $N$  do
        if  $rand < p$  then
            Draw a ( $d$ -dimensional) step vector  $L$  which obeys a Lévy distribution
            Apply the the procedure of the global pollination by utilizing Eq. (4)
        else
            Draw  $\epsilon$  from the uniform distribution  $\in [0, 1]$ 
            Select randomly  $j$  and  $k$  among all possible individuals from the population
            Apply the Local pollination process by using Eq. (6)
        end if
        Evaluate the newly generated individuals
        If the newly generated individuals have better fitness value, update them in the population
    end for
    Find and save the best current individual  $g_*$ 
    if  $t \leq MaxIter * 0.3$  then
        Replace the worst solution from population with random solution
    end if
end while
Return the fittest individual in the population

```

Table 1 Comparison of EEFPA and FPA with other methods

| No. of tasks | EEFPA | FPA | PBACO | ACO | Min-Min | FCFS |
|--------------|-------|-----|-------|-----|---------|------|
| 100 | 20 | 22 | 23 | 24 | 23 | 28 |
| 200 | 23 | 26 | 26 | 27 | 26 | 67 |
| 300 | 27 | 29 | 32 | 48 | 35 | 72 |
| 400 | 43 | 45 | 45 | 59 | 52 | 123 |
| 500 | 47 | 50 | 51 | 113 | 70 | 156 |
| 600 | 70 | 75 | 73 | 149 | 99 | 194 |

**Fig. 1** Visual representation of the comparative methods**Table 2** EEFPA performance versus performance of other methods

| Number of tasks | EEFPA versus FPA (%) | EEFPA versus PBACO (%) | EEFPA versus ACO (%) | EEFPA versus Min-Min (%) | EEFPA versus FCFS (%) |
|-----------------|----------------------|------------------------|----------------------|--------------------------|-----------------------|
| 100 | 9.09 | 13.04 | 16.67 | 13.04 | 28.57 |
| 200 | 11.54 | 11.54 | 14.81 | 11.54 | 65.67 |
| 300 | 6.9 | 15.63 | 43.75 | 22.86 | 62.5 |
| 400 | 4.44 | 4.44 | 27.12 | 17.31 | 65.04 |
| 500 | 6 | 7.84 | 58.41 | 32.86 | 69.87 |
| 600 | 6.67 | 4.11 | 53.02 | 29.29 | 63.92 |

and FCFS (First Come First Serve) allocation strategies. In the comparative analysis, the results of other methods are taken from [17]. The proposed method is tested with a population size 20 with 50 iterations. The results are shown in the Table 1 and its visual representation in Fig. 1. The proposed methods of performance improvement are presented in Table 2. On average, the proposed approach achieves higher performance than all other compared heuristics techniques as well as metaheuristics methods.

6 Conclusion

This work presents the proposed scheduler algorithm (EEFPA) in a cloud model for allocating tasks to the available virtual machines; the scheduler is on the basis of enhanced flower pollination swarm intelligence algorithm. This work has aim to reduce and minimize the makespan (MS) value. The experiments are carried out on the CloudSim simulation platform. The experiment is conducted on a different number of tasks, 100, 200, 300, 400, 500, and 600. The experimental outcome of the proposed technique is compared to other comparable approaches, such as PBACO, ACO, Min-Min, and FCFS allocation strategies. It can be concluded from the obtained results of simulations that the proposed scheduler has the potential to allocate submitted tasks by the user to the available resources on the cloud. In future work, we are going to do multi-objective optimization and, besides the makespan, to optimize other objectives, such as energy consumption, costs minimization of resources, etc. by utilizing other modifications and hybridized versions of the FPA.

References

1. M. Tuba, N. Bacanin, Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems. *Neurocomputing* **143**, 197–207 (2014)
2. M. Tuba, N. Bacanin, Artificial bee colony algorithm hybridized with firefly metaheuristic for cardinality constrained mean-variance portfolio problem. *Appl. Math. Inf. Sci.* **8**, 2831–2844 (2014). November
3. I. Strumberger, E. Tuba, N. Bacanin, R. Jovanovic, M. Tuba, Convolutional neural network architecture design by the tree growth algorithm framework, in *2019 International Joint Conference on Neural Networks (IJCNN)* (2019 July), pp. 1–8
4. J. Senthilnath, S. Omkar, V. Mani, Clustering using firefly algorithm: performance study. *Swarm Evol. Comput.* **1**(3), 164–171 (2011)
5. M. Sayadi, R. Ramezanian, N. Ghaffari-Nasab, A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *Int. J. Ind. Eng. Comput.* **1**(1), 1–10 (2010)
6. I. Strumberger, E. Tuba, N. Bacanin, M. Beko, M. Tuba, Wireless sensor network localization problem by hybridized moth search algorithm, in *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)* (2018 June), pp. 316–321
7. I. Strumberger, M. Minovic, M. Tuba, N. Bacanin, Performance of elephant herding optimization and tree growth algorithm adapted for node localization in wireless sensor networks. *Sensors* **19**(11), 2515 (2019)
8. N. Bacanin, M. Tuba, R. Jovanovic, Hierarchical multiobjective rfid network planning using firefly algorithm, in *2015 International Conference on Information and Communication Technology Research (ICTRC)* (2015 May), pp. 282–285
9. I. Strumberger, N. Bacanin, M. Tuba, E. Tuba, Resource scheduling in cloud computing based on a hybridized whale optimization algorithm. *Appl. Sci.* **9**(22), 4893 (2019)
10. N. Bacanin, E. Tuba, T. Bezdan, I. Strumberger, M. Tuba, Artificial flora optimization algorithm for task scheduling in cloud computing environment, in *Intelligent Data Engineering and Automated Learning—IDEAL 2019*, ed. by H. Yin, D. Camacho, P. Tino, A.J. Tallón-Ballesteros, R. Menezes, R. Allmendinger (Springer International Publishing, Cham, 2019), pp. 437–445

11. N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, M. Tuba, M. Zivkovic, Task scheduling in cloud computing environment by grey wolf optimizer, in *2019 27th Telecommunications Forum (TELFOR)* (2019), pp. 1–4
12. X.-S. Yang, Flower pollination algorithm for global optimization, in *International Conference on Unconventional Computing and Natural Computation* (Springer, 2012), pp. 240–249
13. I. Gupta, A. Kaswan, P.K. Jana, A flower pollination algorithm based task scheduling in cloud computing, in *Computational Intelligence, Communications, and Business Analytics*, ed. by J.K. Mandal, P. Dutta, S. Mukhopadhyay (Springer Singapore, Singapore, 2017), pp. 97–107
14. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95—International Conference on Neural Networks*, vol. 4 (1995), pp. 1942–1948
15. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn. (Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1989)
16. I. Pavlyukevich, Lévy flights, non-local search and simulated annealing. *J. Comput. Phys.* **226**, 1830–1844 (2007)
17. L. Zuo, L. Shu, S. Dong, C. Zhu, T. Hara, A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. *IEEE Access* **3**, 2687–2699 (2015)

Counterfeit Currency Detection Using Supervised Machine Learning Algorithms



R. K. Yadav, Pulkit Valecha, and Shaivya Paliwal

Abstract Counterfeit currency is an extremely common yet pertinent problem of presence of fake, inauthentic or copies of real currency in the market and economy that is faced by various nations across the globe. Cash transactions still make up for over 80% of all transactions. Fake and inauthentic notes, therefore, continue to be a major source of nuisance for the economy. In this paper, the aim is to identify the authenticity of the currency notes by using various machine learning algorithms and also to compare and contrast which of these algorithms is best suited for the same. The machine learning algorithms classify the currency notes on the basis of features extracted from images. The dataset was taken from UCI Machine Learning Repository. We got the best results from K-nearest neighbours classifier with 99.8% accuracy and F-score of 0.992 ($\beta = 2$).

Keywords Counterfeit currency · Fake notes · Supervised learning · Discrete wavelet transform · Machine learning · K-nearest neighbours

1 Introduction

Currency is a major part of every economy and even though the world is moving towards what is said to be a cashless economy, identification of these notes would be of great help not only to banks and big industries and shopkeepers but also to the everyday common man who can use this feature to prevent being scammed. Counterfeit money causes various problems to the society and the economy—it leads

R. K. Yadav (✉) · P. Valecha · S. Paliwal
Delhi Technological University, New Delhi, India
e-mail: rkyadav@dtu.ac.in

P. Valecha
e-mail: pulkitvalecha98@gmail.com

S. Paliwal
e-mail: shaivyapaliwal@gmail.com

to the reduction in value of the real money, it causes inflation since more currency flows in the market, and it is a part of black money and corruption that takes place. Therefore, it is easy to understand the requirement of getting rid of the counterfeit currency in the market. As per a report by the Reserve Bank of India (RBI), 5.22 Lakh fake notes were discovered during the financial year 2017–2018. There was a 28 times increase in the presence of fake notes of 2000 denomination after the launch of the new notes in 2016. There was an overall 35% increase in the counterfeit currency in India whose estimated value is around Rs. 28.1 Crore [1].

Indian currency notes have a variety of security features and multiple image processing techniques have been used to distinguish real from fake [2–7].

2 Database

The database was taken from the UCI Machine Learning Repository. It contains extracted features from images of banknotes, both real and in-genuine ones. Features were extracted from the images using wavelet transform tools [8, 9]. The final images were grey-scaled, with size 400×400 pixels and a resolution of about 660 dpi. The dataset contained 1372 samples, 762 fake and 610 real [10].

3 Performance Measures

Various metrics were used to measure and determine the performance of the various classifiers used to determine the authenticity of the currency notes. For all the classifiers, confusion matrices, accuracy and F-score were calculated. F-score is used with beta value equal to 2, as it weighs recall more than precision. It is important to have fewer false negatives, i.e. we need to have less samples that are classified as real (negative, since not fake) but are actually fake (positive), hence, false negative. On the other hand, when we consider precision, an error in precision would be a note classified as fake (positive) when it is not (hence, false positive). This is less detrimental as compared to the false negative error.

3.1 Benchmark Model

The performance of the various supervised machine learning algorithms was compared to a Naive predictor. This Naive predictor classified all the banknotes as fake. In the currency detector, it is more vital to identify a fake note as fake than it is to identify a real note as real, and therefore, the Naive Predictor becomes an ideal comparator. The accuracy of the benchmark model or the Naive predictor comes out to be 55.5% and the F-score ($\beta = 2$) is 0.862.

4 Pre-processing

The dataset was pre-processed before the classification by the supervised learning algorithms. The five algorithms of supervised machine learning used were—support vector machine (SVM), decision trees, K-nearest neighbours (KNN), gradient boosting classifier and artificial neural networks (ANN). The data distribution was checked and it was found that all four features used for classification were well distributed and the dataset does not have any feature which has outlier values. All four data distributions were well distributed. The dataset was normalized to ensure equal treatment of the features by the classifiers. The features were scaled between [0, 1]. It was then split into the training set and testing set in the ratio of 6:4 (60% of the dataset used for training and 40% used for testing) for the supervised machine learning algorithms. The dataset was shuffled prior to the split to ensure equitable distribution of samples from both classes.

5 Training and Testing Data on Machine Learning Algorithms

A function was defined that took four classifiers as the input—KNN, SVM, decision trees and gradient booster. A separate function was made for the definition, compilation and training of the artificial neural network. These two functions were used to fit their respective learners or classifiers on the sample size of the dataset, where the sample size was given as 10, 25 and 100% of the training set. This was done to see the variation of the performance of the models with the various sizes of training set. After the training was done, the time taken for training the model was also noted. The next task was testing the model on the test data, wherein the time for prediction was also noted. After the models have predicted the data, the results were stored (Fig. 1).

6 Result and Analysis

Table 1 summarizes the accuracy and F-score of various models on which the dataset was trained and tested.

Figure 2 gives the comparative analysis of the time taken, accuracy and F-score on training and testing dataset, for each of the 10, 25 and 100% of the dataset.

All the classifiers perform well on the dataset, with KNN having the highest F-score as well as the highest accuracy on the testing dataset (results on 100% of the dataset), and hence becomes the best classifier for the dataset. The following observations were made on the basis of the results:

1. SVM and KNN and ANN detect all the fake notes.

Fig. 1 Flowchart describing training and testing of data

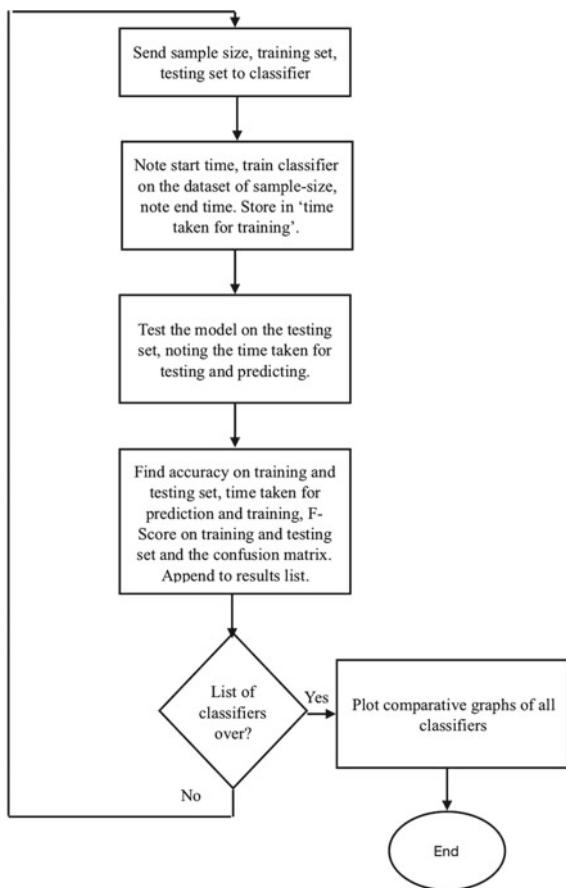


Table 1 Accuracy and F-score on training and testing data for all classifiers

| | KNN | Support vector machine | Gradient boosting classifier | Decision trees | ANN |
|-----------------------|-------|------------------------|------------------------------|----------------|-------|
| Training set accuracy | 1 | 0.986 | 1 | 1 | 0.98 |
| Testing set accuracy | 0.998 | 0.979 | 0.994 | 0.974 | 0.983 |
| Training set F-score | 1 | 0.994 | 1 | 1 | 0.985 |
| Testing set F-score | 0.992 | 0.991 | 0.99 | 0.974 | 0.982 |

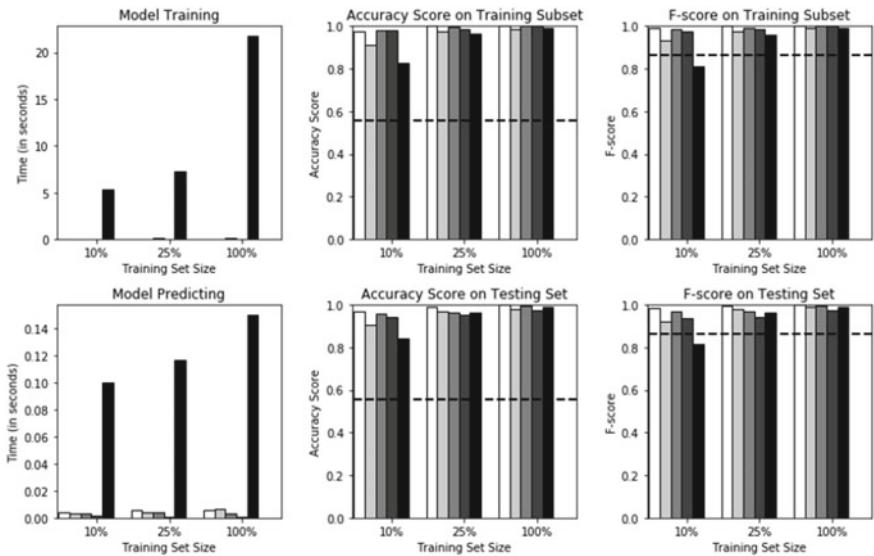


Fig. 2 Comparative analysis

2. Gradient boosting classifier and KNN have lowest false positives and highest precision.
3. SVM and KNN and ANN have lowest false negatives and highest recall.
4. KNN has an accuracy of 99.8 on test set and F-score of 99.2 which is the best performer out of all the classifiers.
5. From the graphs plotted, it is clear that KNN is the winner in all cases, although other algorithms also perform really well.
6. ANN takes much more time than any other classifier and if ANN is excluded from the set, KNN takes maximum time during testing, however, the time is in fraction of seconds and can be said to be a really good performance.

All five classifiers performed much better than the benchmark model used to evaluate the supervised classifiers. The accuracy and F-score of all of these were much better than the benchmark model. The results also show the variation of the performance on the basis of the size of the training set and testing set. The time taken for the training of the model and testing of the model is also clearly shown, wherein the ANN takes most time.

KNN is selected as the best classifier because it has the highest accuracy and highest F-score. On the testing dataset, it correctly identified all the fake currency notes. It also had the best performance even when parts (10%, 25%) of the dataset were taken. Only one genuine note was misclassified by KNN. In the problem of detection of fake and counterfeit currency, a few false positives are more acceptable than false negatives, since calling a real banknote inauthentic is less problematic than calling a fake banknote authentic. It was run with various values of number of

neighbours and gave good results and it is therefore robust with various values of nearest neighbours and sizes of dataset.

Therefore, on the basis of the aforementioned reasons, it was selected as the best classifier.

7 Conclusion

Detection of fake and inauthentic currency in the economy is vital and need of the hour. This system solves the problem using features extracted from wavelet transform and machine learning techniques to classify the banknote as real or fake. The best performance out of all the classifiers is obtained using the KNN classifier with 99.8% accuracy and 0.992 F-score on the chosen database. The time taken by the classifier is also in order of fraction of seconds, which can be considered as good performance. Focusing on this problem, it can be said that a few false positives are relatively more acceptable than any false negatives. KNN having the highest F-score out of all the classifiers thus becomes the best classifier.

References

1. Reserve Bank of India, Annual Report (2017–18). <https://rbidocs.rbi.org.in/rdocs/AnnualReport/PDFs/0ANREPORT201718077745EC9A874DB38C991F580ED14242.PDF> Last accessed 03/02/2020
2. A. Upadhyay, V. Shokeen, G. Srivastava, Counterfeit currency detection techniques, in *8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (2018)
3. R.M. Raut, K.K. Warhade, Counterfeit currency detection. *Int. J. Trend Sci. Res. Dev.* **2**(4) (2018)
4. S.R. Darade, G.R. Gidveer, Automatic recognition of fake Indian currency note, in *International Conference on Electrical Power and Energy Systems (ICEPES)* (2016)
5. Z. Ahmed, S. Yasmin, R.U. Ahmed, Md.N. Islam, Image processing based Feature extraction of Bangladeshi banknotes, in *8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)* (2014)
6. V. Lohweg et al., Banknote authentication with mobile devices, in *IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics* (2013)
7. M. Deborah, C.S. Prathap, Detection of fake currency using image processing. *Int. J. Innov. Sci. Eng. Technol.* **1**(10) (2014)
8. E. Choi, J. Lee, J. Yoon, Feature extraction for bank note classification using wavelet transform, in *The 18th International Conference on Pattern Recognition (ICPR'06)* (2006)
9. S. Kamal, S.S. Chawla, N. Goel, B. Raman, Feature extraction and identification of Indian currency notes, in *Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)* (2015)
10. UCI Machine Learning Repository, Banknote Authentication Database. <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>. Last accessed 15/01/2019

Spam Mail Classification Using Ensemble and Non-Ensemble Machine Learning Algorithms



Khyati Agarwal, Prakhar Uniyal, Suryavanshi Virendrasingh, Sai Krishna, and Varun Dutt

Abstract Spam in emails has been a prevalent issue ever since the inception of the email service. However, the use of ensemble (aggregate) and non-ensemble algorithms for the detection and filtering of spam has been less explored. In this paper, we develop certain ensemble and non-ensemble machine learning (ML) algorithms for classifying emails as spam or ham (i.e., not spam). Using the Enron-SMS dataset from the UCI ML repository and an 80 and 20% training and test split, we develop and calibrate non-ensemble ML algorithms like KNN, Naïve Bayes, and Support Vector Machine. Also, we develop and calibrate ensemble ML algorithms containing the non-ensemble algorithms via voting, bagging, and boosting methods. Results reveal that the non-ensemble Support Vector Machine performed the best with 98.47% accuracy on test data and it was followed by the ensemble voting algorithm with 96.80% accuracy on test data. We highlight the implications of using non-ensemble and ensemble methods for spam classification in the real world.

Keywords Spam · Ham · Tokenizing · KNN · Naïve Bayes · Support vector machine · Ensemble classifiers · Voting · Bagging · Boosting

K. Agarwal · P. Uniyal (✉) · S. Virendrasingh · S. Krishna · V. Dutt
Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi, Mandi, HP 175075, India
e-mail: b18128@students.iitmandi.ac.in

S. Virendrasingh
e-mail: b16037@students.iitmandi.ac.in

S. Krishna
e-mail: b18056@students.iitmandi.ac.in

V. Dutt
e-mail: varun@iitmandi.ac.in

1 Introduction

Spam mail detection is a challenge not just for electronic mail (email) services but also for every free or cheap communication channel service provider [1]. As distinguishing spam from ham (i.e., not spam) in emails is a classification exercise, a number of machine learning methods may be relevant for this classification [1, 2].

Prior research has proposed several non-ensemble ML algorithms like KNN, Naïve Bayes, and Support Vector Machine for email spam classification [2–5]. For example, reference [4] proposed a support machine vector-based Naive Bayes-SVM-NB-filtering system. The SVM-NB constructed an optimal separating hyperplane that divided samples in the training set into two categories, spam or ham. Similarly, reference [3] compared the KNN algorithm, Naïve Bayes algorithm, and the reverse DBSCAN algorithm to classify spam from ham on the Enron-SMS dataset from the UCI ML repository. Results revealed that the Naïve Bayes approach yielded the best accuracy of 86.83% on processed test data.

Prior research has also proposed several ensemble ML algorithms relying upon voting, bagging, and boosting [6–9]. For example, reference [10] proposed a new dynamic weighted voting method based on the combination of clustering and weighted voting and applied it to the task of spam filtering. Results revealed that the voting ensemble algorithm outperformed pure SVM. Similarly, reference [10] showed that the adoption of an ensemble under-sampling classification strategy, which exploits the information involved in a large number of reputable Websites to full advantage, improved the classification of webspam on a standard WEBSPAM-UK2006 dataset.

Although prior research has proposed both non-ensemble and ensemble algorithms, an evaluation of these approaches on a standard dataset has been less explored. The main objective of this research is to compare the performance of a number of popular non-ensemble and ensemble approaches on a standard email dataset. Specifically, we develop and calibrate both non-ensemble algorithms (KNN, Naïve Bayes, and Support Vector Machine) and ensemble versions of the non-ensemble algorithms via voting, bagging, and boosting methods on the standard Enron-SMS dataset from the UCI ML repository [11] to classify email messages as “spam” or “not spam”.

In what follows, we first detail the Enron-SMS dataset and its preprocessing for extracting features. Next, we detail the ensemble and non-ensemble ML algorithms. Furthermore, we detail the method of calibration of these algorithms on the Enron-SMS dataset. Finally, we detail the results of evaluating different algorithms and discuss the likely reasons for our results.

2 Data Preprocessing and Feature Extraction

The dataset used for training our models is the Enron-SMS dataset available at the UCI ML repository [11]. This dataset contains 5572 text messages of which around 4800 messages are non-spam (termed as ham) and around 750 are marked as spam.

For transforming the messages into a suitable input for our model, we removed punctuation marks and other non-alphanumeric characters from the messages. Also, we discarded the occurrences of “stop words” in the messages as they did not carry much weightage in the vectors used for preparing the models using modules of the nltk python library [12].

Then, each email message was converted into a word frequency tuple using the CountVectorizer function of the scikit-learn library [13]. Formally, given a corpus of text documents, the CountVectorizer function first uses a utility for tokenizing the contents of each document and then checks the number of occurrences of each of the tokens in all the documents. All tokens are indexed and their frequencies in each document are represented in the form of a matrix.

Even though the stop words utility of nltk [12] helps in eliminating most of the frequently occurring words of the English language from the email messages, we used the Tf-Idf transformer method [14] as well to reduce the influence of frequently appearing words across messages. Formally, given the frequency matrices corresponding to the documents in a corpus for all the tokenized words, the transformer function transformed the frequencies into floating-point values that would be suitable for usage in classifier models.

$$\text{tfidf}(\text{token}, \text{document}) = \text{tf}(\text{token}, \text{document}) \times \text{idf}(\text{token}) \quad (1)$$

here tf denotes the frequency of token in a document and idf denotes the inverse document frequency of the token. It can be calculated using the formula:

$$\text{idf}(\text{token}) = \log \frac{1 + n}{1 + \text{df}(\text{token})} + 1 \quad (2)$$

here n is the number of documents in the corpus and df is defined as the number of documents in the corpus which have the specific token in them. This transformation ensures that the words that occur more frequently across the corpus have accordingly negligible weights in the vector representation of the document.

We split the dataset in the training: testing ratio of 80:20 and used non-ensemble and ensemble ML algorithms for predicting spam and ham emails.

3 Classifier Models

3.1 K-Nearest Neighbor

Nearest neighbor algorithm has been the foundation of machine learning both for supervised as well as unsupervised learning methods [15]. The algorithm being a lazy learner that first predefines training samples according to their assigned labels and then compares every test sample's distance from the training samples. For classification tasks, Nearest Neighbor algorithm has two popular implementations, the K-Nearest Neighbor Classifier method and the Radius-Based Nearest Neighbor Classifier method [15]. Considering the Enron-SMS dataset, the Radius-Based classifier seemed less effective as it would give more significance to the frequency of tokens rather than the tokens themselves. For example, if there is a token representing the phrase “100% OFF”, heuristically the message would be considered spam irrespective of its frequency in the document; but, our model would only be able to classify it as spam if the frequency of its usage in all documents of the corpus has less variance. Thus, we used K-Nearest Neighbor method in the algorithm. The Nearest Neighbor (NN) method firstly calculates the Euclidean distance D between a vector and other vectors in the database and then select the minimum distance vector. The distance is computed as [15]:

$$D(S, F_i) = \sqrt{\sum_{j=1}^n (s_j - r_j^i)^2} \quad (3)$$

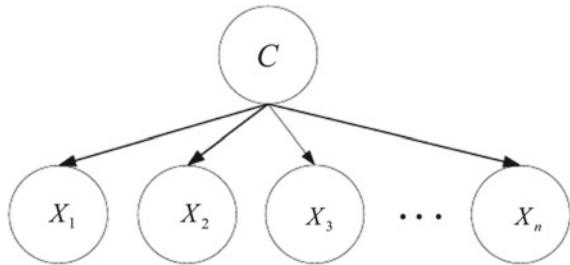
where, S_i represent the anchor point and r_j^i represents the other vector, respectively; and, n represents the number of vectors.

Unlike the NN algorithm, the KNN algorithm requires us to find K ($K > 2$) database vectors which are closest to the resulting vector position measurement. Then, the algorithm figures out the mean of position coordinates that the K database vectors represent. Finally, it gives the average position [10] according to the following equation:

$$(K j^r) = \frac{1}{k} \sum_{i=1}^k (x_i, y_i) \quad (4)$$

where, x_i and y_i represent the coordinates of the i th vector and the positioning result coordinates, respectively.

Fig. 1 Different observations belonging to a class



3.2 Naïve Bayes

The Naive Bayes (NB) classifier [15] uses the assumption that each feature only depends on the class as shown in Fig. 1.

It implies that each feature has the only parent that is class. Naive Bayes is attractive because it has an explicit and sound theoretical basis. This allows optimal induction given a set of explicit assumptions. The independency assumptions of features with respect to the class are violated in some real-world problems, which is a drawback. However, it has been shown that Naive Bayes is remarkably robust in the face of such violations [15]. Also, it is one of the 10 top algorithms in data mining as listed by Wu et al. (2008). Let, C be the class of a particular observation X . To predict the class of the observation, the highest posterior probability should be found, according to the Bayes rule.

$$P(C|X) = \frac{P(C)P(X|C)}{P(X)} \quad (5)$$

where, $P(C)$, $P(X|C)$, $P(X)$ represent the prior probability, likelihood, evidence (probability of X), respectively. We get that features $X_1, X_2, X_3, \dots, X_n$ are conditionally independent of each other given the class, using the assumption.

$$P(C|X) = \frac{P(C) \prod_{i=1}^n P(X_i|C)}{P(X)} \quad (6)$$

where, $P(C)$, $\prod^n P(X_i|C)$, $P(X)$ represent the prior probability, product of all likelihoods, and evidence (probability of X), respectively.

3.3 Support Vector Machines

The concept of Support Vector Machines is to map the training data points from the input space into a higher dimensional vector space of features through a mapping function Φ . Given a training set $S = \{(x_i, y_i) | x_i \in H, y_i \in \{\pm 1\}, i = 1, 2, \dots, l\}$, where

x_i are the input vectors and y_i the labels of the \mathbf{x}_i , the target function is

$$\begin{cases} \min \Phi(\mathbf{w}) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) + C \sum_{i=1}^l \xi_i \\ \text{s.t. } y_i(\langle \mathbf{w} \cdot \phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad i = 1, 2, \dots, l, \end{cases} \quad (7, 8)$$

where, C is the parameter of penalty; ξ_i are non-negative slack variables and, \mathbf{w} is a function variable. The approximating feature map for the Mercer kernel is $K(x, y) = \Phi(x)^T \Phi(y)$. Non-linear mapping is performed by Mercer kernel. Thus, the problem of constructing an optimal hyper-plane is transformed into a problem of quadratic programming as follows:

$$\begin{cases} \max L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t. } \sum_{i=1}^l \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, l. \end{cases} \quad (9)$$

where $L(\alpha)$ is the Lagrangian in z-space, and it is maximized in the equation above.

The decision function can be shown as

$$f(x) = \text{sign} \left[\sum_{i=1}^l y_i \alpha_i K(x_i \cdot x) + b \right]. \quad (10)$$

Depending upon the kernel function chosen, the bias term b may implicitly be included in the kernel function. For example, the bias is not required when kernel function is Gaussian RBF. The most basic kernel functions that are used in SVM are [9]:

$$\begin{aligned} \textbf{Linear Kernel} : K(x, x_i) &= \langle x \cdot x_i \rangle \\ \textbf{Polynomial Kernel} : K(x, x_i) &= (\langle x \cdot x_i \rangle + c)^d \\ \textbf{RBF Kernel} : K(x, x_i) &= \exp(-\|x \cdot x_i\|^2 / 2\sigma^2) \end{aligned} \quad (14)$$

In Eq. 14, σ is the variance of the Gaussian or Radial Basis Function kernel. In all these parameter variations, we got maximum accuracy on the linear function kernel because of its low degree and universal application.

3.4 Ensemble Method Classifiers

Ensemble methods [15] are techniques that build multiple models and then merge them to produce improved results. Ensemble methods usually deliver more accurate results than a regular model as they combine the best of these regular models.

Majority Voting

Every model makes a prediction for each test sample and the final prediction is the one that was made by more than half of the models. If none of the predictions get more than half of the votes, we label the prediction as an unstable one as this is a widely used technique.

Bootstrap Aggregating

The name Bootstrap Aggregating or Bagging sums up the key components of this strategy [15]. In this algorithm, first, we create multiple models working on a similar algorithm with sub-samples of the data drawn randomly from the original data set with bootstrap sampling method. In this sampling method, some examples may appear more than once while others may not even be present in the sample. We make n datasets with m samples each all selected randomly from the original dataset. The second step is aggregating the generated models. In bagging, each sub-sample is generated independently from each other, allowing to do generation and training in parallel.

Boosting

The term “boosting” is given to a family of algorithms which can convert weak models to strong models [15]. The weakness of a model determines whether it has a substantial error rate, but the performance of the model is not random (giving an error rate of 0.5 for binary classification). Boosting method builds an ensemble by training each model with the same data set but the weights of instances are adjusted considering the error in the previous prediction. The concept of optimization is to force the models to focus on the instances which are hard by emphasizing their weight in prediction making. Unlike bagging, boosting is a sequential method, and so the sub-operations of the method cannot be done in parallel.

4 Optimization of Model Parameters

4.1 K-Nearest Neighbor (NN)

In the K-NN algorithm, the most important parameter was the value of K , which decided the number of neighboring training data samples to analyze. We varied the value of K as all odd numbers from 1 to 21. We also varied the weights parameter which decided the weightage of the distance of training data points from the test

sample. For the uniform method, all the K nearest data points were equally weighted, whereas, for the distance method, a closer training data point carried more weightage than a farther data point for classification.

4.2 *Naive Bayes*

In the Naive Bayes algorithm, there are no such parameters to vary and optimize so we chose to vary the distribution model considered for finding likelihoods. We tried the Gaussian distribution, multinomial distribution, and the multivariate Bernoulli distribution.

4.3 *Support Vector Machines*

For SVM, we varied 4 parameters that were Kernel function, Gamma value, constant C , and degree (if the Kernel function was polynomial). The constant C and the gamma function were varied at a logarithmic scale from 0.1 to 1000 while the degree of the polynomial type kernel function was varied at even numbers between 2 and 10. Apart from the polynomial kernel function, some other kernels that we tried were linear, sigmoid, and radial basis function (RBF).

4.4 *Ensembling Classifier with Voting Approach*

In this method, we took the best classifier models we obtained with each algorithm and created a voting classifier from them. The voting type parameter was chosen as hard for obvious reasons.

4.5 *Ensembling Classifier with Boosting Approach*

We used the Adaboost algorithm for this approach and applied it over our best classifier (that was SVM with linear kernel and $C = 10$ and gamma = 10). We kept the number of estimators as 50 and used the SAMME algorithm.

4.6 Ensembling Classifier with Bagging Approach

We used the same best performing classifier as in Boosting and checked the performance changes as there were no other parameters to vary.

5 Results

Table 1 illustrates the results of calibrating different models on the training dataset. On the training data, we got approximately 100% accuracy from the SVM algorithm. From Table 1, SVMs are better models for problems of natural language processing even if compared with other more advanced models and enhancement techniques. SVM in linear kernel with gamma = 10 and $c = 10$ gave the best accuracy on this dataset. In order to validate our findings, we also tried our trained models on the test datasets (see Table 2). The results obtained during test gave almost identical trends across models as in the training dataset. Thus, our models could be used with other datasets for the creation of spam filters or similar applications.

Table 1 Accuracy obtained for different models on training dataset

| Algorithm | Best parameter values | Accuracy | TP ¹ | FP | FN | TN |
|-----------------------------|---|----------|-----------------|----|-----|-----|
| SVM | Kernel: Linear, Gamma: 10, C:10 | 100 | 3851 | 0 | 0 | 606 |
| KNN | $K: 1$, weight: distance | 100 | 2886 | 0 | 0 | 457 |
| Naive Bayes | Multinomial | 97.69 | 3871 | 0 | 103 | 483 |
| Ensemble- Voting Classifier | SVM with $g = 10, c = 10$ | 100 | 3861 | 0 | 0 | 596 |
| Boosting Method | number of estimators: 50, algorithm: SAMME | 86.47 | 3854 | 0 | 603 | 0 |
| Bagging Method | SVM with Gamma: 10 and C: 10 | 99.43 | 3870 | 0 | 25 | 562 |

Note ¹The number of true positive cases

Table 2 Accuracy obtained for different models on test dataset

| Algorithm | Accuracy | TP ¹ | FP | FN | TN |
|----------------------------|----------|-----------------|----|-----|-----|
| SVM | 98.47 | 962 | 6 | 14 | 133 |
| KNN | 94.90 | 1928 | 0 | 142 | 159 |
| Naive Bayes | 95.61 | 982 | 0 | 31 | 102 |
| Ensemble-voting classifier | 96.80 | 968 | 0 | 23 | 124 |
| Boosting method | <90% | 955 | 0 | 160 | 0 |
| Bagging method | 97.3 | 958 | 1 | 29 | 127 |

Note ¹The number of true positive cases

6 Discussion and Conclusion

In this paper, our primary goal was the classification of emails as spam and not-spam (ham) using ensemble and non-ensemble techniques. Our results showed that among the ensemble methods the bagging method performed the best. However, among the non-ensemble methods, the SVM performed the best. Also, overall, the non-ensemble methods performed slightly better compared to the ensemble methods.

First, we found that the SVM algorithm performed the best during both training and test. A likely reason for the better performance of the SVM algorithm is its ability to generate more attributes to an existing attribute set, which helps in linearizes the problem in higher dimensional attribute space.

Second, we found that the ensemble methods did not perform as well as the non-ensemble methods. A likely reason could be that the ensembling of different algorithms confuses the algorithm due to some disagreements compared to a non-ensemble approach that does not get any disagreements during classifications.

We plan to extend our evaluation as part of our future work in several ways. For further testing, we are planning to consider the existing models with newer datasets. In these datasets, we plan to also consider other features of a mail such as sender information, hyperlinks, attachments, and other metadata that can be analyzed for the successful classification of spam as well as malicious mails. We also plan to develop a new classification model over the outcomes of the various prediction models used so far and check how well it can compete with the other models in the literature.

References

1. B. Dada et al, Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon* **5**(6) (2019)
2. W.A. Awad, S.M. ELseuofi, Machine learning methods for spam E-Mail classification. *Int. J. Comput. Sci. Inform. Technol.* **3** (2011). 10.5121/ijcsit.2011.3112
3. A. Harisinghaney, A. Dixit, S. Gupta, A. Arora, Text and image-based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm, in *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, Faridabad, pp. 153–155 (2014)
4. W. Feng, J. Sun, L. Zhang, C. Cao, Q. Yang, A support vector machine based naive Bayes algorithm for spam filtering, in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, Las Vegas, NV, pp. 1–8 (2016)
5. Y. Chen, C. Lu, C. Huang, Anti-spam filter based on Naïve Bayes, SVM, and KNN model (2009)
6. Z. Yang, X. Nie, W. Xu, J. Guo, An approach to spam detection by Naive Bayes ensemble based on decision induction, in *International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 861–866. <https://doi.org/10.1109/isda.2006.253725>
7. J. Carpinter, Evaluating ensemble classifiers for spam filtering (2005)
8. S. Trivedi, S. Dey, Interplay between probabilistic classifiers and boosting algorithms for detecting complex unsolicited emails. *J. Adv. Comput. Netw.* 132–136 (2013). <https://doi.org/10.7763/jacn.2013.v1.27>

9. M.F. Saeedian, H. Beigy, Spam detection using dynamic weighted voting based on clustering, in *2008 Second International Symposium on Intelligent Information Technology Application*, Shanghai, pp. 122–126 (2008)
10. C.W. Geng, Q. Li, L. Xu, X. Jin, Boosting the performance of web spam detection with ensemble under-sampling classification, in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, Haikou, pp. 583–587 (2007)
11. T.A. Almeida, J.M.G. Hidalgo, A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, in *Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11)*, Mountain View, CA, USA (2011). <https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection>. <http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/>
12. NLTK library for Python. <https://www.nltk.org/>
13. CountVectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
14. TfidfTransformer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html
15. C.M. Bishop, *Pattern recognition and machine learning* (Springer, 2006)

On the Desired Properties of Linear Feedback Shift Register (LFSR) Based High-Speed PN-Sequence-Generator



Le Cuong Nguyen, Vu Kien Tran, and Chi Quynh Le

Abstract The linear feedback shift register (LFSR) based PN-sequence generator has found wide-ranging applications in network security, cryptography, compressive sensing, watermark, and so on. Therefore, in the literature, the PN sequences with different specifications have been introduced. Among them, the sequences with interleaved structure (time-multiplexed) have been widely used. However, most of the attentions have been paid to the length and speed of the PN-code-generator while the linear complexity (LC) while the statistic properties of the proposed sequences: have not been thoroughly discussed. In this paper, we will show that one needs to employ different effective tools to evaluate the performances of such sequences so that suitable sequences for application case can be correctly selected.

Keywords High-speed PN-sequences · Period length · Linear complexity · Compressive sensing · Watermark steganography · Encryption · Scramblers

1 Introduction

In recent years, a lot of attentions have been paid for design and evaluation of PN sequences used in network security, cryptography, watermark, steganography, compressive sensing [1–9]. For the above-mentioned applications, the sequences with the following properties are most desired.

- Good pseudorandom (PN) properties (large period length, uniform distribution, etc.).
- Low periodic correlation property (ACF, CCF).

L. C. Nguyen (✉) · V. K. Tran
Electric Power University, Hanoi, Vietnam
e-mail: cuongnl@epu.edu.vn

C. Q. Le
Hanoi Open University, Hanoi, Vietnam
e-mail: quynh.lechi@gmail.com

- Low aperiodic correlation property.
- Large linear complexity (LC).
- Large cardinality (number of sequences in the set).
- High-speed (up to GHz band)

Unfortunately, no sequence known so far can satisfy these demands completely. Therefore, depending on the application case, these properties need to be investigated carefully and make the right compromise. In [10] a method to generate the optimal PN-sequences with best ACF, balanced distribution and high LC is introduced. In this regard, we would like to mention that even the most recent publications only the length or the speed of the sequences are addressed while little attention has paid to the other statistic properties [11–17]. It is obviously not at all enough for the purpose. In this paper, we will try to fill this gap.

The paper is organized as follows: In the next section, Sect. 2, some related works will be reviewed. In Sect. 3, the mathematical tools and operations for investigating the interleaved structure are explained. In Sect. 4, the linear complexity is presented. In Sect. 5, some statistical properties of the sequences generated by LFSR based PN-generators will be presented. In Sect. 6, the hardware implementation in time multiplexing technique is shown. Sect. 7 is a conclusion and further work.

2 Related Works

Let start with [1] because it contains some typical wrong claims; now we would like to give our comments on related topics.

2.1 *Some Comments on the “Novel Complex PN-Code-Generator”*

The article is interesting since it addressed the “complex PN-code-generator-based data scrambler and descrambler”, which is widely used in compressive sensing signal processing, cryptography, network security, watermark, and steganography.

We appreciate the idea of using “a relatively low-length shift register for the generation of highly secure m -sequence generator, wherein the feedback tappings keep on changing in a pseudo-random manner, which makes the generated codes quite complex”.

The FPGA implementation of the complex PN-code generator is also given in detail, which makes the application aspect more attractive.

In fact, the reason for our attention to this article is: we have our-self done some works and published a few papers on this issue. We are now also looking for a solution to create PN with high complexity.

First, let us consider the following statement:

“The security of the encrypted data is a direct function of the number of stages of the shift register used to generate the key. This means, to increase data security, the number of stages of the shift register is to be increased, which leads to an increase in the complexity of the system in terms of power, space and cost”.

In our opinion, this statement is not quite correct. It is well-known that data security depends on the linear complexity (LC) of the key sequences. The LC in its turn depends not only on the number of stages of the shift register used to generate the key (sometimes it is called the degree of the feedback shift register FSR). The most important factor that affects the complexity of the key sequences is the non-linear structure of the sequence. The structure proposed in that article is time multiplexing with line multiplexer. (The term: 8-to-1 line multiplexer is used by [1]). That can be seen clearly in Waveforms obtained at various checkpoints of a complex code generator. In the literature, this construction is termed as an interleaved structure [2–5]. In a word the non-linear procedure can be explained like this: first, we decompose the linear interleaved sequences b_n with the composite length $L = NT$ to get T subsequences a_n of length N . Then we replace a by subsequences c_n of the same length (N) while keeping the interleaving order unchanged. The linearity complex will be increased and reaches the maximal value if c_n is the reciprocal version of a_n (they are timely reversed). The sequences introduce in [1] is obviously failed to meet the criteria specified above except for length and hight-speed. It is to expect that this configuration can support differential I/O standards at up to 1 Gbps when using dynamic phase alignment (DPA) and 840 Mbps when not using DPA. We can clearly prove that much have to be done to ensure the balance distribution (therefore ACF). According to [2, 3, 11], the relative phase between subsequences (short element sequences) have to satisfy some strict relations so that the resulting sequences (long sequences) have the best possible ACF. Furthermore, there must be null-subsequences in the interleaving so that the resulting sequences will be balanced since in each subsequence the number of “ones” exceeds the number of “zero” by one. So, if the resulting sequences in [1] consist of T subsequences then, the number “one” will exceed the number of “zero” by T and they are no mean balanced and having the best ACF. For details, see [2, 3, 10, 11]. In short: this PN-generator is, therefore, cannot meet the requirements of some new applications presented below.

2.2 Another New High-Speed PN-Generator

The new applications are highlighted in some recent publications [12–15], related to some very new topics: compressed sensing, physical layer security, where the high-speed PN-sequences at the frequencies 2.4–3.2 GHz or 4 GHz are needed. In [15] a frequency range of 28.5 GHz is mentioned. In [16] a PN-sequence generator at 20 GHz is implemented with IC circuits, FPGA and DSP and the LFSR of degree 9 and 12. It is well-known that the PN-sequences can also be generated by MATLAB

software. However, as it is pointed out in [14] that “The use of a programmable shift-register is strictly for testing purposes and could be replaced with an appropriate combination of much lower power consumption LFSRs”. In our opinion, when it comes to hardware implementation it is most straightforward to describe the sequence structure in a hardware-oriented manner (D-transform) [2, 3, 10]. Therefore, we need to review some useful mathematical tools related to the interleaved structure.

3 Mathematical Tools for Representation of Interleaved Structure

From an extensive literature survey we can see that the following mathematical tools can be employed:

- i. Algebraic structure: namely difference sets, finite field theory, trace function. These ways of expression are very compact and widely used. However, these representations give no hints to the hardware implementation. For the hardware implementation of interleaving (time multiplexing) of a larger number of subsequences exactly so that the resulting long sequences satisfy strict requirements on ACF, LC, and distribution we have to pay a lot of attention to time relations, relative phase relations between subsequences. In this regard, the hardware-oriented description (D-transform) is most desired. The difference set theory gives us a clear picture of ACF. The finite field theory, trace function are effective in the study of the structure and relations (mapping), between fields: extension field, base field. For details, please see [4, 5, 7–9].
- ii. The hardware-oriented or D-transform (time multiplexing): This method is intuitive and good for hardware implementation [2, 3, 10]. Therefore, we would like to pay special attention to this method.

3.1 *The Interleaved Structure and Methods for Representation and Analysis of LC of the Interleaved Sequences*

Bearing in mind that most of the useful sequences (linear as well as nonlinear) in the practice are having very great length L and except for few prime values (Mersenne prime), L is a composite number. Generally speaking all the sequences of composite length $L = T.N$, can be implemented by interleaving T subsequences, each of length N . For this reason, the interleaving technique is widely used for sequences generation [2–5, 7].

The interleaved structure is specified by subsequences $\{a_i\}$ and shift sequence (interleaving order I_p^T).

The subsequences $\{a_i\}$ can be determined by decomposition (decimation) of the sequence of composite length L into T subsequences. The shift sequence (or interleaving order I_p^T) determines the way the subsequences are interleaved. This construction will be shown later in related examples. It is important to know that the interleaved structure which gives rise to the sequences with the best possible ACF and large LC is most interesting and useful. As it will be shown later, with one interleaving order I_p^T many GMW sequences can be constructed by changing the subsequences (denoted as $\{e_i\}$).

As stated earlier, the non-linear structure is the deciding factor for the complexity (security) of the key sequences. Therefore more attention should be given to this issue.

3.1.1 The Nonlinear Operation to Generate Sequences with Larger LC

Nonlinear operation in D-transform

The D-transform method is stated by the following theorem:

Theorem 1 Let $\{b_i\}$ be the PN sequence of length $L = 2^n - 1$, $m|n$ and I_p^T and $\{a_i\}$ of length $N = 2^m - 1$ be the shift sequence and subsequence of $\{b_i\}$, respectively.

The sequence $\{c_i\}$ obtained by replacing $\{a_i\}$ with another subsequence $\{e_i\}$ of the same length N and two leveled ACF while keeping I_p^T unchanged will have the following properties:

- (i) The ACF $\theta(j)$ can be easily determined by: $\theta(j) = 1 - 2P_j(1) = -\frac{1}{L}, j \neq 0$.
- (ii) $L = T.N$ (interleaved structure).
- (iii) $L_1 = \frac{L+1}{2}$ (balanced).
- (iv) $\{c_n\}$ is no more linear.

Note that: (i) The nonlinearity of $\{c_i\}$ is achieved by replacing the subsequence $\{a_i\}$ of $\{b_i\}$ by subsequence $\{e_i\}$.

(ii) Maintaining the best ACF by keeping I_p^T (structure) unchanged. In fact, if the relative phase shift of subsequences in the interleaving is identical to the shift sequence I_p^T of an interleaved m -sequence, then the ACF is two leveled.

Nonlinear operation in Trace representation

According to [2, 10, 11], nonlinear interleaved sequences with ideal ACF (GMW sequences) are generated through 3 mapping processes:

- First mapping is from $\text{GF}(2^n)$ to $\text{GF}(2^m)$ which yields the I_p^T (shift sequence).
- Second mapping which is nonlinear: raise the elements of subfield $\text{GF}(2^m)$ to the power r with $\gcd(r, 2^m - 1) = 1$, $1 \leq r \leq 2^m - 1$.
- Third mapping is from the elements of subfield $\text{GF}(2^m)$ into $\text{GF}(2)$.

Two nonlinear operations in D-transform and Trace function are equivalent (giving the same LC value) [2, 3, 10].

Profile of linear complexity of interleaved sequences of length $2^n - 1$

The linear complexity of a periodic sequence is considered as a primary measure of its randomness and strength against the Berlekamp–Masey algorithm. Therefore, a lot of efforts have been made to generate sequences not only having good ACF but also large LC (nonlinear).

In the algebraic method, the LC of GMW sequence can be calculated either:

By the minimum number of terms in its trace function expression (by the sum of elements in $\text{GF}(2^n)$). For example, the GMW sequence of period 63 given by can be expanded as [2, 5, 10, 11]

$$g(t) = \text{Tr}^3\left(\{\text{Tr}_3^6(\alpha^t)\}^3\right)$$

where α is a primitive element of $\text{GF}(2^m)$, and r is any integer relatively prime to $2^m - 1$, $1 < r < 2^m - 1$, and $\text{LC} = 12$, since there are 12 terms in that expansion.

LC can also be determined based on the Hamming weight of the decimation r (which creates the non-linearity) in the expression [11]:

$$b_1 = \text{Tr}_1^m\left(\left[\text{Tr}_m^n(\alpha^i)\right]^r\right)$$

which result in $\text{LC} = m.l.w(i)$. With $l = \frac{n}{m}$, $w(i)$ is the Hamming weight of r (decimation).

In D-transform based approach, there are two methods to calculate LC, which are given as follows:

- i. DFT (Discrete Fourier D-transformation) can be used to calculate the LC of periodical sequences:
 $\text{LC} = w(\text{SN})$ where w is the Hamming weight of the DFT SN of the sequence $S(t)$. This is called Blahut theorem.
- ii. Euclid method: Before applying the Euclid algorithm for LC calculation two simple operations are introduced.

Time-division: In the time frame T , the consecutive bits of subsequences are separated by T time slots. In D-transform, this equivalent to the operation $Z_i(D) = Z_i(d^T)$.

Timeslot assignment: This operation is equivalent to multiply the subsequences by d^i therefore $b(D) = \sum_{i=0}^{T-1} d^i Z_i(d^T)$.

Accordingly, we have [2, 3, 10] $Z_i(d^T) = \frac{S_{ei}(d^T)}{G_{es}(d^T)}$,

And $C(D) = \sum_{i=0}^{T-1} \frac{d^i S_{ei}(d^T)}{G_{es}(d^T)}$, where $S_{ei}(d^T)$ and $G_{es}(d^T)$ represent the initial state and generating polynomial for $\{e_n\}$, $Z(D^T)$, $C(D)$ presents the D-transform of $\{e_n\}$, $\{c_n\}$ respectively.

The Euclid algorithm applied on $C(D)$ renders the least degree polynomial for $C(D)$ and the LC is thus obtained.

Table 1 LC values of interleaved sequences with different subsequences

| Order | GF(2 ¹²) | GF(2 ⁶) | | | | | |
|-------|----------------------|---------------------|---------|---------|---------|---------|---------|
| | | 1100001 | 1000011 | 1101101 | 1011011 | 1110011 | 1100111 |
| 1 | 1100101000001 | 12 | 192 | 48 | 48 | 24 | 96 |
| 2 | 1000001010011 | 192 | 12 | 48 | 48 | 96 | 24 |

4 LC of Proposed PN-Generator

4.1 *The Big Differences in LC Due to the Nonlinear Structure*

It will be seen in, Example 1 and 2 (Table 1).

For illustration, we have the following examples: for $f(x) = 1 + x + x^6$. The interleaving order in m -sequence is $I_p^T = \{\infty, 3, 6, 5, 5, 2, 3, 5, 3\}$ and the LC = 6.

```

0 0 0 0 0 1 0 0 0
0 1 1 0 0 0 1 0 1
0 0 1 1 1 1 0 1 0
0 0 1 1 1 0 0 1 0
0 1 0 1 1 0 1 1 1
0 1 1 0 0 1 1 0 1
0 1 0 1 1 1 1 1 1
0 1 0 1 1 0 1 1 1
0 1 0 0 0 1 1 0 1

```

Note 1 The number of stages in FSR remains the same: 6, but the LC can be significantly improved if the subsequences are replaced by Theie reciprocal (12, double) [10].

Example 2 Some interleaved sequences of greater length.

The first column is the order number.

The second column lists the primitive polynomials of degree $n = l.m$, corresponds to the sequences of the interleaved structure.

The next columns list the polynomials of degree m of the subsequences.

For example, $(11101111)_2 \Leftrightarrow g(D) = 1 + D + D^2 + D^4 + D^5 + D^6 + D^7$.

The binary values 1 s represent the feedback taps of the LFSR.

Note 2 The degree of the FSR: $n = 12$, the initial LC = 12, the subsequence is generated by an FSR of degree $m = 6$. If the subsequences are replaced by its reciprocals, the LC_{max} of the interleaved sequence = $6 \cdot 2^5 = 192$ —a big difference. The PN sequences with very large LC and best possible ACF are called optimal sequences [10].

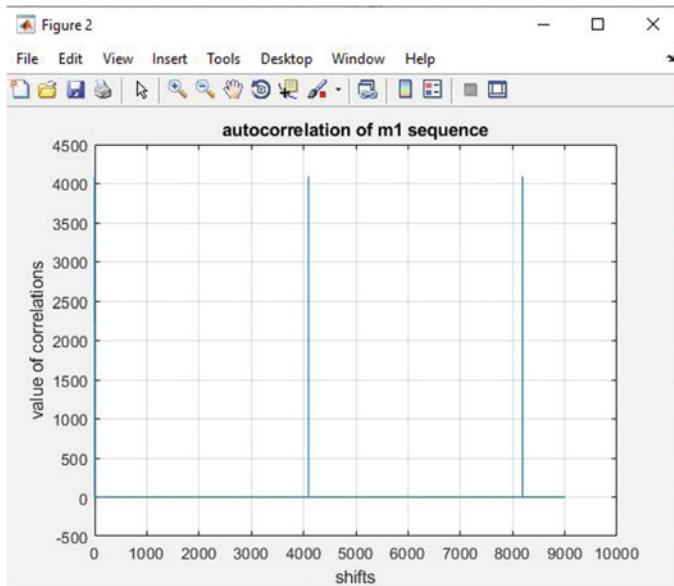


Fig. 1 ACF diagram of the sequences specified by length $L = 64$

- The theoretical background and examples show that the statement-making the starting point for the proposal [1] is not quite correct.

5 The Statistic Properties of the Interleaved Sequences

For applications related to cryptography, watermarking, steganography, etc. some requirements such as distribution, run, ACF, spectrum, etc. of the sequences much be carefully considered.

Example 3 In this example, a Matlab simulation is used to explore the AFC and spectrum of a typical sequence: $g(D) = 1 + D^5 + D^6$.

See Figs. 1, 2 for the AFC properties of the sequence:

It can also be seen that while ACF is identical for any interleaving, the distribution of runs is almost random and the LC shows a great difference.

6 Hardware Implementation

The hardware implementation simplicity of D-transform is one of the most desired properties. We will show here the simplified hardware configuration for shifted versions for interleaving. The ALTERA AC601 v1.2 FPGA tool kit with FPGA

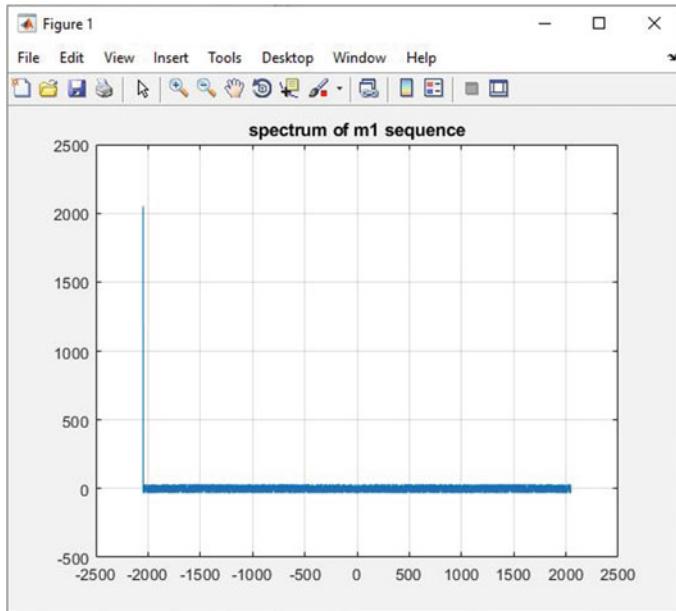


Fig. 2 The spectrum of the nonlinear interleaved sequence of length $L = 4096$ [2]

Cyclone IV chip is used to simulate the creation of a PN sequence application in AIC. PN sequence is created on Matlab software and saved as a file format with bit patterns. This data file is loaded into the ROM memory of FPGA. Each high-speed PN sequence is used for sampling created using a continuous mechanical type switch through the FPGA output pins (Fig. 3).

The output chain is formed after scanning the ring on the FPGA pins (Figs. 4, 5).

We can see that a hardware-oriented manner of D-transform is straightforward and can be implemented in a simple programmable gate array. The obtained Waveforms can be seen clearly at various checkpoints of a complex code generator.

Fig. 3 Scheme of the implemented AIC system



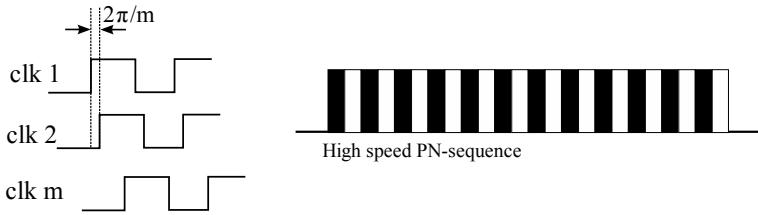


Fig. 4 Output pulse diagram after switching

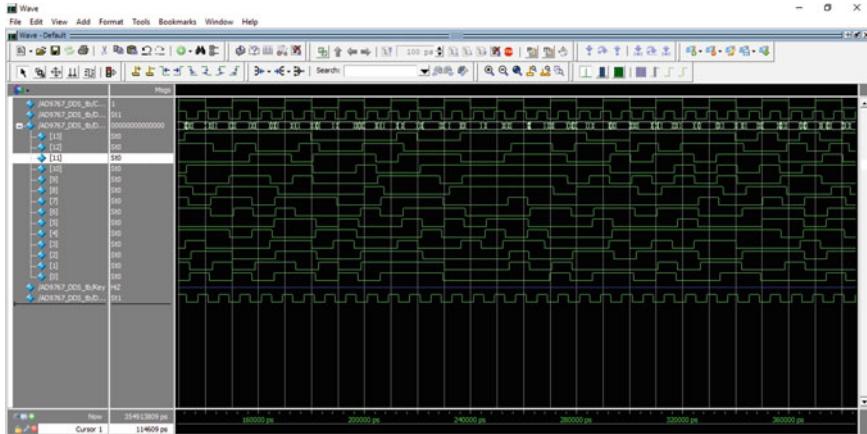


Fig. 5 Timing diagram after simulation on ModelSim software

7 Conclusion and Further Works

In this paper, we discussed the properties and applications of high-speed PN-sequences and reviewed the mathematical tools and methods used.

Taking into account the dangerous and complicated situation in the IoT world, the above-mentioned topic (high-speed PN sequences) is still very much actual and that kind of sequences will be more widely used for the following:

- Scrambling and descrambling in network security [1–5, 11]
- Generating the key sequences in cryptography [2, 11, 14]
- Generating the random clock for compressive sampling [12–17]
- Embedding, masking (scrambling) and extracting the signal in watermark and steganography [18, 19].

The in-depth study of these sequences is therefore necessary and interesting. We want to have some contributions to that issue.

In future work, we would like to give some suggestions:

- Select the best-interleaved sequences regarding LC, ACF and run distribution

- Work out the hardware structure of the above sequences and FPGA implementation
- Investigate the efficiency of these sequences in some applications like network security, cryptography, compressive sensing, watermark, steganography.

References

1. G.M. Bhat et al., Field programmable gate array (FPGA) implementation of novel complex PN-code-generator-based data scrambler and descrambler. *Maejo Int. J. Sci. Technol.* **4**(01), 125–135 (2010)
2. Q. Le Chi et al., A hardware oriented method to generate and evaluate nonlinear interleaved sequences with desired properties. *J. Inform. Eng. Appl.* **6**(7), 1–12 (2016)
3. L.M. Hieu, L.C. Quynh, Design and analysis of sequences with interleaved structure by d-transform. *IETE J. Res.* **51**(1), 61–67 (2005)
4. H.J. Zepernick, *Pseudo Random Signal Processing Theory and Application* (Wiley, 2005)
5. J. He, *Interleaved Sequences Over Finite Field*, Ph.D. Thesis Carleton University Ottawa, Ontario, 2013
6. J.M. Velazquez-Gutierrez et al, Sequence sets in wireless communication systems: a survey. *IEEE Commun. Surv. Tutorials* **19**(2), 1225–1248 (2017)
7. V. Edemskiy, On the linear complexity of interleaved binary sequences of period 4p obtained from Hall sequences or Legendre and Hall sequences. *Electron. Lett.* **50**(8), 604–605 (2014)
8. C. Ding, *Codes From Difference Sets* (World Sci, Hackensack, NJ, USA, 2015)
9. P. Xia, S. Zhou, G.B. Giannakis, Achieving the Welch bound with difference sets. *IEEE Trans. Inf. Theor.* **51**(5), 1900–1907 (2005)
10. Q.L. Chi, K.T. Vu et al, *FPGA Implementation of Optimal PN Sequences by Time_Multiplexing Technique*. Springer Nature Switzerland AG 2020 K.-U. (Eds.): ICERA 2019, LNNS 104, pp. 373–380 (2020)
11. S.W. Golomb, G. Gong, *Signal Design for Good Correlation—For Wireless Communication, Cryptography, and Radar*
12. L. Wang et al, A random sequence generation method for random demodulation based compressive sampling system. *Int. J. Signal Process. Image Process. Pattern Recogn.* **8**(1), 105–114 (2015)
13. N. Wang et al, Physical-layer security in internet of things based on compressed sensing and frequency selection. *IET Commun.* **11**(9), 1431–1437 (2017)
14. J. Yoo et al, *Design and Implementation of a Fully Integrated Compressed—Sensing Signal Acquisition System* (California Institute of Technology, 2012), pp. 1–4
15. W. Liu et al, Researches on the wideband spectrum sensing prototype system based on MWC. *Int. J. Sig. Process. Syst.* **5**(2), 70–74 (2017)
16. X. Chen, *Sub-Nyquist Rate Sampling Data Acquisition Systems Based on Compressive Sensing*, Ph.D. Dissertation Texas A&M University, May 2011
17. J.N. Laska et al., *Theory and Implementation of an Analog-to-Information Converter using Random Demodulation* (Rice University, Houston, Texas, 2014)
18. J. Kufel et al, Sequence-aware watermark design for soft IP embedded processors. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 1–14 (2015)
19. M. Li, PN-sequence masked spread-spectrum data embedding, in *IEEE/CIC ICCC 2015 Symposium on Privacy and Security in Communications*, pp 1–6

Syngas Assessment from Plastic Waste Using Artificial Neural Network—A Review



Maulik A. Modi and Tushar M. Patel

Abstract On our planet, pollution is one of the major issues for humankind. Pollutants like, pollution of land, pollution of water, pollution of air, pollution of noise, and many other types of pollution are directly or indirectly affect on humankind. Due to this pollution, our planet got global warming effect, climate change, and also very hazardous effect on human body due to air pollution, and also the issue of waste product dumping that creates the land pollution. Plastic plays one of the most important roles for land pollution. Plastic waste which does not decompose naturally in the environment causes pollution. Plastic also affects on wild life, in land due to plastic pollution, rainwater not able to go at the desired level, and plastic also affects the farming, due to plastic pollution in land. Crops are not able to get the desired nutrition from land. The survey from the literature available found that with the help of gasification and pyrolysis process can change waste plastic as a transportation fuel and also solve the problem of dumping of plastic waste. The quality of syngas is using the ANN model. The production of syngas, and gasification techniques was widely used, and for the quality of improvement of syngas, various techniques reported in the literature review.

Keywords Gasification process · Plastic waste · Biomass (sugarcane bagasse · Groundnut shell · Coconut shell · Rice husk) · Garbage of kitchen · Paper · Textile · Nanomaterial (potassium alumina · Copper · Nickel) · ANN

1 Introduction

Motivation: I have seen the cow eating the waste plastic with waste food, and this plastic goes in its stomach and will create diseases like cancer, and from this incident,

M. A. Modi (✉) · T. M. Patel
Mechanical Engineering Department, KSV, Gandhinagar, India
e-mail: maulikmodi325@gmail.com

T. M. Patel
e-mail: tushar.modasa@gmail.com

I have decided that there must be a solution of this problem, and here two types of solution: first one arrangement of awareness program in people does not throw waste food with plastic and second find the methods how to utilize plastic waste as a source of energy. The unique thing in work is using nanomaterial quality improvement in syngas from plastic waste. In 1907, plastic was introduced and becomes more popular material for the development of new era of material in the world. Invention of Parkesine in 1856 is the first man-made plastic; plastics play an important role for the development of society [1]. Plastic is having superior quality like strength, light in weight, longevity, economical, resistance of water, and resistance of erosion. Plastic becomes an integral part of human life in twenty-first century. From the morning to go in bed at night, we use hundreds of plastic material in daily routine life [2]. Due to this much use of plastic in daily life, we found plastic from top of the Everest to the bottom of sea and create pollution of land, pollution of air, pollution of water, and also many other pollution [3].

2 Plastics Global Production and Statistics

Plastic word is derived from Plastikos from Greek and with meaning ‘Better for Molding.’ Organic synthetic and with large molecular weight’s plastic manufactured. Base of the plastic is coming from petroleum industry. Composition of plastic contains long chain of carbon and hydrogen and classified into thermoplastic and thermosetting plastic based on their basic changes in structure as per temperature changes. Plastics like polyvinyl chloride (PVC), PP, polystyrene (PS), HDPE, and LDPE were not able to any change in chemical composition by changes in temperature. In Fig. 1, seven plastics resin code included and from that one can choose

|  1 PETE |  2 HDPE |  3 PVC |  4 LDPE |  5 PP |  6 PS |  7 OTHER |
|---|--|---|--|--|---|---|
| polyethylene terephthalate soft drink bottles, mineral water, fruit juice container, cooking oil | high-density polyethylene milk jugs, cleaning agents, laundry detergents, bleaching agents, shampoo bottles, washing and shower soaps | polyvinyl chloride trays for sweets, fruit, plastic packing (bubble foil) and food foils to wrap the foodstuff | low-density polyethylene crushed bottles, shopping bags, highly-resistant sacks and most of the wrappings | polypropylene furniture, consumers, luggage, toys as well as bumpers, lining and external borders of the cars | polystyrene toys, hard packing, refrigerator trays, cosmetic bags, costume jewellery, CD cases, vending cups | other plastics, including acrylic, polycarbonate, polyactic fibers, nylon, fiberglass |

Fig. 1 Resin identification code for various plastics given by the society of plastic industries

the plastic which is recyclable. 25% PET plastic bottles are recycled. 30–35% of HDPE plastic recycled, 1% of PVC material is recycled. Products made using recycled LDPE are not as hard. Products made using LDPE plastic are reusable, but not always recyclable. Recycling is not widely available for polystyrene products [4]. Plastic pyrolysis oil 75 and 25% diesel oil were mixed, and experiment was carried out on diesel engine. From the experimental work, performance and exhaust parameter were analyzed. Failure of the engine is due to the parameters like lubricant, parts of engine and piston wear and tear; as a result it was found in the experimental work that 75% of plastic pyrolysis oil not used as an alternative because after 36 h running of engine, engine failed for further experiment [5].

3 Effects of Plastic Waste on Environment

Plastics waste pollutes groundwater, freshwater and marine environments; plastic clogging creates the disturbance in urban drainage system to choke up the drainage line and increasing the diseases like malaria in monsoon due to flooding of water [6].

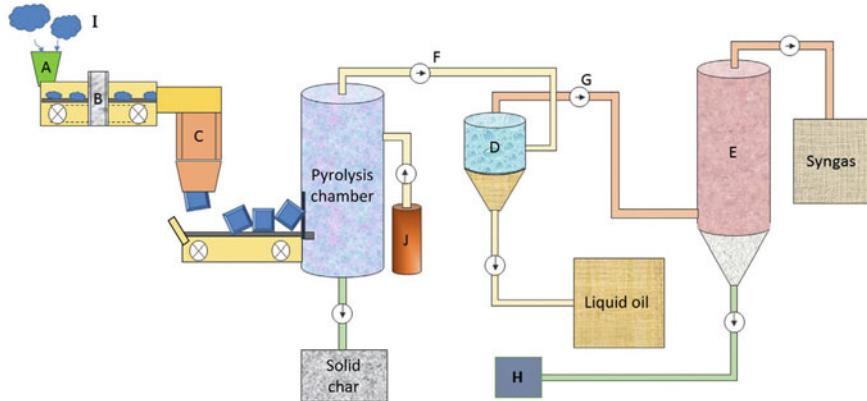
4 Mechanical Recycling

Mechanical recycling is a process which converts scrap of material into useful product with approximately similar properties of parent material. As an example, all types of thermoplastic are manufactured using mechanical recycling [7]. Plastic recycling is also known as re-extrusion process [8] (Fig. 2).

Producer gas derived from pyrolysis process is one of the best alternative sources of energy with higher calorific value. So producer gas becomes one more option for energy source. Pyrolysis of plastic is classified in two ways: catalytic and thermal. Conversion of waste plastic into liquid fuel can be possible using catalytic pyrolysis, and liquid fuel becomes the best alternative source to generate electricity and fuel is also used as a transportation industry [9].

5 Effect of Nanomaterial on Gasification

For the production of hydrogen, gasification of water for sugarcane bagasse is used with catalyst. As a catalyst, they used 2.5% potassium alumina and 20% copper found the best catalyst for the production of hydrogen in the medium of supercritical water [10]. Nickel as a nanomaterial is used in experimental work. Gases were derived from biochar, and rice husk is helpful as a source of energy. Structural size of nanomaterial also plays a very important role in experimental work. Nickel particle worth of 0.1 gm is taken for experimental work for the distillation of pyrolytic from rice husk and



A, Feeder; B, drying unit; C, compacting unit; D, condenser unit; E, air pollution control unit; F, gaseous emissions; G, Noncondensable gas; H, Dust and particulate matter; I, raw feed; J, nitrogen purging

Fig. 2 Plastic waste pyrolysis process

result derived from experiment as 46.67% of char, 20% of oil, and 33.3% of producer gas in pyrolysis process with catalyst at 400 °C [11].

6 Artificial Neural Network in Gasification

George et al., for experimental work, were performed on bubbling fluidized-bed gasifier which developed artificial neural network. They used raw material as a biomass like coffee husk, sugarcane bagasse, groundnut shell, and coconut shell in the temperature range of 650–800 °C with equivalence ratio in between 0.23 and 0.34. The value of regression coefficient and mean square error is 0.987 and 0.71, respectively, for the above work which is found in desired limit [12]. Kadir et al., in their experimental work, got hydrogen-enriched gas using ANN. The result is obtained by ANN modeling required systematic logical perspective. The comparative analysis is done through which they obtained good output result for each input variable [13]. Baruah et al. formulated the concentration of gases such as percentage of methane, percentage of carbon dioxide, percentage of carbon monoxide, and percentage of hydrogen. Parameter used as input ash content, C, H, O content, temperature of reduction zone and moisture content. The ANN model consists of one layer of parameter as output, input, and hidden. In methane and carbon monoxide, ANN model gives a good result with experimental data in terms of fraction of variance which is greater than 0.99 and fraction of variance which is greater than 0.98 in carbon dioxide and hydrogen model [14]. Shankat et al. used ANN's multilayer module for the prediction of LHV for gas products with tar, char, and yield of syngas from MSW in FBR gasifier. All these variables were having a better effect for the artificial neural network. After

doing experiment, they find in between 8 and 29% of biomass composition and have better result for the production of producer gas. Temperature reduction was important variable for the quality improvement in producer gas (carbon monoxide and hydrogen) [15]. Xiao et al. used organic compounds like garbage of kitchen, paper, textile, waste plastic, and Municipal solid waste with gasification temperature of 400–800 °C and 0.2–0.6 equivalence ratio. Gasification products, low heating value of syngas and yield of gas were output parameters. Gasification characteristics are different in all samples. Artificial neural network used for the prediction of gas characteristics, train the module, and validating with ± 15 and $\pm 20\%$ relative error. Industrial sample having $\pm 25\%$ relative error is noted. From the result concluded that experiment shows good comparison of gas characteristics with artificial neural network model [16]. Shahbaz et al. used ANN modeling for gasification of steam in palm kernel shell with Cao as an adsorbent. Input parameters were like steam/biomass ratio at a constant heat, weight of bottom ash from coal, constant biomass, steam and CaO proportion. As an output parameter, they take ingredients of hydrogen, carbon monoxide, carbon dioxide and methane, syngas yield, LHV and HHV. Seven hidden layers of backpropagation algorithm were used [17]. Maria et al. used two methods, CFB and BFB, for determining composition of gas (carbon monoxide, carbon dioxide, oxygen, hydrogen, methane, and yield of producer gas). In CFB Biomass Composition (C, H, O) at a rate of 31.7 and 54.1 % of CO₂, Carbon Monoxide, CH₄, Hydrogen. While in BFB 28.9 and 52.3 % yield of producer gas. In CFB, 37.6 % input equivalence ratio and ER decreased in BFB upto 10.8 % [18].

7 Conclusion

From the literature review, fuel gas behavior of biomass depends upon the following parameters like temperature, biomass composition, equivalence ratio, and S/B ratio. Gas generated from system depends upon the temperature of the gasification process. The temperature between 650 and 750 °C is the most rewarding range because, in this temperature range, the gas yield is reaching its maximum level. Catalytic reaction and higher ash content in the biomass influence on the weight of liquid volatiles and increase the gas composition. The literature review may be summarized as follows: In steam gasification, the optimum value for the generation of hydrogen was reported 55.97% by volume for larch wood, 55.5% by volume for almond shell, and 53.08% by volume for rice husk. Carbon conversion efficiency can also be considerably increased in steam gasification. The optimum value is found to be 96.95% for larch wood biomass fuels. The agricultural residues had the potential to give energy in the future. There are also some research papers on the effect of nanomaterial on gasification for the production of clean and highly enriched gases.

References

1. Plastics Europe, *Plastics—The Facts 2014/2015: An Analysis of European Plastics Production, Demand and Waste Data* (Plastics Europe, 2015), pp. 1–34
2. V.E. Yarsley, E.G. Couzens, *Plastics in the Modern World* (Penguin, Baltimore, MD, 1945)
3. R.C. Thompson, C.J. Moore, F.S. vom Saal, S.H. Swan, Plastics, the environment and human health: current consensus and future trends. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **364**, 2153–2166 (2009)
4. A.L. Andrade, M.A. Neal, Applications and societal benefits of plastics. *Philos. Trans. R. Soc. B Biol. Sci.* **364**, 1977–1984 (2009). www.learn.eartheasy.com
5. I. Kalargaris, G. Tian, S. Gu, Investigation on the long-term effects of plastic pyrolysis oil usage in a diesel engine. *Energ. Procedia* **142**, 49–54 (2017)
6. J.J. Adibi, F.P. Perera, W. Jedrychowski, D.E. Camann, D. Barr, R. Jacek et al., Prenatal exposures to phthalates among women in New York and Krakow, Poland. *Environ. Health Perspect.* **111**, 1719–1722 (2003)
7. S.M. Al-Salem, P. Lettieri, J. Baeyens, Recycling and recovery routes of plastic solid waste (PSW): a review. *Waste Manag.* **29**, 2625–2643 (2009)
8. M. Sadat-Shojaei, G.R. Bakhshandeh, Recycling of PVC wastes. *Polym. Degrad. Stab.* **96**, 404–415 (2011)
9. S.R. Chandrasekaran, B. Kunwar, B.R. Moser, N. Rajagopalan, B.K. Sharma, Catalytic thermal cracking of postconsumer waste plastics to fuels. 1. Kinetics and optimization. *Energ. Fuels* **29**, 6068–6077 (2015)
10. A. Tavasoli et al., Sugarcane bagasse supercritical water gasification in presence of potassium promoted copper nano-catalysts supported on g-Al₂O₃. *Int. J. Hydrogen Energ.* (2015). <https://doi.org/10.1016/j.ijhydene.2015.09.026>
11. R.S.S. Prabhahar, P. Nagaraj, K. Jeyasubramanian, Enhanced recovery of H₂ gas from rice husk and its char enabled with nano catalytic pyrolysis/gasification. *Microchem. J.* <https://doi.org/10.1016/j.microc.2019.02.024>
12. J. George, P. Arun, C. Muraleedharan, Assessment of producer gas composition in air gasification of biomass using artificial neural network model. *Int. J. Hydrogen Energ.* **43**, 9558–9568 (2018)
13. A. Karaci, A. Caglar, B. Aydinli, S. Pekol, The pyrolysis process verification of hydrogen rich gas (HerG) production by artificial neural network (ANN). *Int. J. Hydrogen Energ.* **41**, 4570–4578 (2016)
14. D. Baruah, D.C. Baruah, M.K. Hazarika, Artificial neural network based modeling of biomass gasification in fixed bed downdraft gasifiers. *Biomass Bioenerg.* **98**, 264–271 (2017)
15. D.S. Pandey, S. Das, I. Pan, J.J. Leahy, W. Kwapinski, Artificial neural network based modelling approach for municipal solid waste gasification in a fluidized bed reactor. *Waste Manag.* (2016)
16. G. Xiao, M. Ni, Y. Chi, B. Jin, R. Xiao, Z. Zhong, Y. Huang, Gasification characteristics of MSW and an ANN prediction model. *Waste Manag.* (2009)
17. M. Shahbaz, S.A. Taqvi, A.C.M. Loy, A. Inayat, F. Uddin, A. Bokhari, S.R. Naqvi, Artificial neural network approach for the steam gasification of palm oil waste using bottom ash and CaO. *Renew. Energ.* (2018)
18. M. Puig-Arnavat, J. Alfredo Hernández, J.C. Bruno, A. Coronas, Artificial neural network models for biomass gasification in fluidized bed gasifiers. *Biomass Bioenerg.* **49**, 279–289 (2013)

Applications of Data Mining in Predicting Stock Values



Aparna Raghunath and A. R. Abdul Rajak

Abstract This is a study paper on the various applications of data mining in stock market. Different methods have been taken into consideration that can solve the problem. Stock market prediction is a complicated task due to its extremely random and unpredicted nature. With a large number of people investing in stocks, it is important that we develop models that can predict their nature. Data mining is used to solve such predictive problems. Various techniques have been discussed, and corresponding models were implemented on the provided data to generate results. The results from all models are compared to give the best fitting model.

Keywords Data mining · Stock market · Machine learning · Support vector machine · R language · KNN · Random forest · Regression · Sentiment analysis

1 Introduction

Stock market is a platform where buyers and seller interact in a digital form. Many people invest in stock as a supplement to their annual income. Investing in stock markets has become so popular due to the high profits that one can make. At the same time, a small error or misjudgment can lead to unexpected loss.

Many researches have been conducted in the field of predicting stock markets. However, this is not an easy task as stock values are determined by various factors. A country's political state and economical market are a few to mention. For example, during political elections, stock values change drastically. This change is entirely random and can either lead to profits or enormous loss.

A. Raghunath · A. R. Abdul Rajak (✉)
Birla Institute of Technology Science Pilani, Dubai Campus, Dubai, United Arab Emirates
e-mail: abdulrazak@dubai.bits-pilani.ac.in

A. Raghunath
e-mail: aparnaraghunathuae@gmail.com

Data mining is a non-trivial extraction of implicit, previously unknown and potentially useful information from data. It has two tasks namely prediction and description. These methods are implemented along with machine learning, statistics and database systems to generate results. This knowledge is then used to derive patterns that can help in solving predictive measures.

The process involves preprocessing of data, data mining and verification of result. There are many data mining tools available among which classification, regression and clustering are the most common ones.

Stock market is one of the applications of data mining tools along with machine learning and artificial intelligence. These tools are used to make predictions on the future trends of stock values. This prediction is done by assessing the past values and deriving patterns from them. The accuracy of such models depends on the method used. The results from the models can be used to predict the behavior of stocks.

2 Literature Survey

Stock market has a nature that is volatile and cannot be predicted easily. In order to make better choices while buying or selling the stocks, a few models have been developed which can deduce patterns and hence give reliable predictions. [1] discusses about two models namely random forest and support vector machine. Both these models are widely used for classification, a data mining technique. The dataset consisted of data collected over the years 2000–2016. Top twenty-five headlines for a day are arranged with a label ‘0’ if the stock price declines or remains unchanged and a label ‘1’ if the price increases. The use of ‘Bag of Words’ model counts the occurrences of words in the training set. N-gram model or a skip-gram model is used along with ‘Bag of Words’ to store the words in order. Vectors from these models are used for training the dataset. Random forest, a classification algorithm, creates multiple trees where each tree generates a result. Higher the number of trees, higher is the accuracy of the generated result. If random forest is considered as a regression algorithm, then the result is the mean of the generated results from each tree. If there are n classes in the dataset, subsamples are chosen randomly with replacement, and trees are created for each subsample. Another model discussed in [1] is support vector machine in which the data plotted on a plane are divided into classes. The plane dividing the dataset into better points is chosen as the correct one and is called the hyperplane. Other method used is radial basis function (RBF) which creates complex regions for classes. The data were tested with three models namely random forest, linear support vector machine and nonlinear support vector machine. The accuracy rate of each indicates that random forest is the best suited model for stock market prediction. However, it also indicates that support vector machine is preferred if the data are time dependent. Therefore, the model to be chosen depends on the parameter available.

Recent years have seen a growth in the number of people investing huge amounts in stock market. There is an equal probability of suffering a loss as gaining a profit. It

is important that we make the right decision at the right time. However, this is easier said than being practical due to the volatile and unpredictable nature of stock market. Gupta et al. [2] aim to develop a model using data mining algorithms such as KNN algorithm along with machine learning tools like genetic algorithm and SVR together with sentiment analysis. This model is expected to provide reliable predictions on the nature of stocks. The system was developed with the help of two models, primary and secondary. While primary model focuses on the prediction, secondary model has its focus on sentiment analysis. The dataset for the system is chosen to be six stocks under the banking sector. Since this dataset can have various missing values and redundancy, logistic regression is applied as the preprocessing tool. Three different algorithms were deployed for predicting the stock values. The first algorithm used is K nearest neighbors (KNN) which is primarily used for classification and can be extended to regression. The dataset has four attributes namely open, high, close and low. This data are then divided into training and testing sets in the ratio 70:30. The value of k is calculated as the \sqrt{t} (total data points in training set). The fit and predict methods are used to train and extrapolate the data, respectively. The accuracy of this algorithm is measured using accuracy score technique. For obtaining better predictions, genetic algorithm is used. As the first step, a chromosome of length five is defined. The components are taken as minimum, maximum, previous minimum, previous maximum and the predicted value. It uses open and closing prices as the two important parameters. An initial set of 500 data is taken on which min and max calculations are performed, and these values are compared to the next day's change. The data then flow through selection, crossover and finally mutations through the predictions of best score. The results from this algorithm can be complemented by SVR which checks the reliability of the output. As the stock values are highly affected by various factors, sentiment analysis is combined to the three algorithms to get better results. In this paper, Twitter feeds are chosen to be the source of news which are analyzed for keywords that can attain movements in the system. Having used different algorithms, it is observed that a better analysis can be obtained when these algorithms are implemented complement to each other. Keeping in consideration the extremely volatile nature of stock market, the developed model is considered reliable as the accuracy of predicted values is around 70–75%.

Kumar and Bala [3] talk about how accurate predictions of the stock market are important to the investors. Machine learning, a developing trend of data mining, is given the prime focus. Decision tree, random forest and linear model are a few algorithms that are used for predictive data mining with random forest having the highest accuracy. Results from these are compared to classification parameters such as H, FPR and TPR. The dataset for this study consisted of 57,772 entries with twenty-one features. One entry has been chosen as the target. Being a binary classification, the output can take values 0 or 1. The data are preprocessed to .csv format with no null values and with a minimum noise. In the next step, the data are divided into testing and training sets in the ratio 30:70. Classification techniques are applied on the dataset. Evaluation criteria like GINI or error rate generate the result. The positive rates are calculated using the confusion matrix for actual and predicted values. R model has been used to create graphical results as it gives the best representation.

The objective of [3] was to solve binary classification data using various machine learning techniques. The results show that random forest is the best model having the highest accuracy rate followed by linear model and decision tree. The model can be extended to large data set with target values represented in binary.

Today, stock market prediction is an area of extensive research and study. The entirely random relationship between its input and output makes it a complex task. A good predictive technique should be less time consuming, adaptable and exact. Iyer and Mehra [4] talk about the different data mining trends available for such predictive tasks like fuzzy systems, ANN, Bayesian rules, etc. The results from this study will reflect light on which method is the best suited for forecasting stock market. Being a survey study, they have collected twelve study papers that discuss various techniques to predict stock market behavior. Each method is identified to have its own specific dataset type which should be followed while executing that particular model. Later, advantages and disadvantages for each tool are studied. This information is converted to a tabular form to conduct a comparative study on which method is the best suited for forecasting stock market. The methods focused in this paper are support vector machine, clustering algorithms, neural network, sentiment analysis, decision tree, BPNN, etc. The ease of prediction, adaptability and accuracy are compared along with their advantages and disadvantages to generate the result. The result of the survey showed that each model has its own advantages and disadvantages. Any person or company willing to know the future stock behavior can make use of any models depending on the framework. It shows that it is possible to make predictions by using new hybrid methods or existing methodologies. There are various factors to be considered before implementing a model.

Prediction of the stock market is of immense importance to the stock business. Its nonlinear nature makes it difficult to make accurate predictions on the behavior of stock market. Fundamental techniques have failed to provide the best result. Regression is the most commonly used model for making predictions. Sharma et al. [5] study various regression approaches to understand how it is better than other technical ways. It also talks about the scope of developing on multiple regression. Sharma et al. [5] take into consideration four forms of regression namely polynomial, RBF, linear and sigmoid. Polynomial regression is used to depict nonlinear problems. However, it also fits linear regression as a statistical estimation. In regression, value of a dependent variable is predicted in terms of an independent variable. In radial basis function, the values depend on the distance from the origin or a center. The distance is calculated as the Euclidean distance. The value for a function is estimated from the sums of the RBFs. A sigmoid regression has a graph of the shape ‘S’. Linear regression deduces relationship between a dependent and an independent variable. It focuses on conditional probability. It is used to develop a model on a dataset with values of y for corresponding values of x . After such a model is developed, for an input value of x with no value for y , the model can be used to predict the value of y . The point of the research is to assist the investors and brokers for contributing cash in the stock market. The prediction plays an awfully important role in stock showcase trade which is very complicated and challenging process because of the extremely dynamic and random nature of the stock.

Many people invest in stock market as a supplement to their income. This, if done correctly, can get you nearly twice your wage, but at the same time, a small misjudgment can get you huge losses. Data mining, in collaboration with machine learning and AI, is put in use to predict the movement of stocks. Mankar et al. [6] focus mainly on two things, sentiment analysis and closing values. Sentiment analysis, here, is applied on Twitter tweets. Using these techniques, a model is developed that can predict stock market behavior which will help investors. The data collection from tweets can be done by streaming API or search API, where API is provided by Twitter. In [6], the search API has been used which allows specific requests from users. These tweets are then split by space and separated into words which will be used as features to train the dataset. As the next step, stopping words are removed from the list using Python's Natural Language Toolkit. After preprocessing the data, classifiers are applied to it. Two classifiers namely Naïve Bayes and support vector machine have been discussed. After choosing the top features, they are used to train the data for sentiment analysis. On implementing this on testing set, tweet sentiment is received as the result. This is then applied to predict the stock values. The stock data consists of past stock values. The tweet data and stock data are collected for the same timeline, and a correlation is drawn between the two which helps us to predict the stock values using tweet feeds. The study has concluded that support vector machine is the best suited model due to its feasibility and efficiency. Machine learning was incorporated to obtain data as it is inexpensive compared to other methods. Using Cloud services enabled storing of large datasets. The obtained Twitter tweets were classified into three classes namely positive, negative and neutral. This is an indication of public mood.

People can earn profits in financial markets, where sellers and buyers make transactions over ownership claims. However, these profits are purely conditional, based on the market price of stocks. Singh and Sharma [7] aim to create a model to predict the movement of stocks. The market is treated as a mathematical problem on which various algorithms can be applied to generate the result. This model predicts the value based on the past stock values. Such a model can help people in making right choices in order to maximize their profits. The data for the study were collected from numerai. Two main tools of data mining namely clustering and regression have been used to generate results. Under clustering, three different models, depending on the requirement and the availability of software and hardware, produced different datasets. In the partitioning model, each object is identified with a cluster. Larger datasets will be further divided hence creating hierarchy. The second model, hierarchical agglomerative method, creates clusters on different hierarchies, and trees are developed based on cluster nodes. It is a bottom up decomposition where clusters are grouped till a stop condition is met. Model-based method, the third clustering method, is based on probability. Regression can be linear or multivariate depending on the number on variables determining the output. In this study, multivariate regression is used in the form of partial least squares regression (PLSR). The tool used in this model development is R. In [7], the model was developed using PLSR algorithm. The results indicate the efficiency of this model with room for improvement. Having implemented various algorithms, it is seen that each model has its own advantages

and disadvantages. Therefore, there is a scope for developing new models which has features incorporated from the existing ones.

Stock market prediction is the process of evaluating the previous data and trying to find the stock behavior in the future. This is very important as many people have started using stock market to increase their earnings. Stock market is where transactions between buyers and sellers occur in the form of stocks, which can be digital or in physical form. Data mining deals with studying huge datasets and deducing useful knowledge which were previously unknown. In [8], an important data mining tool based on a decision tree called CTree is used to predict stock trends. The data are collected from numerai. Global null hypothesis is used to check the independent relation between the response and input values. Binary recursive portioning is required for implementing CTree. There are four different methods used in this paper. First method is censored regression in which the dependent value is unknown, but values of independent variables are known. Maximum likelihood estimation is used to estimate the model. Multivariate regression involves modeling multiple dependent variables with a definite set of predictors. This is different from the multivariable model which pertains to one dependent variable and multiple independent variables. Ordinal regression is used for predicting variables whose significance is subjective. It is called the ranking model in machine learning. Recursive binary partitioning explains the resultant variable of conditional distribution. There is a restriction on the portioning of the data. A generic algorithm can be developed using weighted numeric values. These models are created and implemented on *R*. Extensibility and the availability of wide statistical techniques make *R* an important tool. The CTree algorithm is proved to be an efficient method for the predictive tasks with an accuracy rate around 81%. However, it is seen that every model has its own properties, and therefore, there lies scope for developing new models. One can also use these models in collaboration to get better results depending on the availability and requirement.

Stock market prediction is a complicated task. However, a reasonable prediction would make a huge difference in the overall profit scale. Various researches try to address this by applying data mining techniques to deduce patterns. In [9], logistic regression has been used to analyze important and relevant ratios. It focuses on classification as the prime data mining tool. Logistic regression model is an extended version of linear regression where a single variable determines the output value. It forms an important component of machine learning in which the unknown data are computed using past data. Linear and logistic regression differ in the parameters chosen and the assumptions made. The dataset consists of top ten companies in Indian stock. They are classified as ‘good’ or ‘poor’ choice. In this paper, this classification is done based on the comparison between a company’s annual stock value and its market return. Twenty-six ratios are considered as training set to develop the model. The next topic discussed is multicollinearity, where the predictive features are correlated. This tends to cause instability in the data which is fixed mathematically using variance inflation factor (VIF). The square root of VIF measures the approximate error rate. Logistic regression is implemented on the data. Further, the model equation is created using backward elimination. The cutoff value for classification is selected through trial and error method, choosing the one giving the best predictive

fit. Anantakumar and Sarkar [9] deal with binary logistic regression. As ratios can be interdependent, the developed model deals with multicollinearity. Seven ratios were found to accurately classify companies into two categories, good and bad. On the investor aspect, it is seen that ratios produce results more effectively than most other available techniques. This concludes that the logistic regression used to create ratios is the best suited algorithm.

3 Conclusion

This paper shows how each method has its own advantages and disadvantages. Various methods can be used depending on the requirement and availability of parameters. The most common tool for implementing these models is *R* language. It is extremely efficient and cost effective. There is scope for future improvements on each model, and also, new models can be developed by incorporating features from existing tools. The limitation of this approach is that of collecting continuous data and preprocessing such huge datasets.

References

1. S.S. Maini, K. Govindam, Stock market prediction using data mining techniques, in *ICISS* (2017)
2. A. Gupta, P. Bhatia, K. Dave, P. Jain, Stock market prediction using data mining techniques, in *ICAST* (2019)
3. P. Kumar, A. Bala, Intelligent stock data prediction using data mining techniques, in *ICICT* (2016)
4. M. Iyer, R. Mehra, A survey on stock market prediction, in *PDGC* (2018)
5. A. Sharma, D. Bhuriya, U. Singh, Survey of stock market prediction using machine learning approach, in *ICECA* (2017)
6. T. Mankar, T. Hotchandani, M. Madhvani, A. Chidrawar, Stock market prediction based on social sentiments using machine learning, in *ICSCET* (2018)
7. S. Singh, S. Sharma, Forecasting stock price using partial least squares regression, in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (2018)
8. S. Thakur, S. Sharma, S. Singh, Forecasting stock price using conditional inference tree, in *ICACCN* (2018)
9. U. Anantakumar, R. Sarkar, Application of logistic regression in assessing stock performances, in *2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 15th International Conference on Pervasive Intelligence and Computing, 3rd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress* (2017)

Smart Artificial Intelligent-Based Controller for Hydroponic: New Technique for Soilless Plantation



Anurag S. D. Rai, Reeta Pawar, Alpana Pandey, C. S. Rajeshwari, and Ashok Kumar Gwal

Abstract This paper contains Smart Hydroponic Control Scheme (HCS), which is used to monitor, control, and analyze the fault in hydroponics culture. The first part of the paper contains a brief literature about hydroponic systems and their utility. Then in next part, hydroponics systems and greenhouse development along with need of artificial intelligence (AI) and Internet of things (IoT)-based sensor technology for greenhouse culture of hydroponic are being highlighted. The last part of the paper comprises of controller block diagram and controlling approach. Hydroponic internal biological parameters are not directly measured in general. As in plants their conditions would be better predicted and monitored by their root-zone-micro-environment conditions. Therefore in hydroponics, we can control climate and harvest same plants round the year.

Keywords AI-Artificial intelligent · CC-Carbon credits · IoT-Internet of things · NN-Neural network · HCS-Hydroponic Control Scheme

A. S. D. Rai (✉) · R. Pawar

Department of Electrical and Electronics Engineering, Rabindranath Tagore University, Bhopal- Chiklod Road, Raisen 464993, MP, India

e-mail: rai.anurag71@gmail.com

R. Pawar

e-mail: reetapawar2010@gmail.com

A. Pandey

Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, MP, India

e-mail: alpanasubodh@gmail.com

C. S. Rajeshwari

Department of Electrical and Electronics Engineering, National Institute of Technical Teachers Training and Research, Shyamla Hills, Bhopal, MP, India

e-mail: csrajeshwari@nittrbpl.ac.in

A. K. Gwal

Rabindranath Tagore University, Bhopal- Chiklod Road, Raisen 464993, MP, India
e-mail: ashok.gwal@gmail.com

1 Introduction

Hydroponics is technique to cultivate the plants in nutrients feed culture which is one of the techniques to grow plants in soil less medium. Hydroponics in past is gray shaded with facts as this process is utilized for marijuana. But later it was realized that this technique is having more scope in this growing world of 8.1 billion expected in 2025 whereas 9.6 billion at 2050 report of new United Nations Survey. Most of the part of world is not having proper water for utility, and hence it is very essential to re-explore techniques with less polluting medium and environmental utility. One of the method is hydroponics, and it is having following sub-classifications' :

1. Aeroponic systems
2. Nutrient film technique (NFT)
3. Drip system (recovery or non-recovery)
4. Ebb and flow systems (flood and drain)
5. Water culture
6. Wicks system.

These are the few techniques utilized for the growing of plants in soil less hydroponic culture.

This technique has nowadays become popular as it was coined by new tech-startups in India, with the market potential of 12,106.5 million US dollar globally. In this paper, AI intact IoT-based greenhouse-based Smart Hydroponic Control Scheme is discussed (Fig. 1).

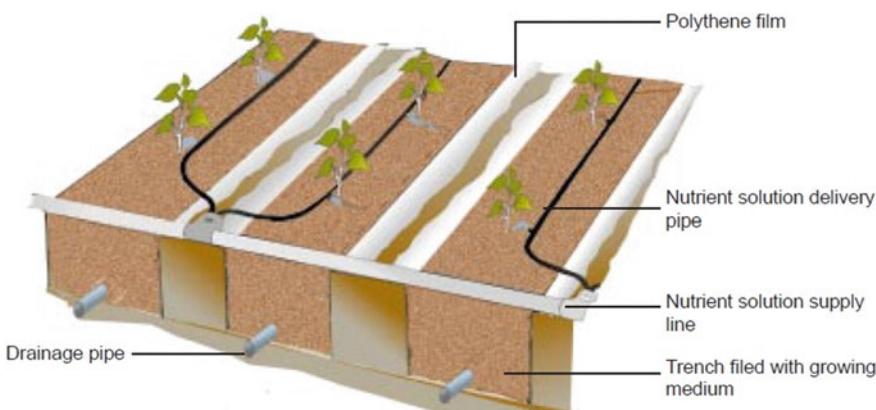


Fig. 1 Hydroponic culture representation

2 Need of Smart Greenhouse

Greenhouse effect is known for the retention of the heat inside the planet by different natural layers by which reflection and refraction of layers have made it possible to achieve conditions essential for life. Similar to this effect, the recreated artificial environment by use of equipment is known as greenhouses. As hydroponic is more intended to use controlled environment which is achieved by the up-gradation of conventional greenhouse, use of automation for controlled environment is needed for the growth of plants in optimum nutrition's and supported conditional parameters. As the hydroponics is having different classifications, and by this difference control scheme, monitoring parameters were changed. The reference controlling parameters were changed from method of utility of hydroponic technique adopted for growth of plant, and this lead to a problem in selection and adoption of need oriented controller. But by the study of different classification-based hydroponic greenhouses and study of their parameters will help in creation of Smart Hydroponic Control Scheme (HCS) to be utilized for general purpose use mostly. The IoT-based scheme with adopted AI helped to make an effective controller for hydroponic greenhouse.

The floating gardens at Kashmir where plants are grown in shallow lakes using raft are an example of hydroponic culture presence in our nations. In controlled condition, the growth rate of the plant is increased as it do not have to compete with adverse climatic conditions and weeds too. Hydroponic provides users a space to utilize this scheme in greenhouse, balcony, or in kitchen garden with small space utilization to large-scale commercial production. As done by Fujitsus, a semiconductor transistor manufacturing company of Japan now transformed to biggest greenhouse and fresh plants and vegetable producer an example of hydroponic technique utilizers. As the system utilizes automation and controlled monitoring of parameters:

- (a) Light
- (b) pH
- (c) Electrical conductivity (EC)
- (d) Flow of water
- (e) Pressure
- (f) Humidity
- (g) Aeration
- (h) Nutrition
- (i) Airflow
- (j) Vibration in hydroponic setup (artificial pollination without natural media).

3 AI-IoT Algorithm for Hydroponic Control Scheme

As every sensors we are having provide electrical outputs which leads to operation matrix and balance of system design. In HCS too we have instantaneous AI-IoT algorithm which is as shown in Fig. 2 [16], which describes the instantaneous interaction

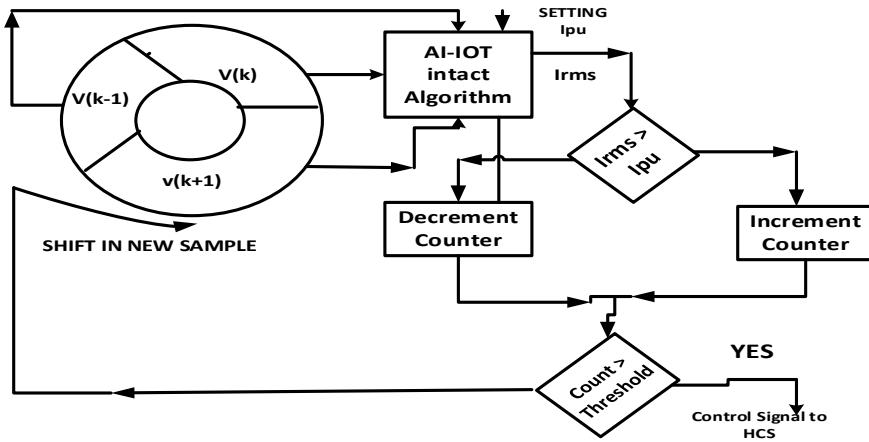


Fig. 2 Instantaneous AI-IoT algorithm

of multiple signal and their actuation on the basis of priority sets. Further relay sets and their actuation pattern were described below as nowadays analog, static, and microprocessor-based both types of relays are used whose differential pattern was shown in Fig. 3.

At relay voltage and current is as:

$$v = V_m \sin(\omega t + \theta_v) \quad (1)$$

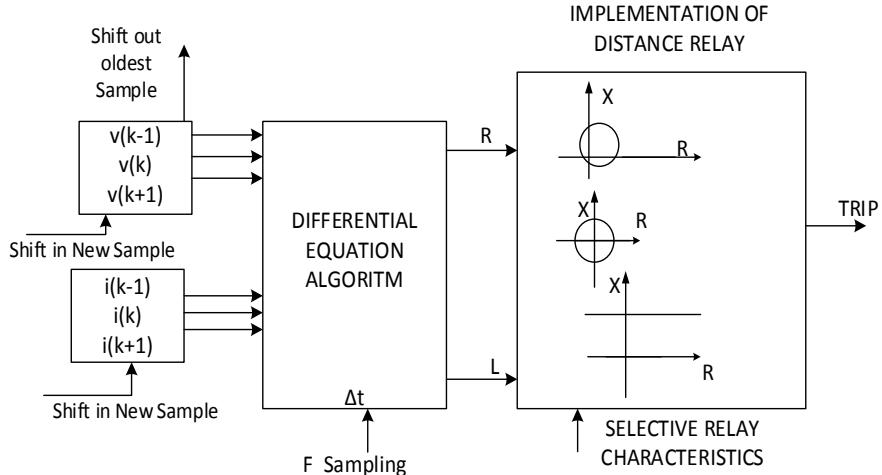


Fig. 3 Block diagram showing inputs to differential equation algorithm and block for implementation of specific relay

$$i = I_m \sin(\omega t + \theta_i) \quad (2)$$

The phasor relation between volte and current is as:

$$\theta = \theta_v - \theta_i \quad (3)$$

We keep on varying our frame containing RMS values of voltage and current, thus variation of phase angle between voltage and current led to computation of the apparent impedance taken from the relay location. The algorithm represents step-by-step procedure and provide mathematical and computational solution with simplicity. However, on examining AI-IoT algorithm technique, it was investigated that it had various input from different sensors and moving toward the HCS which are interacting variable controlling instantaneously with comparative logics. AI is now developed to such degree of intelligent control capable of self-decision.

4 Differential Equation Algorithm for HCS Relaying in Greenhouse

The sensors used have electrical variation in output which is logged and then these differential output signals of different sensors were channelized to achieve the controlling signals of the greenhouse of HCS. Thus, with the help of differential equation, the impedance of fault location can be feed to relay by voltage and current variation [16].

$$v = i_x R + L \frac{di_y}{dt} \quad (4)$$

By this differential algorithm for relay actuators in HCS, faulty operation of relay and sensors was also detected, as the algorithm helps in cross-verification of real-time signal with the reference signal which describes the operational conditions of smart greenhouse. Operation of greenhouse means the internal and external condition of hydroponic system and their variable monitoring parameters along with biological faults which are measured. As root-zone-micro-environment of plants is very critical issues, transpiration rate, proper intake of nutrition, and electrical conductivity (Ec) will effect root-zone. Growth of plant is directly affected by the root-zone condition and their grooming which is monitored by the flow and pressure of fluid inside the system. As an AI-IoT based Hydroponic Control Scheme (HCS) let efficient by imparting it with reference data of different conditions and sub-classification working parameters to take self-logical decision with change in decision.

5 Hydroponic Control Scheme

As in the proposed HCS, sensing of internal and external environment of greenhouse is one of the important tasks along with variation of parameters in root-zone-micro-environment. The above block diagram of control scheme is presenting an architecture which enables artificial intelligent approach intact with Internet of things in hydroponic greenhouse. Feedback loop helps in enabling self-decision features by comparison actuation of reference and present signal (Fig. 4).

1. As HCS is developing and under continuous improvement of algorithms and architecture, technical specifications which we are utilizing are as follows.
- A. Sensor utilized for temperature measurement is LM35DZ with accuracy of $0.4\text{ }^{\circ}\text{C}$. Output voltage variation is proportional to temperature measured in $^{\circ}\text{C}$.
- B. Sensor utilized for humidity is of Honeywell HIH 4000–001.
- C. Node MCU is utilized to monitor modules.
- D. DHT11 is utilized for monitoring of temperature and humidity.
- E. KG003 is used for the soil moisture, and here it is utilized to know the dryness in the system.
- F. Four channel relay boards are utilized by different combination for switching between various sensors-driven logical implementation.
- G. HC-SR04 an ultrasonic sensor module (Fig. 5).

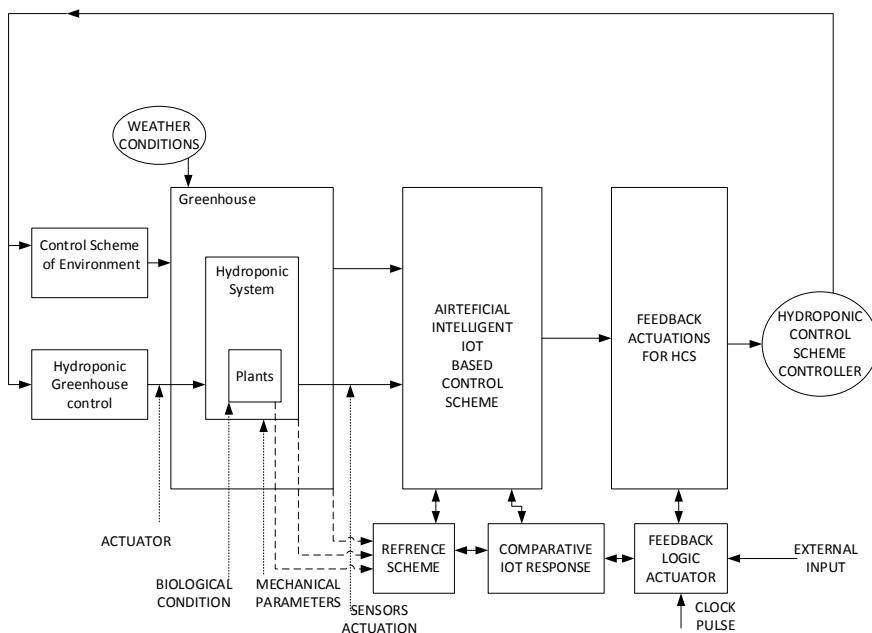


Fig. 4 Block diagram showing Hydroponic Control Scheme (HCS)

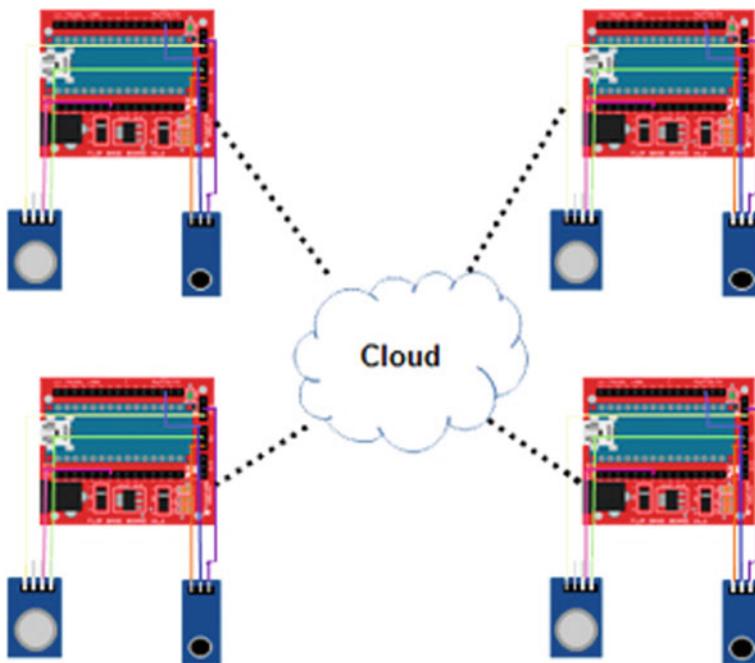


Fig. 5 IoT with hardware and cloud interaction

6 Methodology of the System

Step 1—Sensor devices continuously sense the environment of greenhouse and root-zone-micro-environment along with other variable parameters.

Step 2—Data collected by sensors at predefined time.

Step 3—Data is uploaded to Internet through Wi-Fi network and stored in Microsoft Azure Cloud.

Step 4—In case of deviation detected from standard operation, AI-based controller compared the deviation and perform the required task.

Step 5—Stored data can be utilized for further research, and inculcating the intelligent operation required during particular operation.

7 System Design and Realization

A. Future hydroponic HCS-based greenhouse.

An ideal hydroponic, HCS-based greenhouse system which is utilizable in common space nearby is proposed in this research paper whose 3-D model is shown in Fig. 6a, where bulb-type plant erector utilizing NFT model is also shown in Fig. 6b. As the IoT-based systems are improving day by day with implementation and availability of new more efficient sensors and controllers with which this scheme, its reach will become more economical.

B. System Implemented

These are the sensors and module present in center of innovation in IoT which are utilized for preliminary test system designing utilizing Arduino. As the scheme is using AI architecture in Figs. 2 and 3, algorithm for the control system and relaying is explained which forms the base of the HCS.

As the system is under continuous improvement and testing for setting our own practical data in Fig. 7, different sensors are shown. Arduino is utilized for modules control by implementation of Fig. 2—AI-IoT architecture. The system is integrated using Frugal Labs Bangalore, which are providing cloud server and IoT needs to the innovation

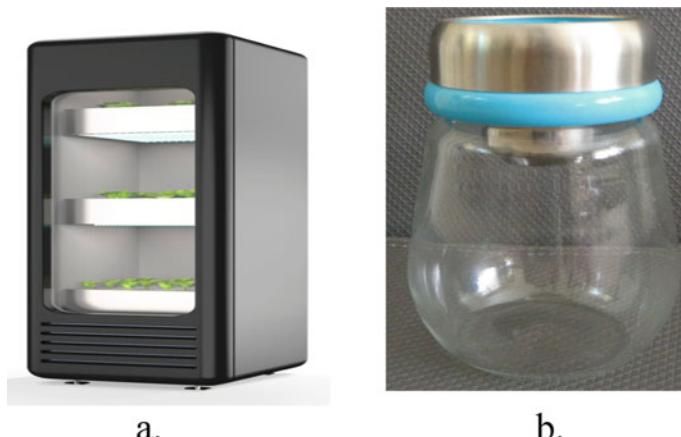


Fig. 6 **a** Prototyped 3-D modeled HCS-based greenhouse system. **b** Plant erector using NFT, 3-D modeled figures

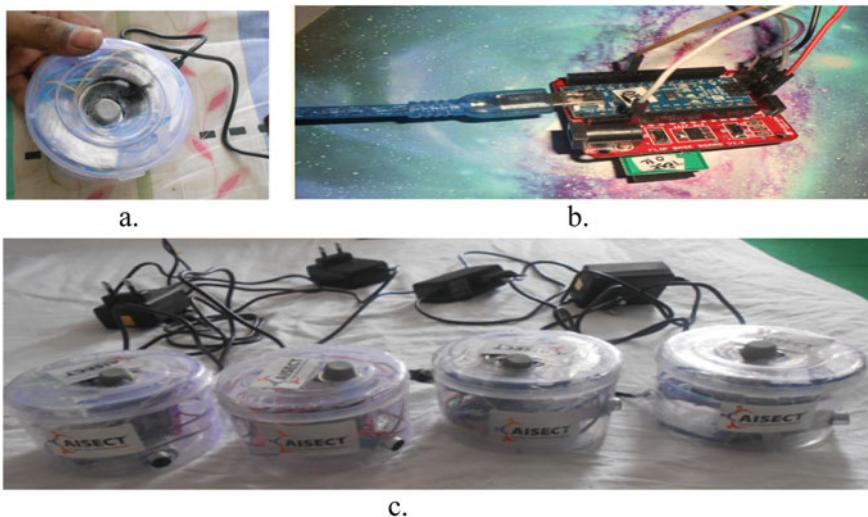


Fig. 7 **a** Temperature sensor **b** arduino-based test controller **c** different sensors utilized in hydroponic control scheme

8 Results

The control scheme is dependent on variation of sensors output, providing intelligence in any change using logic need, reference parameters, and decision to be done in the deviation of any. Artificial intelligence meant taking self-decision without human intervention in change in situation effectively. Here also by continuous variation in parameter testing and impact on system lead to impart that variation effect and decision making related to it. As the child in childhood is taught many basic lessons by doing itself, which develop decision making, here in AI-based system, we had to inculcate those decision making their Dos and Don'ts which effectively provide algorithm to react quickly and correctly on IoT output variation.

Temperature, Humidity, and Air-quality variations are recorded in hydroponic greenhouse system. The variation of the output is shown in Fig. 8. As the data is stored in cloud server, variation of parameter is also recalled at any instant of time and its impact on physical system is analyzed. Other sensors and their parameters are also to be tested, as elementary algorithm for testing the system to design HCS basics is needed.

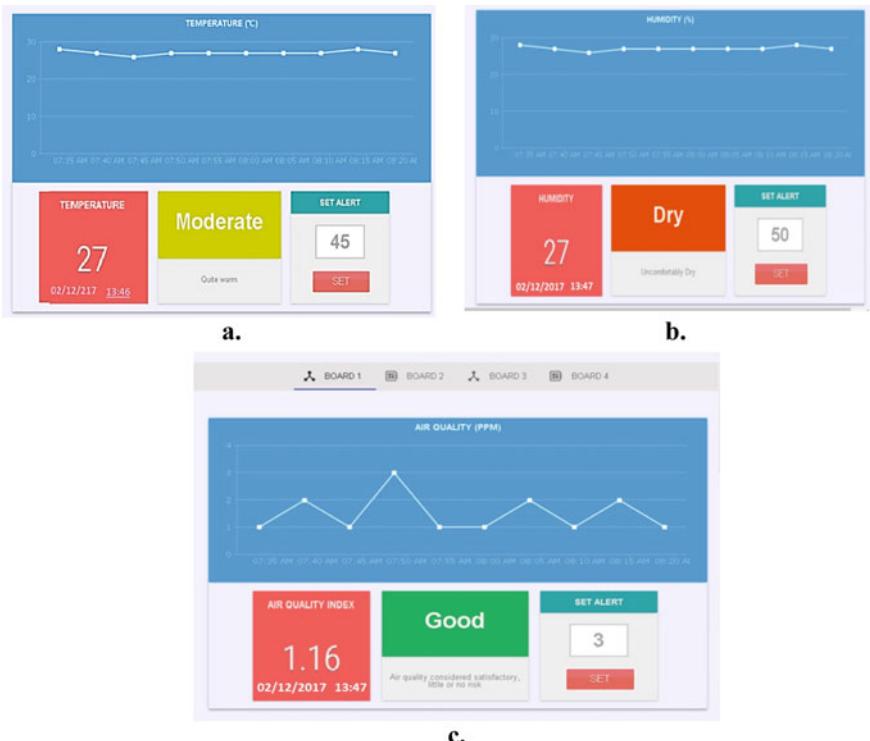


Fig. 8 **a** Temperature variation monitoring **b** humidity variation monitoring **c** air-quality (ppm) variation display, graphs are displayed in OS using cloud data

9 Conclusion

As the Hydroponic Control Scheme is under development, it is very early to predict its impact in the growth and nutrition control of plants. By the controlling of plants in controlled environment and with needed monitoring reduce, the grown-up period is required by plants with controlled nutritions. Artificial intelligent enables the HCS controller for comparative action and implement when required this reduce human intervention in the system. Architecture of the proposed HCS is explained along with sensors and module utilized for developing it. This new approach toward development of hydroponic growing make it easy accessible to farmers and kitchen gardeners in near future.

References

1. Y. Hashimoto, Recent strategies of optimal growth regulation by the speaking plant concept. *Acta Hort.* **260**, 115–121 (1989)
2. J.C. Hoskins, K.M. Kalayur, D.M. Himmelblau, Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE J.* **37**(1), 137–141 (1991)
3. E. Filho, A. de Carvalho, Evolutionary design of MLP neural network architectures, in *Proceedings of the Fourth Brazilian Symposium on Neural Networks*, pp. 58–65, Goiania, GO, Brazil, Dec 3–5 (1997)
4. K.P. Ferentinos, Artificial neural network modeling of pH and electrical conductivity of hydroponic systems, MS Thesis, Cornell University Libraries, Ithaca, NY (1999)
5. K.P. Ferentinos, L.D. Albright, B. Selman, Neural network based detection of mechanical, sensor and biological faults in deep-trough hydroponics. *Comput. Electron. Agric. Spec. Issue Artif. Intell. Agric.* (2002)
6. K.P. Ferentinos, Neural network fault detection and diagnosis in deep-trough hydroponic systems. Ph.D. Dissertation, Cornell University Libraries, Ithaca, NY (2002)
7. A.W. Al-Kayssi, Spatial variability of soil temperature under Greenhouse conditions. *Renew. Energy* **27**, 453–462 (2002)
8. H. Sundmaeker, P. Guillemin, P. Friess, S. Woelflé (eds.), Publications Office of the European Union, Luxembourg (2010)
9. G.L. Atzori, A. Iera, G. Morabito, The Internet of things: a survey computer network. *Comput. Netw.* **54**, 2787–2805 (2010)
10. H.S. Grewala, B. Maheshwaria, S.E. Parks, Water and nutrient use efficiency of a low-cost hydroponic greenhouse for acucumber crop: An Australian case study. *Agric. Water Manag.* **98**, 841–846 (2011)
11. I. Mohanraj, Field, “monitoring and automation using IoT in agriculture domain”. *Procedia Comput. Sci.* **93**, 931–939 (2016)
12. M. Azaza, C. Tanougast, E. Fabrizio, A. Mami, Smart greenhouse fuzzy logic based control system enhanced with wireless data monitoring, vol. 61, pp. 297–307 (2016)
13. J. delSagrado, J.A. Sánchez, F. Rodríguez, M. Berenguel, Networks for greenhouse temperature control. *J. Appl. Logic* **17**, (25–35) 2016
14. O. Dlugosz-Grochowska, A. Kolton, R. Wojciechowska, Modifying folate and polyphenol concentrations in Lamb's lettuce by the use of LED supplemental lighting during cultivation in greenhouses. *J. Funct. Foods* **26**, 228–237 (2016)
15. Libelium, 50 Sensor applications for a smarter world, http://www.libelium.com/top_50iot_sensor_applications_ranking
16. A.S.D. Rai, R. Pawar, D. Sharma, S. Sen, S.K. Gupta, Algorithms for synchrophasor enabled digital relay in differential protection scheme, in *Proceedings of International Conference on Recent Advancement on Computer and Communication, Lecture Notes in Networks and Systems* vol. 34. Springer Nature Singapore Pte Ltd., https://doi.Org/10.1007/978-981-10-8198-9_4. Book Id: 448040_1_EN, Book ISBN: 978-981-10-8197-22018

Human Action Detection Using Deep Learning



S. Gowri, Syed Aarif Suhaib Qadri, Suvam Bhowal, and J. Jabez

Abstract This Method centers around one as of late bringing task up in vision and media look at: seeing human exercises from still pictures. Its key troubles belong in the colossal assortments of human positions and appearance, similarly as the non-appearance of common developments. Watching out for these issues, we propose to develop a significant model to typically organize human configuration and also enveloping settings for progressively raised level movement understanding from stiff pictures. Specifically, a deep belief net is set up to entwine the information from different noises, for instance, body point ID and article disclosure. To interface the semiotics gap, we have a physically named data to immensely improve the suitability and profitability of the pre-getting ready and steady tuning periods of the DBN planning. The consequent structure is exhibited to a great extent unstable information sources (e.g., free recognizable pieces of proof of human parts and questions) and beats the forefront moves close.

Keywords Profound learning · Convolutional neural systems · 3D convolution · Model blend · Activity acknowledgment

S. Gowri (✉) · S. A. S. Qadri · S. Bhowal · J. Jabez
Sathyabama Institute of Science and Technology, Chennai, India
e-mail: gowri.it@sathyabama.ac.in

S. A. S. Qadri
e-mail: syedaarif1999@gmail.com

S. Bhowal
e-mail: suvambhawali80@gmail.com

J. Jabez
e-mail: jabezme@gmail.com

1 Introduction

Perceiving human activities introduced applications in miscellaneous of spaces including smart video reconnaissance, properties of clients, and shopping conduct investigation. Nonetheless, accurate concession of activities is an exceptionally testing errand because of mix foundations, obstacle, slant varieties, and so on. A large portion of new methods make some suppositions (e.g., small quantity and specific changes) of the conditions under which the video was made. Be that as it may, such suspicions only here and there hold in reality. Also, the strategies follow a methodology which is a two advanced wherein the first step registers highlights from some video outlines and the subsequent advance recognizes classifiers dependent on the highlights. In true places, it is the once in a while understands what highlights are significant for the job need to be done since the selection of highlights is random. Specifically for human action recognition, many diverse activity classes may show up unique as far as their point of presentation and movement print. The convolutional neural networks are a type of models which are complex and in which trained filters and local neighborhood pooled operations are applied alternatively on the untouched input, which are resulting in increasingly complex features. It has been proved already that, when trained with all the appropriate regularization convolutional neural networks can achieve great performance on object visual recognition tasks or identification. Also, the convolutional neural networks have shown to be invariable to many variations such as pose, lighting, and surrounding clutter.

2 Related Works

We assessed the created 3D convolutional neural networks model on the TREC Video Retrieval Evaluation (TRECVID) information, that comprises of reconnaissance video information recorded in London Gatwick Airport. We had built a multi-modal occasion location framework, which incorporates the 3D convolutional neural networks as a significant module, and partook in the three assignments of the TRECVID 2009 Evaluation for the surveillance event detection. The framework of ours is accomplished the best execution on each of the three taking an interest activity classes (i.e., CellToEar, Object Put, and Pointing). To give an autonomous assessment of the 3D convolutional neural networks model, we need to report its presentation on the TRECVID 2008 improvement that has been set in this paper work. We additionally make present outcomes on the KTH information as distributed execution for this information that are accessible. Our examinations show that the created 3D convolutional neural networks model beats other gauge techniques on the TRECVID information, and it accomplishes aggressive execution on the KTH information, exhibiting that the 3D convolutional neural networks model is progressively

powerful for genuine conditions, for example, those caught in the TRECVID information. The trials additionally approve that the 3D convolutional neural networks model altogether beats the casing-based 2D convolutional neural networks for the most errands.

3 Literature Survey

Activity acknowledgment has pulled in expanding consideration from RGB contribution to PC vision incompletely because of potential applications on substantial reproduction and measurements of game, for example, virtual tennis match-up and tennis methods and strategies investigation by video. As of late, profound learning-based strategies have accomplished promising execution for activity acknowledgment. In this paper, we propose weighted long short-term memory embraced with convolutional neural system portrayals for three-dimensional tennis shots acknowledgment. To start with, the nearby two-dimensional convolutional neural systems spatial portrayals are separated from every video outline independently utilizing a pre-prepared inception organize. At that point, a weighted long short-term memory decoder is acquainted with take the yield state at time t and the verifiable installing highlight at time $t-1$ to create include vector utilizing a score weighting plan. At long last, we utilize the received convolutional neural networks and weighted LSTM to delineate unique visual highlights into a vector space to create the spatial-transient semantically depiction of visual successions and group the activity video content. Examinations on the benchmark show that our technique utilizing just straightforward crude RGB video can accomplish preferred execution over the cutting edge baselines for tennis shot acknowledgment. This was proposed by Zhaoqiang Chen; Qun Chen; Zhanhuai Li in the year 2017 [1].

Action parsing-driven video summarization based on reinforcement learning by Ihab F. Ilyas; Xu Chu in the year 2015 [2]. This framework proposes an activity parsing-driven video synopsis model dependent on fortification learning. The model is for the most part separated into two sections, video cut by activity parsing and video rundown dependent on fortification learning. In the initial segment, a successive numerous case learning model is prepared with feebly clarified information to take care of the issue of full explanation is tedious and frail comments vagueness. In the subsequent part, we structure a profound intermittent neural system-based video synopsis model which chooses the most discernible casings contrasting and different activities. In the meantime, the nature of the separated key edges could be assessed by the arrangement precision.

Scene categorization using deeply learned gaze shifting Kernel by Xu Chu; Ihab F. Ilyas; Paolo Papotti is in the year 2013 [3]. They proposed a profound look moving portion to recognize views from various classes. In particular, first task districts from every landscape into the purported perceptual space, which is set up by consolidating shading, surface, and semantic highlights. At that point, a novel non-negative network factorization calculation is created which disintegrates the regions including

framework into the result of the premise lattice and the meager codes. The scanty codes demonstrate the saliency level of various areas. Along these lines, the look moving away from every view is determined and a total-based convolutional neural system is structured in like manner to get familiar with its profound portrayal. At long last, the profound portrayals of look moving ways from all the scene pictures are joined into a picture piece, which is additionally sustained into a part SVM for scene arrangement and proposed a novel profound engineering called profound look moving part for scene classification. DGSK centers around precisely distinguishes human look moving ways by successively associating striking portions with subjective shapes, wherein numerous highlights at both low level and significant level are built.

High-dimensional sparsifying transform learning for online video denoising by Joeri Rammelaere; Floris Geerts; Bart Goethals in the year 2017 [4]. This framework proposes techniques for Web-based taking in of sparsifying changes from spilling signals, which appreciate great intermingling ensures and include lower computational expenses than online union lexicon learning. Proposed work applies online change figuring out how to video denoising. This methodology shows a novel system for online video denoising dependent on high-dimensional sparsifying change learning for spatiotemporal patches. The patches are developed either from relating 2D fixes in progressive edges or utilizing an online square coordinating procedure. The proposed online video denoising requires little memory and offers proficient handling. Numerical tests assess the presentation of the proposed video denoising calculations on different video informational collections. The proposed strategies beat a few related and ongoing systems incorporating denoising with 3D DCT, earlier plans dependent on word reference learning, non-neighborhood implies, foundation detachment, and profound learning, just as the famous VBM3D and VBM4D.

Deep learning-based real-time fine-grained pedestrian recognition using stream processing by Philip Bohannon; Wenfei Fan; Floris Geerts; Xibei Jia; Anastasios Kementsietsidis in the year 2007. This framework proposes an answer for fine-grained person on foot acknowledgment in checking situations utilizing profound learning and stream handling distributed computing, which is called DRPRS (profound learning-based constant fine-grained walker acknowledgment utilizing stream preparing). This framework structures an improved convolutional neural system (convolutional neural networks) arrange called fine-convolutional neural network, which is a nine-layer neural system for point-by-point person on foot acknowledgment. In DRPRS, a person on foot in an observation video is sectioned and fine-grained recognized utilizing improved single-shot indicator and a few fine-convolutional neural networks. DRPRS is upheld by equal components given by Apache Storm stream handling structure. Also, so as to additionally improve the acknowledgment execution, a GPU-based booking calculation is proposed to utilize GPU assets in a bunch. The entire acknowledgment process is sent on a major video information handling stage to meet constant prerequisites. DRPRS is broadly assessed regarding exactness, adaptation to internal failure, and execution, which show that the proposed approach is effective.

4 Existing System

The activity parsing in the recordings with the complex scenes is an intriguing however testing assignment in PC vision. Here in the paper, we introduce a non-exclusive 3D convolutional neural system to perform various tasks learning way for compelling deep action parsing (DAP3D-Net) in recordings. Especially, in the preparation stage, activity confinement, grouping, and properties, learning can be mutually improved on our appearance-movement information by means of DAP3D-Net. For up and coming test video [5, 6], we can depict every single activity in the video [7, 8] at the same time as: where the activity happening, what are the activity that are happening and how the activities are performed. To well show the viability of the introduced DAP3D-Net, we additionally contribute another numerous-classification aligned synthetic action dataset, that is, NASA, which comprises of 200,000 activity clasps of in excess of three-hundred classifications and with thirty-three pre-characterized activity traits in two progressive levels (i.e., low-level qualities of fundamental parts of the body's developments and elevated level credits identified with activity motion). Approaches centers around 2D pictures.

5 Proposed System

The proposed system describes each individual action as where the action is occurring, which action it is, how the action are being performed. In other words, it shows accurately localization, categorization and also describes multiple actions in the realistic videos. It also automatically parses the action in videos.

The proposed work contains four different modules. In the first module, from the client, we upload the video file to the server. The video file contains the information about the action to be detected. Once uploaded the server stores the file the server hard disk. In the second module, we retrieve frames one by one from the uploaded videos. Each retrieved frame will be stored in the server for further analysis. In the third module, we retrieve frames one by one from the uploaded videos. Any noise (not sound) (any unwanted dot or line) will be removed from the frame images. In the fourth and the last module using deep learning algorithm, detect the action which is available in the uploaded videos.

For example, when we parse a video, it will be extracting its frames and then through those it will be recognizing or detecting what action it is, for instance, if the person is waving the hands that means the person is showing the action bye. The sequential steps of the proposed system is shown in the below Fig. 1.

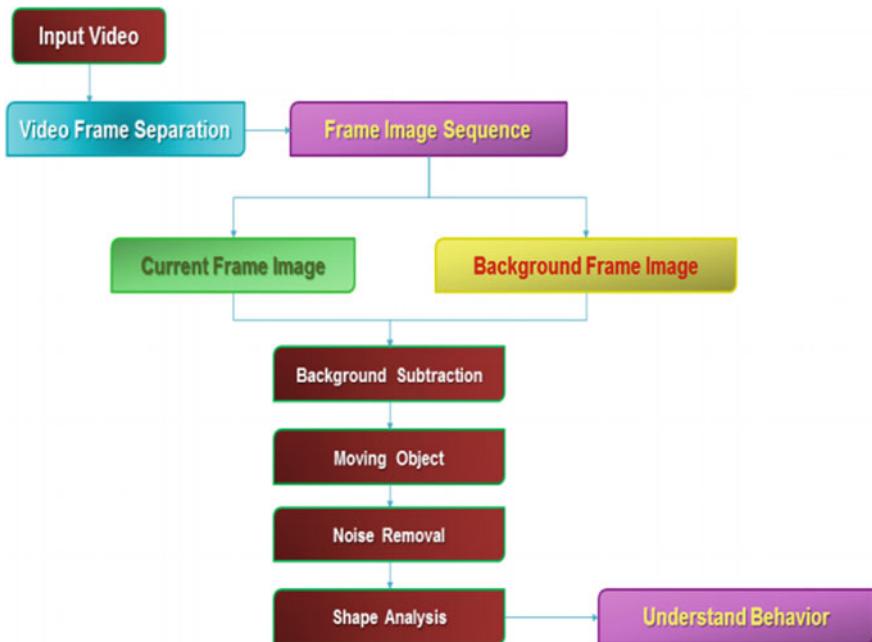


Fig. 1 Proposed system architecture

6 Future Enhancement

We will investigate the solo preparing of 3D convolutional neural networks models later on. Earlier investigations have shown that the quantity of named tests can be fundamentally diminished when such a model of method is pre-trained utilizing unaided calculations.

7 Conclusion

We created 3D convolutional neural networks models for the activity acknowledgement in the paper. These are the models that build highlights from the both spatial as well as transient measurements by performing 3D convolutions. They also create various channels of data from nearby info outlines and perform convolution and also the sub-sampling independently for in each channel. The last component portrayal is acquired by consolidating data from all channels. We created a regularization of model and blend plans to support regularization of the model execution. We assessed the 3D convolutional neural networks models on the TRECVID and the KTH informational indexes. Results show that the 3D convolutional neural networks

model beats analyzed techniques on the TRECVID information, while also it accomplishes aggressive execution on the KTH information, exhibiting the predominant presentation in certifiable conditions.

References

1. Z. Chen, Q. Chen, Z. Li, A human-and-machine cooperative framework for entity resolution with quality guarantees, in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, (2017), pp. 1405–1406. IEEE
2. J. Lei, Q. Luan, X. Song, X. Liu, D. Tao, M. Song, Action parsing-driven video summarization based on reinforcement learning. *IEEE Trans Circ Syst Video Tech* **29**(7), 2126–2137 (2018)
3. W. Zhang, Z. Wang, X. Liu, H. Sun, J. Zhou, Y. Liu, W. Gong, Deep learning-based real-time fine-grained pedestrian recognition using stream processing. *IET Intell Trans Syst* **12**(7), 602–609 (2018)
4. B. Wen, S. Ravishankar, Y. Bresler, VIDOSAT: High-dimensional sparsifying transform learning for online video denoising. *IEEE Trans Image Process* **28**(4), 1691–1704 (2018)
5. B. Antic, B. Ommer, Video parsing for abnormality detection, in *ICCV*, 2011
6. T. Ko, A survey on behavior analysis in video surveillance for homeland security applications, in *Applied Imagery Pattern Recognition Workshop*, 2008
7. J. Qin, L. Liu, M. Yu, Y. Wang, L. Shao, Fast action retrieval from videos via feature disaggregation, in *BMVC*, 2015
8. J. Yuan, Z. Liu, Y. Wu, Discriminative video pattern search for efficient action detection. *T-PAMI* **33**(9), 1728–1743 (2011)

Automatic Detection of Leaf Disease Using CNN Algorithm



S. Nandhini, R. Suganya, K. Nandhana, S. Varsha, S. Deivalakshmi, and Senthil Kumar Thangavel

Abstract In Indian market, the highest commercial staple is tomato crop. The production of apples constituted 2.40% of the total fruits produced in India, and Maize is one of the highest yielding crops in the world, thus known as ‘miracle crop.’ These plants’ health and growth are usually affected by the diseases. There are various types of tomato, maize and apple leaf diseases that affect the crop. This paper uses the convolution neural network to detect and identify the diseases in the leaves by image classification. The main objective of the proposed system is to find a solution for the problem of tomato, corn and apple leaf diseases using the neural network. The proposed convolutional neural network model has eight layers including five convolution and three max pooling layers. The proposed system has achieved accuracy from the range 96–98% for three different types of the leaf images indicating the feasibility of neural network method.

Keywords Leaf diseases · Convolutional neural network · Deep learning · Image classification · Plant village dataset · Data augmentation

S. Nandhini · R. Suganya · K. Nandhana · S. Varsha
Thiagarajar College of Engineering, Madurai 625015, Tamilnadu, India
e-mail: nandhini@student.tce.edu

R. Suganya
e-mail: rsuganya@tce.edu

K. Nandhana
e-mail: nandhana@student.tce.edu

S. Varsha
e-mail: varshas@student.tce.edu

S. Deivalakshmi
National Institute of Technology, Tiruchirappalli, Tiruchirappalli, India
e-mail: deiva@nitt.edu

S. K. Thangavel
Amrita School of Engineering Coimbatore, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: t_senthilkumar@cb.amrita.edu

1 Introduction

Agriculture is the backbone of India, and tomatoes are one of the major horticulture crops in India with Andhra Pradesh being its largest producer. India is one of the leading producers of tomatoes with a production of 163.96 million tons per year. Tomatoes are high in nutrients like Vitamin 'A' and 'C' and are high sources of income as they can be cultivated throughout the year even though they are predominantly summer crops. Likewise, apples are also one of the most cultivated fruits in India constituting 2.40 percent of the total fruits produced. Apples are a high source of Vitamin C and are rich in minerals. Maize is one of the highest produced cereal crops in India that accounts for 42.3 million metric tons per year. It has got a very high yield potential and hence the name 'Queen of Cereals.' Maize is a good source of dietary fiber and protein rich in phosphorus, magnesium, manganese, zinc, copper, iron and selenium. Leaf diseases are very common due to varying climatic conditions and causative organisms like fungi and bacteria. This decreases the crop yield severely and brings a huge loss both in quality and quantity of the crop production. Manual monitoring and detection of diseases are very difficult processes due to the requirement of skilled farmers for accurate prediction of diseases, huge time and cost. The incorrect prediction of diseases may lead to overdosage or underdosage of pesticides leading to total crop damage. Thus, the proposed methodology is introduced for accurate detection and classification of diseases to make farmers part a bit easier. This system consists of common diseases found both in tomato and apple. The leaf is given as an input which is processed and compared with the images in the trained database. As a result, it classifies the given leaf which has any one of the disease class or is healthy. The tomato leaf has been affected by nine different diseases that include bacterial spot, early blight, leaf mold, Septoria leaf spot, mosaic virus and so on. The apple leaf diseases are of three different types that are apple scab, black rot, cedar apple rust. The corn leaf diseases are also of three different types that include common rust, northern leaf blight, Cercospora leaf spot.

The proposed solution in [1] classifies tomato using a simple approach. The dataset consists of 383 images, and Otsu's image segmentation method has been applied for implementation. All the extracted color, shape and texture features are used to form a feature extraction module. For the classification, the supervised learning techniques are used. The accuracy in this proposed model is high. But there are some notable disadvantages in the decision tree like overfitting and reduced manual control on the model.

In [2], the proposed solution is used to automatically identify the NLB lesions with high reliability. In the experimental settings, several CNN models were trained to check the NLB lesions, and finally, their results were combined to evaluate whether there is disease or not. The images were analyzed in three different stages. Firstly, several CNN models were trained to identify the lesion. Secondly, the trained models were used to produce heat maps, and finally, the output was used to classify the image dataset. The system has achieved an accuracy of 96.7% on the test set of images.

In [3], the authors have proposed a solution for identifying the maize leaf disease using improved deep convolutional neural networks. Their methodology includes GoogleNet and Cifar10 models for the leaf disease recognition. In experimental settings, GoogleNet model has used 22 layers, and when there is increase in depth and training data, the model has more features comparatively, whereas the Cifar 10 model has used six layers and there is a pooling layer and ReLU operation after each convolutional layer. The system has achieved an average accuracy of 98.8% using Cifar10 model and 98.9% using GoogleNet model. From the above literature survey, it denotes that deep convolution networks provide better accuracy.

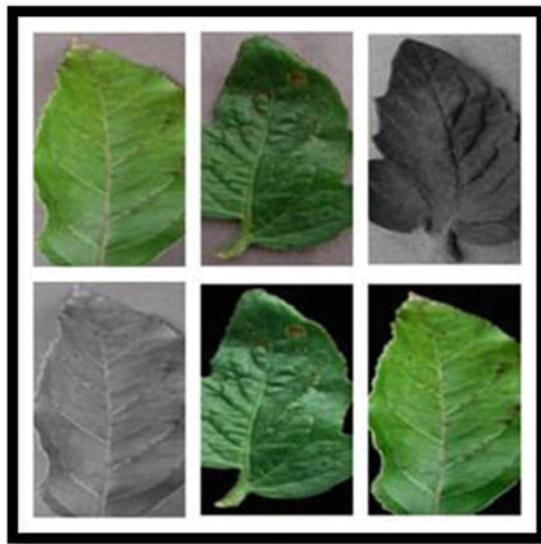
In [4], the proposed methodology is used to detect apple leaf disease using DenseNet-121 as their backbone network. They have proposed three methods of regression, multi-label classification and focus loss function for their identification. They have achieved a maximum accuracy of 93.71% on their test dataset.

Here, the proposed solution uses a convolutional neural network model to classify the leaf diseases. This model provides better accuracy using convolutional neural network than the deep convolutional neural network.

2 Database Collection

The dataset is taken from Github-Plant village. It contains 54,480 images for tomato leaf diseases and 9513 images for apple leaf and 11,556 images for maize leaf diseases with color, gray and segmented images (Fig. 1).

Fig. 1 Leaf images in color, gray and segmented



3 Experimental Settings

The implementation is done using 54,480 of tomato leaf images and 9513 of apple leaf images and 11,556 of corn leaf images from plant village dataset. It consists of 10 different classes of tomato leaf diseases and four different classes of apple and corn leaf diseases, each of which was categorized into color, grayscale and segmented. The dataset is further divided into train and test data set with the ratio of 8:2, respectively. In tomato leaf dataset, there were totally 14,532 images in each of the three folders belonging to nine different disease classes and one healthy leaf class.

Each class belongs to tomato leaf diseases are bacterial spot, early blight, late blight, leaf mold, Septoria leaf spot, spider mites, target spot, mosaic virus and yellow leaf curl virus. In apple leaf dataset, there were totally 3171 images in each of the three folders belonging to three different disease classes and one healthy leaf class. Each class belongs to apple leaf disease are apple scab, black rot and cedar apple rust. The corn leaf dataset consists of 3852 images in each of the three folders belonging to three different disease classes and one healthy leaf class. Each class belongs to corn leaf disease are Cercospora leaf spot, common rust and northern leaf blight.

Data augmentation was carried out to increase the dataset by setting the rotation range as 25 and other augments like horizontal flip, width and height shift range, zoom range and fill mode. Keras is used in this proposed methodology for model implementation and evaluation. The initial learning rate was set as 0.001. Since the computational cost is lesser than tanh and sigmoid function, sequential model is used for data prediction. This model is trained for 50 epochs with a batch size of 32 with 1600 steps per epochs. The optimization was done using Adam optimizer with the loss function, binary cross-entropy. The experiments were performed on Intel Core i5 processor.

4 Proposed Methodology

The proposed methodology consists of three processes that are data acquisition, data preprocessing and classification. The tomato, corn and apple leaf disease datasets were collected from the plant village dataset available in Github repository. The default image size of each image in the dataset is 256×256 . In tomato leaf dataset, there were totally 14,532 images in each of the three folders belonging to nine different disease classes and one healthy leaf class. Each class belongs to tomato leaf diseases that are bacterial spot, early blight, late blight, leaf mold, Septoria leaf spot, spider mites, target spot, mosaic virus and yellow leaf curl virus. In apple leaf dataset, there were totally 3171 images in each of the three folders belonging to three different disease classes and one healthy leaf class. Each class belongs to apple leaf diseases that are apple scab, black rot and cedar apple rust. The corn leaf dataset consists of 3852 images in each of the three folders belonging to three different

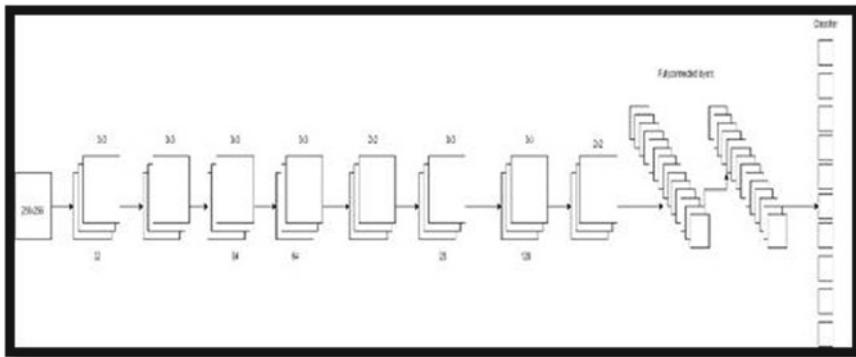


Fig. 2 Model architecture

disease classes and one healthy leaf class. Each class belongs to corn leaf diseases that are Cercospora leaf spot, common rust and northern leaf blight. And all the images were stored in .jpg format by default. These leaf datasets are split into 80 and 20% for training and testing. Further, the 80% of training dataset is split into 60 and 20% for training and validation process. As the result of training the model, the training and validation accuracy is obtained. Further, precision, recall and F-score are also obtained. The remaining 20% of the images were tested, and the belongingness of the particular disease class is obtained (Fig. 2).

A total of five convolution layers are added to extract the features. The size of the filter is increased progressively with an initial 3×3 filter. The number of filters is 32 in the first convolutional block, while it is increased to 64 in the third and fourth layer and 128 in sixth and seventh layer. The max pooling layers shrink the size of feature map to fasten up the training. It is prevented by the progressive increase in the filter numbers. Zero padding applied to prevent the lose of size. A total of three max pooling layers are applied to this proposed method. The convolution filter size for max pooling is 3×3 , 2×2 and 2×2 . ReLU activation layer is applied after each block to increase the nonlinearity of the image. Dropout function has been added to avoid the occurrence of overfitting in the train set. Dropout regularization technique is often applied after each pooling layer and sometimes after the convolution layer. It drops the neurons from the previous activation layer and trains it to 0 to decrease the model variance. Here, in this model, the dropout is set as 0.25, 0.25, 0.25 and 0.5, respectively, after each pooling layer. At last, the model is followed by fully connected layer dense layers and softmax activation function.

5 Results and Discussion

The tomato, apple and corn leaf datasets also contain three folders which are color, segmented and gray image folder. The images in each folder of tomato, apple and corn undergo preprocessing and training separately, producing different results of

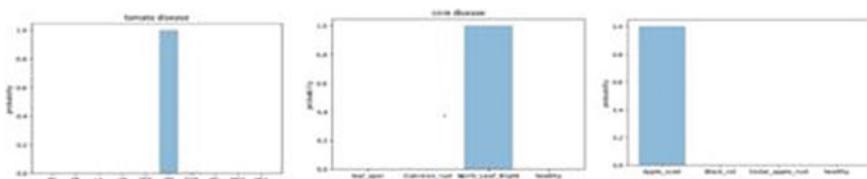


Fig. 3 Resultant bar graph

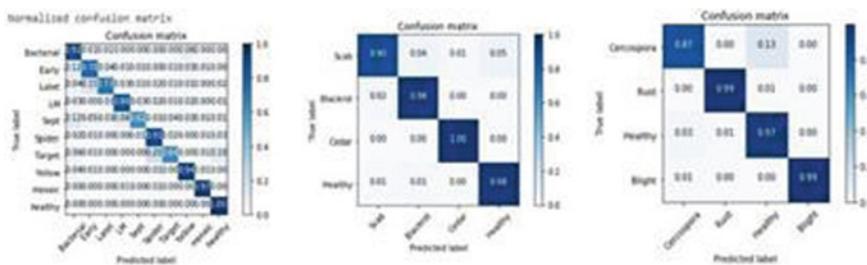


Fig. 4 Confusion matrix

metrics. It has been noted that the accuracy of the model did not change after 20 epochs and remained constant.

Multiclass classifier is used to classify different classes of tomato, apple and corn leaf disease. The resultant array comprises probability of different classes. The index with the highest probability indicates the belongings of the particular image to the respective class. The result of each image is represented in a bar graph (Fig. 3), depicting the class to which the leaf belongs to.

The confusion matrix is used to summarize the test results of the images. The confusion matrix provides better clarity of the results of the classification model. The confusion matrix of all leaf images is provided (Fig. 4).

5.1 Metrics

Precision is also known as positive predictive value which is used as an evaluation metrics to ensure correctness of the prediction. It is the fraction obtained from relevant instances among the retrieved instances.

Precision:

$$\text{TP}/\text{TP} + \text{FP} \quad (1)$$

where

TP = true positive

FP = false positive.

Recall is also known as sensitivity which is defined as the fraction obtained from the total amount of relevant instances that were retrieved. It is based on the understanding of relevance.

Recall:

$$\text{TP}/\text{TP} + \text{FN} \quad (2)$$

where

TP = true positive

FN = false negative.

F1 score metric is a combination of both recall and precision. It is the mean of both metrics. It is used to evaluate the performance of the model without specifying the recall and precision values.

F1 score:

$$\text{TP}/\text{TP} + \text{FP} + \text{FN} \quad (3)$$

where

TP = true positive

FP = false positive

FN = false negative (Tables 1, 2 and 3).

Table 1 Metrics of tomato leaf

| Type | Precision | F1 score | Recall | Test accuracy (%) |
|-----------|-----------|----------|--------|-------------------|
| Color | 0.86 | 0.83 | 0.84 | 96.8 |
| Segmented | 0.92 | 0.91 | 0.91 | 98.14 |
| Gray | 0.78 | 0.79 | 0.77 | 95.53 |

Table 2 Metrics of apple leaf

| Type | Precision | F1 score | Recall | Test accuracy (%) |
|-----------|-----------|----------|--------|-------------------|
| Color | 0.97 | 0.97 | 0.97 | 98.31 |
| Segmented | 0.99 | 0.99 | 0.99 | 99.01 |
| Gray | 0.93 | 0.94 | 0.93 | 96.43 |

Table 3 Metrics of corn leaf

| Type | Precision | F1 score | Recall | Test accuracy (%) |
|-----------|-----------|----------|--------|-------------------|
| Color | 0.97 | 0.96 | 0.96 | 98.3 |
| Segmented | 0.94 | 0.93 | 0.94 | 98.8 |
| Gray | 0.92 | 0.93 | 0.92 | 96.46 |

6 Conclusion

The symptoms of tomato, corn and apple leaf diseases are a visible effect. Symptoms include a change in color, properties, shape of the plant when it is responded to the pathogen. Tomatoes, corn and apple are produced in large quantities. Hence, this paper is implemented to detect and identify different types of disease in the plant's leaves.

Here, the proposed solution uses a convolutional neural network model to classify the leaf diseases. The model architecture used is a convolutional neural network model with eight layers that are five convolution layers and three max pooling layers to classify the tomato, corn and apple leaves into different classes of diseases.

It can also be experimented using new model architectures for improving the performance of accuracy of the model. Thus, the above model will be helpful for the farmers to identify the disease found on the leaf. With the training and validation accuracy of 98–99%, the proposed solution is used to detect the tomato, corn and apple leaf diseases more accurately.

References

1. H. Sabrol, K. Satish, Tomato plant disease classification in digital images using classification tree, in *2016 International Conference on Communication and Signal Processing (ICCP)* (IEEE, 2016), pp. 1242–1246
2. C. DeChant, T. Wiesner-Hanks, S. Chen, E.L. Stewart, J. Yosinski, M.A. Gore, R.J. Nelson, H. Lipson, Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology* **107**(11), 1426–1432 (2017)
3. X. Zhang, Y. Qiao, F. Meng, C. Fan, M. Zhang, Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **6**, 30370–30377 (2018)
4. Y. Zhong, M. Zhao, *Research on Deep Learning in Apple Leaf Disease Recognition*, vol. 168 (Elsevier, China, 2020)
5. T.T. Santos, L.L.de Souza, A.A. dos Santos, S. Avila, Grape Detection, Segmentation, and Tracking Using Deep Neural Networks and Three-Dimensional Association, vol. 170 (Elsevier, Brazil, 2020)
6. S. Zhang, S. Zhang, C. Zhang, X. Wang, Y. Shia, Cucumber Leaf Disease Identification with Global Pooling Dilated Convolutional Neural Network, vol 162 (Elsevier, China, 2019), pp. 422–430
7. K. Thenmozhi, U. Srinivasulu Reddy, Crop pest classification based on deep convolutional neural network and transfer learning, in *Computers and Electronics in Agriculture*, vol. 164, India (2019)

Prediction of Emotional Condition Through Dialog Narratives Using Deep Learning Approach



SaiTeja Segu, Yaswanth Reddy Poreddy, and Kiran L. N. Eranki

Abstract Human emotion recognition has gained significant importance in artificial intelligence. Several machine learning studies are also been conducted to emulate the human behavior. Emotion recognition has wide range of applications from opinion mining, understanding customer feedback and also in automation. In this paper, conversational text data is used to detect the emotion which is unnoticed or ignored most of the time during conversations. Recent advances in ML and AI have given impetus to detect emotion states in conversation like listener state and previous emotion which contributes to current, as they improve the accuracy of the emotion prediction. In the current approach, a modified version of recurrent neural network (RNN) is being used to out perform the existing state-of-the-art algorithm (SOTA) conversational memory network (CMN). Since the algorithm uses text-based technique, it eventually decreases the turnaround time of the algorithm. While our proposed algorithm outperforms the existing popular CNN algorithms with a significant accuracy and Fscore metrics using IEMOCAP Dataset.

Keywords Emotion recognition · Recurrent neural network · Conversation memory network · IEMOCAP

1 Introduction

Human emotion recognition is a complex process to interpret and analysis. However, recent advances in artificial intelligence and machine learning have opened several avenues to cutting edge research in human cognition and emotion is one among them.

S. Segu · Y. R. Poreddy · K. L. N. Eranki (✉)

School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu 613402, India

e-mail: erankikiran@gmail.com

S. Segu

e-mail: segusaiteja12345@gmail.com

Y. R. Poreddy

e-mail: poreddyaswanthreddy@gmail.com

Artificial intelligence is all about understanding and replicating the human behavior [8]. The key point in replicating the human behavior is predicting the emotion like a human being does and respond according to it. Automatic emotion detection from text has attracted growing attention due to its potentially useful applications [3]. It has many applications in automation, customer satisfaction recognition in reviews and opinion mining like for an example psychologists can better assist their patients by analysing their session transcripts for any subtle emotions; reliable emotion detection can help develop powerful human–computer interaction devices; and deep emotional analysis of public data such as tweets and blogs could reveal interesting insights into human nature and behavior. In the early stages of development of this technology, convolutional neural network (CNN) is trained and used as emotion classifier by creating some sample datasets in house. The problem with CNN is that it uses back propagation to find the weight contributions. Back propagation was not efficient and requires huge dataset to train which was not available, thereby resulting in very less accuracy and making it unusable. Next phase is where long short-term memory (LSTM) was used and it was originally created for time series applications. Next phase is conversational memory networks(CMN) which uses facial expressions from video, voice modulation form audio, textual features from text to recognize the emotion. The problem with this approach is that it requires audio and video signals which is not possible in all the applications (ex: In voice assistant video, signals are not available). Most of the existing text classification methods are not reliable to predict the emotion classification [8]. Although studies have shown improved methods to predict conversational context, still the accuracy of most these system is way below expected. As most these model fail to predict the utterance of the speaker in totality [4]. In contrast, model which can predict the utterances from both speaker and listerner and evaluate the pattern of words used during the conversation. In the current approach, three aspects of the scenario are considered primarily speaker and listerner states along with their context of the utterance considering the state of preceeding utterance and its emotion. Although these three states may not be independent from each other but while modelling the emotional condition, these three states have a significant role to play. This study proposes a modified version of recurrent neural network which addresses the problem of emotion recognition using conversational text and features of the conversation (listener state and attitude). The proposed algorithm employs the RNN and bi-directional RNN as the single recurrent unit at every stage of the conversation. The rest of the paper is organized as follows: Sect. 2 discusses related work; Sect. 3 provides detailed description of dataset and methodology applied; Sect. 4 discusses about existing models and proposed model. And finally, we conclude the paper with results and conclusion.

2 Literature Review

Deep learning, which refers to deep neural networks (DNNs), is a branch of machine learning algorithms that have been widely applied to traditional artificial intelligence fields such as computer vision [4], speech recognition, and natural language process-

ing. Deep learning is capable of achieving state-of-the-art performance on emotion detection, one of the most active topics in NLP, because it allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [1]. Emotion recognition has gained wide popularity in various fields from natural language processing, psychology to cognitive science. Despite having huge volume of textual data, we still rely on audio and video signals as most models fail to predict and also require a lot of computational power to implement such models. We still do not have efficient algorithm to process video or audio signals for emotion detection [6]. We are using the text-based classification algorithms to compensate for the lower computational power with much greater accuracy for emotion detection. Hazarika et al. [3] used text-based emotion prediction model similar to the work done by Gimenez et al. [2] used contextual information for emotion recognition in multimodal setting. Recent studies [8] show RNN-based deep networks for multimodal emotion recognition are gaining popularity followed by other neural models. Human communicate through gestures and conversations which requires deep understanding of conversation. Hazarika et.al [3] argued emotional states of the conversation play a pivotal role in prediction of the emotional condition. Considering this aspect, in the current approach, two-way conversational narratives have been focused. Further, most of these conversations have a natural tendency to time bound swing as evident through recurrent network. Several RNN studies have used NLP models to envisage memory networks including query systems, language translation models [5], speech synthesis and detection [4]. Other studies by Pan et al. [7] applied in deductive and contextual speaker–listener interactions to predict the emotional state of the conversations.

3 Research Approach

In the proposed model as shown in Fig. 1 we have applied six algorithms to test the prediction accuracy of emotional conditional based on the preprocessed dialogue narratives. We found that dialogue narratives could also include *sarcasm* as well which requires more refinement to the process. However, in the current approach, we have focused on the prediction of emotional condition based on the tokenization, lemmatization and stemming applied to the dialogue narratives obtained from socio-political scenarios gathered from various newspapers and blogs.

3.1 Data Set—IEMOCAP

We have used the Interactive Emotional Dyadic Motion Capture(IEMOCAP) dataset provided by University of Southern California for this study which is specifically designed for the purpose of emotion recognition. IEMOCAP contains videos and its text of two way conversations from ten unique speakers and listeners. The dataset is

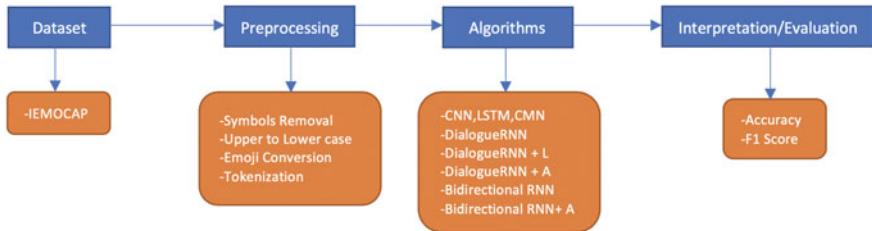


Fig. 1 Flow chart of proposed emotional prediction model

Table 1 Tabulated distribution of IEMOCAP Dataset

| Dataset | Partition | Utterance count |
|---------|-----------|-----------------|
| IEMOCAP | Training | 5810(80%) |
| | Testing | 1623(20%) |

classified into six emotions namely *happy, sad, angry, neutral, excited and frustrated*. It contains a total of 7433 records which are divided into training and testing datasets of 80% and 20%, respectively, as shown in Table 1.

3.2 Data Preprocessing Phases

Data preprocessing is an essential step because the readily available dataset is not in the desired format for algorithmic training. The preprocessing techniques employed in the study contains four stages, namely *symbols removal, upper to lower case conversion, emoji handling, tokenization*. All the above-mentioned operations are carried out using Natural Language Processing Toolkit (NLTK) in Python. Data is loaded and decoded using the *ISO-8859-1* Standard.

1. Symbol Removal

- Symbols do not contribute to the entire meaning of the sentence. These are just used for the grammatical understanding. Mainly whenever the text is typed most of the people does not care about the symbols. So, it makes sense to entirely get rid of them. This will also help in reducing the duplicates in the dictionary as shown in emoji depiction below.

Don't worry? 😕 → Don't worry 😕

2. Conversion from Upper to Lower case

- The word whether it starts with upper case or lower case convey the same meaning and emotion to the entire sentence. This will help the algorithm treat them

as same words and thereby drastically reducing the dictionary and reducing the weight contribution calculations. eg. *DON'T Worry* 😞 → *don't worry* 😞

3. Emoji Conversion

- Emojis are becoming the key of expression. Most of the emotion is expressed using emojis. It is better to handle them instead of ignoring them. The method which this study uses is replacing the emojis with its UNICODE. UNICODE is a universally accepted unique code for emojis. Emoji is converted into UNICODE and treated as a single character like words/alphabets.eg. *don't worry* 😞 → *don't worry U+1F601*

4. Tokenization

- Most the deep Learning algorithms work on numbers rather than text. In this stage, the sentences are divided into list of tokens and are replaced with its index in dictionary. To achieve this, `sent_tokenise` method from `nltk.tokenize` package is used.

4 Proposed Emotion Prediction Model

As mentioned, this study uses existing methods like CNN,LSTM,CMN and proposed methods like Dialogue-RNN, Dialogue-RNN + Listener state, Dialogue-RNN + Attitude, Bidirectional RNN, Bidirectional RNN + Attitude for the comparisons. Some variants of CNN models used in study as shown in Fig. 2.

4.1 Convolution Neural Network (CNN) Model

Convolutional neural network uses a hierarchical structure rather than sequential structure. Hierarchical structure is divided into different layers of neurons. The layers that are stacked in the hierarchical structure are convolution layer, pooling layer and the fully connected dense layer. The convolution layer contains convolution filters which are used to extract the features in the text which are useful for emotion recognition. Convolution filter or *kernel* is applied to generate *feature map* to predict emotions. The pooling layer contains the max pooling filter which is used to identify maximum contributing features to the emotion from previously stacked layer of neurons. Finally, the dense layer of neurons is used to combine all the previously extracted features with its corresponding weights to get the final emotion.

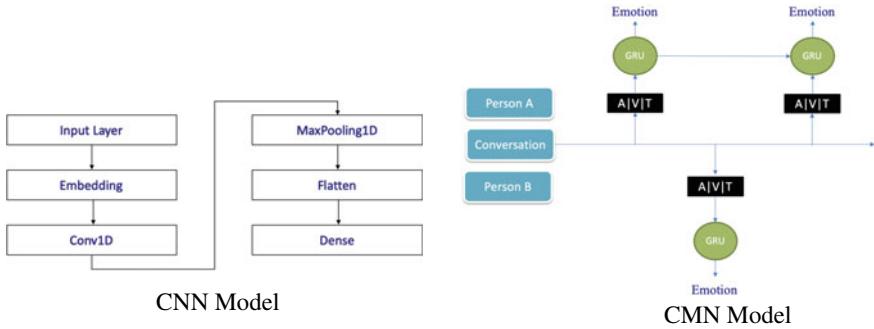


Fig. 2 Variants of CNN models

4.2 Long Short-Term Memory (LSTM) Model

LSTM uses a sequential structure which LSTM units. Each LSTM unit consists of three gates, namely forget gate, input gate and output gate. Forget gate is used to decide whether the previous LSTM unit is important or not. Input gate adds present stage information to the current LSTM unit. Output is used to pass the data from one LSTM unit to other. The architecture diagram is shown in Fig. 3.

$$\text{ForgetGate: } f_t = \delta(W_f * [h_{(t-1)}, x_t] + b_f) \quad (1)$$

$$\text{Input: } i_t = \delta(W_i * [h_{(t-1)}, x_t] + b_i) \rightarrow C_t = \tanh(W_C * [h_{(t-1)}, x_t] + b_C) \quad (2)$$

$$\text{Output: } O_t = \delta(W_O * [h_{(t-1)}, x_t] + b_O) \rightarrow h_t = O_t * \tanh(C_t) \quad (3)$$

where h_{t-1} is the previous unit emotion, x_t presents input value, W is weights of specific units, b is the bias.

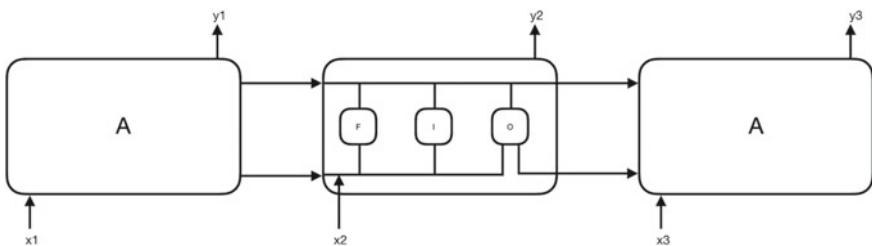


Fig. 3 Architecture of LSTM model for prediction of emotional condition, middle node shows three gates, namely—F-forget, I-input, O-output. Functionality of these gates is shown in Eqs. 1–3

4.3 Conversational Memory Network Model

Conversational memory network is state-of-the-art algorithm used for emotion recognition nowadays. CMN requires video, audio and text from the listener speaker configuration. It uses gated recurrent units for every single sentence to identify the emotion. As CMN deals with the video, it requires CUDA-enabled GPU to perform all the required operations and it ignores the main features of the conversation like listener state as shown in Fig. 4. Type of content used for analysis have been shown as A for audio, V for video and T for text. CMN model applies the necessary activation function to perform the analysis and generates a predicted list of emotional conditions from the provided dataset.

4.4 Dialogue RNN

Dialogue RNN is the special purpose RNN which is especially designed for the purpose of emotion recognition in conversations as shown in Fig. 4. The recurrent neural network is used and a unique RNN is being for listener and speaker to find the emotion. It uses a two-layer perceptron in every single RNN unit with final argmax layer to find max probable emotion from six probable emotions.

- Extended dialogue RNN with listerner state ($D - \text{RNN}_t$)

$$l_t = \text{ReLU}(Wl * e_t + l + b_l) \rightarrow P_t = \text{softmax}(W * l_t + b_2) \quad (4)$$

$$Y_t = \text{argmax}(P_{t[i]}) \quad (5)$$

where W —weights of respective attributes, e_t is tokens of the present sentence, b_l and b_2 are bias of respective functions L is listener state while S is speaker on conversation.

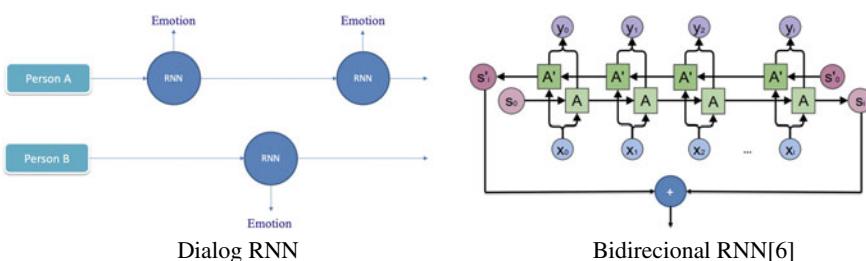


Fig. 4 Architecture of bidirectional RNN and dialogue RNN models

- Dialogue RNN with Attitude ($D - \text{RNN}_a + \text{Att}$)

$$l_t = \text{ReLU}(Wl * e_t + a + b_l) \rightarrow P_t = \text{softmax}(W * l_t + b_2) \quad (6)$$

$$Y_t = \text{argmax}(P_{t[i]}) \quad (7)$$

Attitude: tell us the emotion of the previous dialogue. It is very similar to *Dialogue RNN + Listener State* but instead of considering listener state consider the *emotion* of the speaker in the *previous* dialogue.

- Bidirectional dialogue RNN (Bi-RNN_t)-Bidirectional dialogue RNN uses bidirectional RNN as the backbone. Bidirectional RNN contains two Unidirectional RNNs which are used for forward and backward sequence respectively. Outputs from two RNNs are concatenated to get the final emotion in the current situation.
- Bidirectional dialogue RNN with attitude (Bi-RNN_a + Att)—Similar to Bi-RNN_t, but in every emotion prediction, all the other attitudes in the conversation are also considered to better understand the context.

$$\text{FSeq} : l_t = \text{ReLU}(Wl * e_t + a + b_l) \rightarrow P_{t1} = \text{softmax}(W * l_t + b_2) \quad (8)$$

$$\text{BSeq} : l_t = \text{ReLU}(Wl * e_t + a + b_l) \rightarrow P_{t2} = \text{softmax}(W * l_t + b_2) \quad (9)$$

$$Y_t = \text{argmax}(P_{t1} + P_{t2}) \quad (10)$$

where W are weights of respective attributes, e_t is tokens of the present sentence, b_l and b_2 are bias of respective functions L is listener and S would be speaker who speaks and a is the attitude observed while speaking the previous dialogue

5 Results and Discussion

5.1 Comparative Analysis Among Dialogue RNN and CNN Models

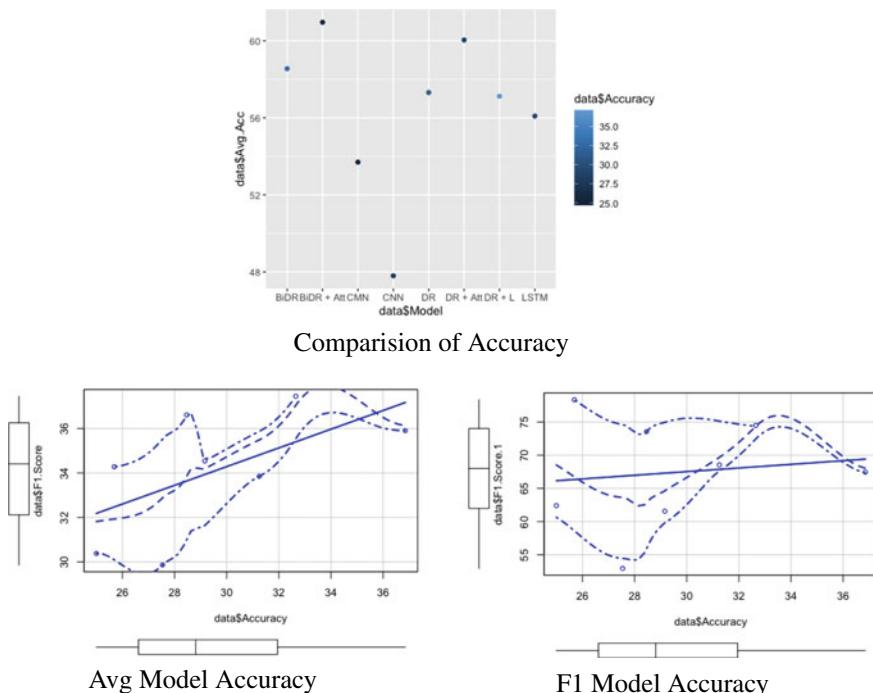
In this section, we discuss compared results among *D-RNN* and its variants against CNN, LSTM and CMN (SOTA) using textual data as shown in Table 2. We found the our results similar to on studies showing *D-RNN* performs better than all the base line methods including conversational memory networks (CMN) on IEMOCAP dataset. From Table 2, we can also find that *D-RNN* performs lot better than CMN model by 2.47% accuracy and 3.82% F1-score results on an average. While dealing with six unbalanced emotion labels, model performance of individual labels is also computed. *D-RNN* shows better scores on five emotional classes with significant accuracy and F1 scores as compared to CMN model. While for frustrated emotion, we noticed a drop in metrics by *D - RNN* compared to CMN. Otherwise, we found D-RNN lot better than CMN considering classifiers metrics for frustrated emotion. However, from Table 2, we can say that other variants of *D - RNN*, like *Bi - RNN* show significant performance than CMN for frustrated class.

Table 2 IEMOCAP data emotional conditions

| Model | Happy Accu. F1 | Sad Accu. F1 | Angry Accu. F1 | Excited Accu. F1 | Frustrated Accu. F1 | Avg Accu. F1 |
|------------|---------------------|-------------------|---------------------|-----------------------|--------------------------|-------------------|
| CNN | 27.54 29.86 | 56.84 52.94 | 61.17 52.44 | 46.15 51.02 | 61.45 53.87 | 47.80 46.99 |
| LSTM | 29.16 34.54 | 58.35 61.56 | 57.89 55.87 | 68.43 57.45 | 65.14 57.51 | 45.23 53.32 |
| CMN [3] | 25.00 30.38 | 55.92 62.41 | 61.76 59.83 | 55.52 60.25 | 71.13 60.69 | 56.56 56.13 |
| DR + L | 36.85 35.90 | 64.35 67.45 | 62.65 61.85 | 59.20 62.34 | 64.21 60.43 | 57.12 56.94 |
| DR + Att | 28.46 36.61 | 66.52 73.56 | 67.75 66.24 | 70.90 68.61 | 62.36 62.48 | 60.04 61.04 |
| BiDR + Att | 25.69 34.28 | 76.45 78.35 | 64.71 65.28 | 80.27 71.86 | 61.15 58.91 | 60.96 61.39 |

5.2 Comparision Among Dialogue RNN Varients

Based on results shown in Fig. 5 and Table 2, using explicit listener state with *D-RNN* shows a drop in performance as compared to regular *D-RNN*. However, we found an exception for *happy* emotion where D-RNN exceeds regular D-RNN performance by 2.05% F1-score. We believe the performance gain occurs when

**Fig. 5** Compartion of emotional accuracy among CNN models and proposed model

listener becomes attentive to the speaker conversation and responds back. As a result in D-RNN, as speaker speaks his/her emotional state e_t with attitude context c_t are captured through conversational dialogue textual data which includes current and previous utterances making listener state update unneccesary in $D-RNN$.

1. ***BiDialogue-RNN*** (Fig. 4): Based on the Eqs. 1–10. *Bi-RNN* utilizes current state and predicts future emotional states, providing improved performance compared to $D-RNN$. This is confirmed in Table 2, where *Bi-RNN* exceeds $D-RNN$ in terms of accuracy and F1 scores of the dataset.
2. ***Dialogue-RNN + Att***: $D-RNN + Att$ also uses information from the future conversations captured through two dialogues. Considering both past and future conversations by mapping them with the current emotional state and computing the attitude score based on the generated emotion. As a result gives better performance than $Bi-RNN_{t1}$, showing 1.43% F1-score.
3. ***BiDialogue-RNN + Att***: This version of RNN provides the final emotion state from current emotional state of $B-RNN_{t1}$. As a result shows better performance than both $B-RNN$ and $D-RNN_{t1} + Att$ which is evident from Table 2 showing a lot better performance as compared to other models on both datasets. Results show 5.26% higher F1-score on average than existing CMN and 2.59% higher F1-score than $D-RNN$ for IEMOCAP dataset.

6 Conclusions

In this paper RNN-based approach was used for prediction of emotion in conversations. Comparison of existing methods to our proposed model to identify emotion state from two-way dialogues performs better with hyper-parameters(learning rate, regularity etc.). Our model surpasses the existing CNN, RNN and CMN, models in textual settings. Our method is designed to be scalable for multi-conversational narratives which we plan to implement as our future work.

References

1. M.B. Akçay, K. Oğuz, Speech emotion recognition: emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Commun.* **116**, 56–76 (2020)
2. M. Giménez, J. Palanca, V. Botti, Semantic-based padding in convolutional neural networks for improving the performance in natural language. A case of study in sentiment analysis. *Neurocomputing* **378**, 315–323 (2020)
3. D. Hazarika, S. Poria, A. Zadeh, E. Cambria, L.P. Morency, R. Zimmermann, Conversational memory network for emotion recognition in dyadic dialogue videos, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), pp. 2122–2132

4. N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, E. Cambria, Dialoguernn: an attentive rnn for emotion detection in conversations, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33 (2019), pp. 6818–6825
5. M. Nabati, A. Behrad, Video captioning using boosted and parallel long short-term memory networks. *Comput. Vis. Image Underst.* **190**, 102840 (2020)
6. C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, S. Carter, Zoom in: An introduction to circuits. *Distill* **5**(3), e00024–001 (2020)
7. B. Pan, Y. Yang, Z. Zhao, Y. Zhuang, D. Cai, Bi-decoder augmented network for neural machine translation. *Neurocomputing* **387**, 188–194 (2020)
8. D. Peng, M. Zhou, C. Liu, J. Ai, Human-machine dialogue modelling with the fusion of word-and sentence-level emotions. *Knowl. Based Syst.* **192**, 105319 (2020)

Software Requirements Classification and Prioritisation Using Machine Learning



Pratvina Talele and Rashmi Phalnikar

Abstract Software Development Lifecycle (SDLC) is a systematic process used to achieve high quality software that meets customer requirements. During SDLC requirements, engineering plays an important role. Prioritisation helps to focus on the most important requirements in terms of importance, cost, penalty, time and risk. Stakeholders (users, developers) of the software product identify requirements. The two major activities of requirement engineering process are requirements classification and requirements prioritisation. Sometimes requirement mentioned by stakeholder can be of both types, i.e. functional and non-functional. So it is challenging to classify requirements separately in two different categories. There are many fundamental prioritisation techniques available to prioritise software requirements. In this paper, we have compared existing requirements prioritisation techniques based on ease of use, speed, scalability and accuracy. Our literature study suggests that the appropriate requirements prioritisation technique has to be selected that can help software developer to minimise the risk, penalty. In automating various tasks of software engineering, machine learning (ML) has shown useful positive impact. This paper discusses the various algorithms used to classify and prioritise the software requirements. The results in terms of performance, scalability and accuracy from different studies are contradictory in nature due to variations in research methodologies and the type of dataset used. Based on the literature survey conducted, we propose a new architecture that will use both types of datasets, i.e. Software Requirement Specifications (SRS) and user text reviews to create a generalised model. Our proposed architecture will attempt to extract features which can be used to train the model using ML algorithms. The ML algorithms for classifying and prioritising software requirements will be developed and assessed based on performance, scalability and accuracy.

P. Talele (✉) · R. Phalnikar

School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India

e-mail: pratvina.talele@mitwpu.edu.in

R. Phalnikar

e-mail: rashmi.phalnikar@mitwpu.edu.in

Keywords Machine learning · Requirement engineering · Software requirements

1 Introduction

The aim of Software Development Lifecycle (SDLC) is to produce high quality software. SDLC is a standard process for developing software that ensures the correctness of the software built. The software development should be completed by the deadline and in pre-defined cost. Requirement engineering (RE) is a very important phase of the SDLC. It is the process of gathering, defining, documenting and maintaining the requirements of the software. The software requirements are explanation of features and functionalities of the software used while developing software. RE Process consists of the main steps as shown in Fig. 1.

Requirements elicitation is the step where requirements for software are gathered by communicating with people involved in the software development and users of the software. The techniques used for requirements elicitation include interviews, brainstorming, task analysis, etc. Software requirements categories (Table 1) are functional requirements and non-functional requirements. Functional requirements are the requirements which should be performed by software. A non-functional requirement defines the quality attribute of a software system.

Requirement specification phase is used to construct formal software requirement model. All functional as well as non-functional requirements are specified by this

Fig. 1 Requirement engineering process

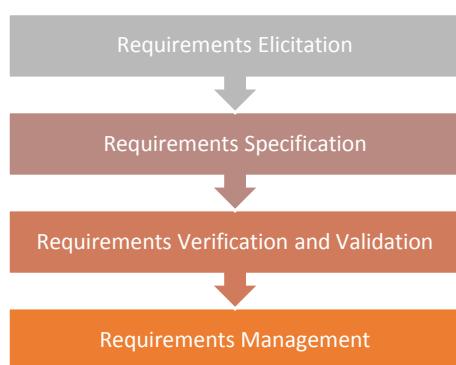


Table 1 Software requirement categories

| | |
|----------------------------------|---|
| Functional requirements | Calculations, technical details, data manipulation, processing |
| Non-functional requirements [29] | Performance, scalability, capacity, availability, reliability, recoverability, maintainability, serviceability, security, regulatory, manageability, environmental, data integrity, usability |

model. During this phase, if more knowledge about the software is required, it can again trigger the first phase, i.e. elicitation process. While developing a formal software requirement model, requirement prioritisation technique can be used. While planning a product of single and multiple releases, prioritisation is an important step for taking good decisions. Using different requirements prioritisation techniques, requirements are arranged according to their importance such as cost and time. [1]. Requirements verification phase is used to check whether software correctly implements both software requirements or not. Requirements validation phase is used to ensure that software has been built is traceable to user requirements. If requirements are not validated, errors in requirement classification and prioritisation would propagate to the successive stages resulting in a lot of modification and rework. Requirement management phase takes care of the changing nature of requirements. Software requirement specification should be as modifiable as possible so that new changes in requirements specified by the user can be incorporated at the later stages.

The Standish Group Chaos Report 83.9% of IT projects partially or completely fail [2]. Different factors found in failed projects include: Incomplete requirements, poor classification of requirements, poor requirements prioritisation techniques and lack of user involvement. For example, to develop the Armed forces software tool it took \$2.7 billion which was not convenient and did not fulfil the purpose of the application due to poor requirements classification as prioritisation process [3]. Nowadays, various jobs of software engineering are automated using ML. It is especially used in the domains wherever big data are available and a requirement for finding some kind of dependability exists, or wherever there is a requirement of program that should adapt the changes. Requirements engineering is an important phase of software engineering and it gives the idea of using suitable ML methods for various jobs of requirement engineering. As requirements specified in documents are mostly in natural language, ML can be beneficial in classification of software requirements and software requirements prioritisation tasks of requirement engineering by emulating human processing [4, 5].

The process of classifying software requirements in a textual document using ML algorithm involves three major phases—text preparation, learning and evaluation. In text preparation phase, the pre-processing of text of the requirements and selection of features are involved. In learning phase, classification of software requirements is done by applying ML algorithms. The evaluation phase is used to assess applied ML algorithm. The existing text pre-processing techniques such as stop word removing, stemming and feature selection technique such as bag of words are used [6]. The existing ML algorithms such as Naïve Bayes (NB), support vector machine (SVM) are used to classify software requirements whose accuracy is inferior [7]. Some authors do not consider all categories of NFRs [8–13]. Software quality is decided by the capability to fulfil the requirements of the customers. So in order to develop the quality software, it is very important to provide prioritisation to user's requirements to select the best possible set of requirements from a set of all requirements [14]. There is issue of assigning priority values to requirements using pair-wise comparisons amongst software requirements with existing requirements prioritisation process. This issue is in terms of increase in time complexity, decrease in scalability and

accuracy. It can be addressed by using ML techniques [5]. This study provides an outline of the NLP and ML techniques used for the support of RE tasks (classification and prioritisation of software requirements) and an architecture that can be used to bridge the gap between research and state of the art.

2 Related Work

Requirement engineering is an important step of SDLC. Significant research and empirical studies have taken place in the area of first two phases of requirement engineering, i.e. requirements classification [4] and requirements prioritisation [5].

To automate the process of identifying requirements [4, 7] and assigning priorities to requirements from requirement documents using ML algorithms [14] and different prioritisation techniques [5], a number of attempts have been made. To identify requirements, the first step made is text preparation using natural language processing (NLP). Second step is classify requirements using ML algorithms. Once the requirements are classified, priority values can be assigned using prioritisation techniques.

2.1 *Text Preparation Phase*

This phase has two steps—text pre-processing and feature selection

Text Pre-processing—Documents containing word-based requirements are provided as input to this step. To pre-process the text various, natural language processing techniques are applied to these documents. Mostly used NLP techniques to pre-process are stemming, tokenization, part of speech, stop words removal, N-gram, lemmatization. Tokenization is an important task when an input is text. Tokenization is splitting a sentence, paragraph, phrase, or an entire text document into smaller units, called as tokens. This technique is used in [5, 6, 8]. Words are reduced to their word root, base or stem form in stemming technique. For example, clicked and clicking reduced to the same stem word ‘click’. This technique is used in [5, 8, 10]. Part of speech technique assigns tags to verb, noun, adjective and so on. This technique is used in [13, 15]. Stop words removal technique is used to remove auxiliary verbs, conjunctions and articles in sentences. For example do, be, or, and, a, an, the and so on. It is the mostly used technique [5, 8, 10, 13, 16]. In N-gram technique, each given string can be separated using this technique into subsequent N items. N can be letters, words or statements. Mostly used N-grams are bigrams (2-grams) and trigrams (3-grams). This technique is used in [13]. Lemmatization technique is used to determine lemma word, adjectives of every word, the singular form of nouns and an infinitive form of verbs. For example, the words ‘goes’, ‘gone’, ‘going’ and ‘went’ will map to ‘go’. This technique is used in [10, 13].

Feature selection—The model can be trained by taking input as features. This step, feature selection, is used to convert pre-processed requirements document into features. Mostly used feature selection techniques are: bag of words and term frequency-inverse document frequency (TF-IDF). Bag of words technique is used to calculate the occurrence of every word in a file or sentence. It is used in [10, 13]. TF-IDF technique is used to calculate the frequency of a word that occurs in the file or sentence, inverse of the number of times the word occurs in the quantity. It is the mostly used technique [5, 10, 11].

2.2 Classification of Requirements Methods

A number of studies have used ML algorithms to classify software requirements. The learning phase of a model is done after feature selection process. The supervised learning algorithms, unsupervised learning algorithms and semi-supervised learning algorithms are frequently used to classify the requirements into FRs and NFRs.

Supervised learning algorithms need a collection of requirements that can be accurately categorised to functional requirements, non-functional requirements. The number of requirements used to train the model differs in the specified studies. 58 requirements are used to train its data and focused on security NFRs [5]. Many supervised learning algorithms are used in this area of research such as SVM [11, 13, 17], NB [10, 11, 18], Adaptive Boost [13], Random Forest [13], Decision Tree (DT) [8, 10, 11, 13, 19], K-Nearest Neighbour [11, 17].

Similarly researchers have used unsupervised learning algorithms such as K-means [18, 20], LDA [12, 18, 19, 21], Hierarchical Agglomerative Clustering algorithm [18, 20], single link clustering algorithm [16] and Bi-term [19]. Performance of these algorithms in terms of accuracy is poor [7].

Semi-supervised learning algorithms are used in this research area such as self-training [11], RAndom Subspace Method for Co-training (RASCO) [11], Relevant RAndom Subspace Method for Co-training (Rel-RASCO) [11] and active learning [17]. Self-training, RASCO and Rel-RASCO algorithms are used with supervised learning algorithms such as SVM, NB, DT and KNN to classify requirements. While classifying requirements using these algorithms, all categories of NFRs are not considered [11]. 3 SRS are considered as dataset [17] and user reviews for app store are used as dataset [11]. Using these algorithms, human efforts are reduced in labelling training instances. But recall levels were ranging from 54.6% to 81.9% and precision levels were ranging from 73.9% to 92.4% on average for different categories of NFRs of three projects [17] and transductive and inductive accuracy levels are less than 70% [11].

Many more other ML algorithms are also used such as Stochastic Gradient Descent (SGD) [16], Goal Oriented Requirements Elicitation (GORE) [22, 23]. Convolutional Neural Network (CNN) is also used classify the requirements [16]. After applying CNN and SGD to different documents to test the learned model, received results are inferior [16].

2.3 Requirements Prioritisation Methods

Important Research and experiential studies is conducted in the area of requirements prioritisation [5]. The requirements prioritisation methods can be categorised into two groups. The first group of fundamental methods such as cumulative voting, AHP, cost-value method, numerical assignment, priority grouping and the second group of methods which are a combination of the fundamental methods. It is difficult to say that whether the second group is validated and are in use in practice or not. Since applying specific set of methods to similar requirements which are reasonably well analysed and are at the similar abstraction level in an experiment has been a difficult step in practice for research.

Analytic hierarchy process (AHP) is an efficient statistical technique based on relative assessment that has been used to prioritize software requirements in software community. To consider both quantitative and qualitative facets of a decision, the AHP is an influential and adaptable decision-making process that helps individuals to assign the priorities and take the preeminent decision. By dropping multifaceted decisions to a sequence of one on one comparisons, the best decision can be taken by decision makers using AHP. With AHP, one can synthesise the results, which provide a clear rationale for choosing the candidate requirements. During the process, $n \times (n - 1)/2$ comparisons are mostly made at every hierarchy level, where n is the number of requirements. This is the drawback in this process because if the number of requirements increases, the number of comparisons increases with a magnitude of $O(n^2)$. It used in [14, 15, 22, 24].

The cumulative voting (CV) or the 100-dollar test is a simple technique in which the stakeholders are provided 100 unreal dollars those needs to be considered as units. This technique is based on the ratio scale. This prioritisation technique is complex in terms of sophistication and fine in terms of granularity. These imaginary units could be different aspects (e.g. money cost of implementation, importance, penalty, hours etc.). The advanced version of CV, fuzzy hierarchical cumulative s (FHCV) is used as the base algorithm to propose new prioritisation technique Adaptive Fuzzy Hierarchical Cumulative Voting (AFHCV) [25]. AFHCV considers that requirements can change during the development of software. So it accepts the change in requirements at run-time. It analyses these requirements again and assigns priorities to these requirements again. As it repeats process of analysing and assigning priorities to requirements again and again, its performance in terms of time complexity increases as well as it faces scalability issue.

Similarly ML algorithms such as Gradient Descent Rank (GDRank) [14], Apriori [26] are used as prioritisation techniques. For comparison purpose of these new algorithms used as prioritisation techniques, AHP [14, 15] and Case Base Rank (CBRank) [14] algorithms are considered. Must have, should have, could have, and would not have (MoSCoW) is the prioritisation method that decides which requirement must be considered, should be available, could be considered and not to be consider. Fuzzy-based MoSCoW Method [23] is uses the fundamental prioritisation

Table 2 Existing requirements prioritisation techniques

| Technique | Ease of use | Speed | Scalability | Accuracy |
|---------------------------|-------------|--------|-----------------|-----------------|
| Numerical assignment [30] | Complex | Slow | Not scalable | Less accuracy |
| MoSCoW [23] | Complex | Slow | Not scalable | Less accuracy |
| Priority groups [31] | Complex | Medium | More scalable | Medium accuracy |
| Bubble sort [31] | Easy | Slow | Not scalable | Less accuracy |
| AHP [14] | Easy | Slow | Not scalable | Medium accuracy |
| Hundred dollar [25] | Easy | Medium | Medium scalable | High accuracy |

algorithm MoSCoW. This method is not evaluated. It has to be validated with a great number of requirements. Existing requirements prioritisation techniques are listed in Table 2.

2.4 Limitations of Existing Techniques

Performance of ML algorithms such as SVM, Naïve Bayes, KNN, self-training is inferior in case of large datasets. Existing work do not classify all categories of NFRs such as security, performance, usability. Most of the studies those are classifying requirements have used dataset as dummy SRS or user reviews collected for specific product. After applying the learned model to another SRS, received results are poor. Even though fundamental prioritisation techniques such as AHP, CV are used mostly, their performance in terms of scalability, accuracy and complexity is inferior. New prioritisation techniques uses fundamental prioritisation techniques such as AFHCV [25] faces scalability issues and Fuzzy based MoSCoW [23] is not evaluated.

3 Proposed Work

The main objective of the software development is to achieve the high quality of a software. Requirements classification and requirements prioritisation are the most significant parts of software development. These parts enable the software development in planned time using resource and time appropriately. Literature review points to mostly used ML classification and prioritisation methods being researched and gap between research and state of the art.

We have proposed the architecture as shown in Fig. 2.

Step 1: Text preparation—Public datasets are available in the form SRS, user reviews. Requirements mentioned in these datasets are stated in natural language. To apply ML algorithms, features are to be extracted. Text has to be pre-processed first using NLP techniques such as tokenization, removal of stop words, lemmatization. Feature

extraction is the second phase of text preparation. It can be done using feature extraction techniques such as bag of words, TF-IDF. As requirements are mentioned in the natural language, selection of appropriate text pre-processing techniques and feature extraction techniques is an important step to prepare input to be provided to ML algorithm and designing a framework that can bridge gap between research and practice.

Step 2: Classification of requirements—Classification technique needs to be designed and developed to train the model which classifies the requirements into FRs and NFRs.

Step 3: Evaluation of classification technique—Once the model is trained, it can be tested to check the performance of the technique.

Step 4: Prioritisation of requirements—Priority values can be assigned to different requirements based on their importance. This can be done using a method that has to be designed and developed to train the model.

Step 5: Evaluation of prioritisation technique—Once the model is trained, it can be tested to check the performance of the technique.

In summary, the outcomes shown that there is lack of labelled datasets. Datasets (software requirements) are mentioned in natural language. So, there is challenge to

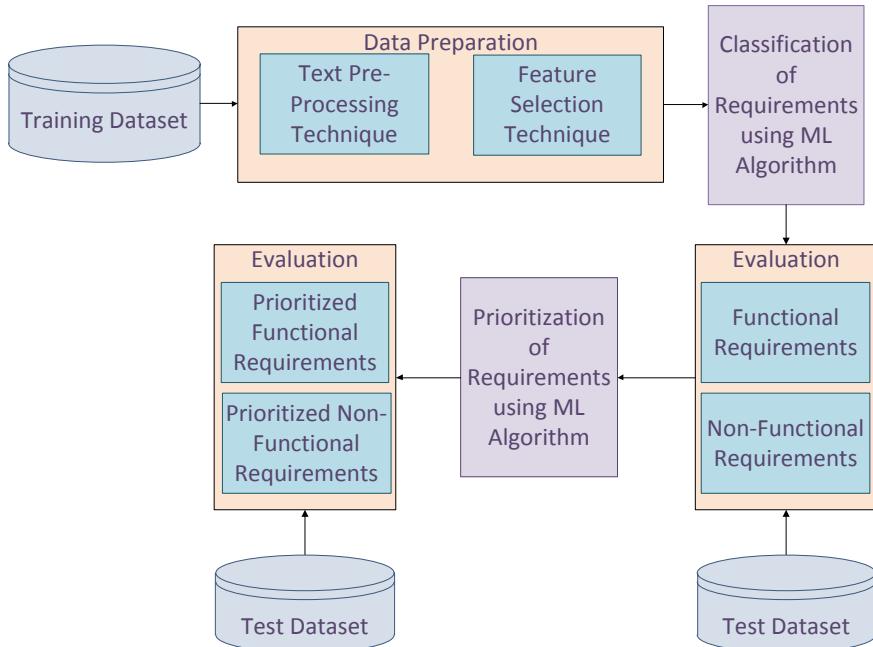


Fig. 2 Proposed architecture

extract features using correct NLP technique. To determine the classification accuracy, features are vital aspect, as the data is used to train the model using ML. The review also shown that still limitations exist although there are existing requirements classification and requirements prioritisation techniques. These limitations include accuracy, scalability and performance.

Paper considers SRS [27] and mobile app user reviews [28]. The expected outcome of our proposed work is to achieve improved performance in term of accuracy, scalability and performance.

4 Conclusion

Most of the software development organizations currently use fundamental methods to classify requirements and prioritize requirements such as AHP, CV, MoSCoW which do not provide scalability and accuracy. Methods proposed in the research are found to be complex for usage in practice as these cannot be applied to any kind of datasets such as SRS, reviews collected from users for different purposes.

Most of the software companies currently uses simple methods which do not provide accurate, flexible and scalable methodology for requirements classification and prioritisation in software development in practice.

With the understanding of the different aspects of classification and prioritisation of requirements, new algorithms need to be developed as solutions to bridge the gap between research and practice.

In our proposed work, we will attempt to extract features which would serve as a generalised solution applicable to all domains. The ML algorithm to classify the software requirements will be designed, developed and compared with other solutions. Similarly, ML algorithm to prioritize the software requirements will be developed and assessed. Once both algorithms are developed and assessed, the framework can be developed. This framework can used by different software companies.

References

1. P. Achimugu, A. Selamat, R. Ibrahim, A clustering based technique for large scale prioritization during requirements elicitation, in *Recent Advances on Soft Computing and Data Mining*, vol. 287, ed. by T. Herawan, R. Ghazali, M. Deris (Springer, Cham, Switzerland, 2014), p. 3039
2. <https://www.opendoorerp.com/the-standish-group-report-83-9-of-it-projects-partially-or-completely-fail/>
3. C. Hoskinson, Army's faulty computer system hurts operations, Politico, 2011, (Online). Available: <http://www.politico.com/news/stories/0611/58051.html>
4. T. Iqbal, P. Elahidoost, L. Lucio, A Bird's eye view on requirements engineering and machine learning, <https://doi.org/10.1109/apsec.2018.00015>
5. F. Hujainah, R.B.A. Bakar, M.A. Abdulgabber and K.Z. Zamli, Software requirements prioritisation: a systematic literature review on significance, stakeholders, techniques and challenges, vol. 6 (IEEE Access, 2018), pp. 71497–71523

6. A. Khan, B. Baharudin, L.H. Lee, K. Khan, A review of machine learning algorithms for text-documents classification. *J. Adv. Inf. Technol.* **1**(1), 4–20 (2010)
7. M. Binkhonain, L. Zhao, A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Syst. Appl.* <https://doi.org/10.1016/j.eswa.2019.02.031>
8. R. Malhotra, A. Chug, A. Hayrapetian, R. Raje, *Analyzing and evaluating security features in software requirements* (Innovation and Challenges in Cyber Security, Noida, India, 2016)
9. P. Singh, D. Singh, A. Sharma, *Rule-based system for automated classification of non-functional requirements from requirement specifications* (Advances in Computing, Communications and Informatics, Jaipur, India, 2016)
10. M. Lu, P. Liang, Automatic classification of non-functional requirements from augmented app user reviews, in *Proceedings of 21st International Conference on Evaluation and Assessment in Software Engineering*, Karlskrona, Sweden, 2017
11. R. Deocadez., R. Harrison, D. Rodriguez, Automatically classifying requirements from app stores: a preliminary study, in *IEEE 25th International Requirements Engineering Conference Workshops*, Lisbon, Portugal, 2017
12. J. Zou, L. Xu, M. Yang, X. Zhang, D. Yang, Towards comprehending the non-functional requirements through developers' eyes: an exploration of Stack Overflow using topic analysis. *Inf. Softw. Technol.* **84**, 19–32 (2017)
13. Z. Kurtanović, W. Maalej, Automatically classifying functional and non-functional requirements using supervised machine learning, in *IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, 2017, pp. 490–495
14. D. Singh, A. Sharma, Software requirement prioritization using machine learning, in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, (SEKE, 2014), pp. 701–704
15. F. Shao, R. Peng, H. Lai, B. Wang, DRank: a semi-automated requirements prioritization method based on preferences and dependencies. *J. Syst. Softw.* (2016). <https://doi.org/10.1016/j.jss.2016.09.043>
16. J. Winkler, A. Vogelsang, Automatic classification of requirements based on convolutional neural networks, in *IEEE 24th International Requirements Engineering Conference Workshops (REW)*, Beijing, 2016, pp. 39–45. <https://doi.org/10.1109/rew.2016.021>
17. C. Li, L. Huang, J. Ge, B. Luo, V. Ng, Automatically classifying user requests in crowdsourcing requirements engineering. *J. Syst. Softw.* **138**, 108–123 (2017)
18. Z.S.H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, K. Schneider, What works better? A study of classifying requirements. Paper presented at IEEE 25th International Requirements Engineering Conference Workshops, Lisbon, Portugal, 2017
19. R. Jindal, R. Malhotra, A. Jain, Automated classification of security requirements, in *Advances in Computing, Communications and Informatics*, Jaipur, India, 2016
20. A. Mahmoud, G. Williams, Detecting, classifying, and tracing non-functional software requirements. *Requirements Eng.* **21**, 2016. <https://doi.org/10.1007/s00766-016-0252-8>
21. I. Morales-Ramirez, D. Muñante, F. Kifetew, A. Perini, A. Susi, A. Siena, Exploiting user feedback in tool-supported multi-criteria requirements prioritization, in *Proceedings of IEEE Region 10 Humanitarian Technologies Conference (R10-HTC)*, 2017, pp. 424–429
22. M. Sadiq, T. Hassan, S. Nazneen, AHP_GORE_PSR: applying analytic hierarchy process in goal oriented requirements elicitation method for the prioritization of software requirements, in *Proceedings of 3rd IEEE International Conference*, 2017, pp. 1–5
23. K.S. Ahmad, N. Ahmad, H. Tahir, S. Khan, Fuzzy_MoSCoW: a fuzzy based MoSCoW method for the prioritization of software requirements, in *Proceedings of International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2017, pp. 433–437
24. L. Alawneh, Requirements prioritization using hierarchical dependencies, in *Information Technology_New Generations*. Cham, Switzerland: Springer, 2018, pp. 459–464
25. B.B. Jawale, G.K. Patnaik, A.T. Bhole, Requirement prioritization using adaptive fuzzy hierarchical cumulative voting, in *2017 IEEE 7th International Advance Computing Conference (IACC)*, Hyderabad, 2017, pp. 95–102. <https://doi.org/10.1109/iacc.2017.0034>

26. R. Anand, R. Dinakaran, M., Handling stakeholder conflict by agile requirement prioritization using Apriori technique. *Comput. Electr. Eng.* **17**(61), 126–136 (2017). <https://doi.org/10.1016/j.compeleceng.2017.06.022>
27. <http://fmt.isti.cnr.it/nlreqdataset/>
28. <http://soolinglim.com/datasets/?wordpress#app>
29. L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos, *Non-Functional Requirements in Software Engineering, International Series in Software Engineering*, vol. 5 (Springer, Heidelberg, 1999), p. 476
30. A.S. Danesh, R. Ahmad, Study of prioritization techniques using students as subjects, in *ICIME '09: Proceedings of the 2009 International Conference on Information Management and Engineering*, Washington, DC, USA: IEEE Computer Society, 2009, pp. 390–394
31. J. Karlsson, C. Wohlin, B. Regnell, An evaluation of methods for prioritizing software requirements. *Inf. Softw. Technol.* **39**(14–15), 939–947 (1998)

Comparison of Hidden Markov Models and the FAST Algorithm for Feature-Aware Knowledge Tracing



Georg Gutjahr, Pantina Chandrashekhar, M. Gowri Nair, Mithun Haridas, and Prema Nedungadi

Abstract In many Indian rural schools, individual students do not receive adequate attention due to the high student–teacher ratio. It is an onerous task for teachers to assess their students’ knowledge levels, and identify their deficient areas of learning. An Intelligent Tutoring System (ITS) enables the teacher to create a report on topics, which students need to study in more detail. Knowledge tracing approaches are a good option for the generation of such reports. For the current paper, we analyze first-grade students from 28 schools, who use Amrita Learning, an ITS developed by Amrita University. There were 211,275 responses obtained in a single academic year. The performance of three knowledge tracing approaches were compared using this dataset: standard Bayesian Knowledge Tracing, Feature-Aware Student Knowledge Tracing (FAST) and Hidden Markov Model (HMM). We find that the HMM approach marginally outperforms the other two methods.

Keywords Intelligent tutoring system · Hidden Markov model · Feature-aware student knowledge tracing (FAST)

1 Introduction

Personalized learning approach addresses the needs and capabilities of individual students and provides a customized learning style for each student [1]. Due to the high student–teacher ratio in many Indian schools, individual students do not get the requisite attention to accomplish their learning in a self-paced manner [2]. Testing is ineffective and students can succeed without the mastery of any of the key concepts, which can create huge gaps in knowledge and make it difficult to learn subsequent lessons. One of the latest advances in educational methodology is the Intelligent

G. Gutjahr · P. Chandrashekhar (✉) · M. Haridas · P. Nedungadi

Center for Research in Analytics and Technologies for Education (CREATE), Amritapuri, India

e-mail: pantinac@am.amrita.edu

M. G. Nair

Department of Mathematics, Amrita VishwaVidyapeetham, Amritapuri, India

© The Editor(s) (if applicable) and The Author(s), under exclusive license

to Springer Nature Singapore Pte Ltd. 2021

A. Joshi et al. (eds.), *Machine Learning for Predictive Analysis*,

Lecture Notes in Networks and Systems 141,

https://doi.org/10.1007/978-981-15-7106-0_27

Tutoring System (ITS), facilitated by the rapidly evolving field of machine learning [3]. An ITS can generate reports which accentuate a student's learning deficiencies.

An ITS tracks student's progress in skill development and provide tailored learning instructions. ITS offers options to focus on the strengths and abilities of each student, and control the pace of learning for the student to master a certain topic. For each student, ITS maintains the usage interaction logs, such as the number of lessons completed, the elapsed time for a student to respond to a question, the number of completed questions, and the number of occasions when a student called for help. ITS evaluates the log data and identifies areas where a student is vulnerable and offers the student the means of instruction in these areas based on their efficiency. This is done using various machine learning models such as Knowledge Tracing Models.

Amrita Learning is an intelligent tutoring system which was developed as an outgrowth of a research to provide personalized education and feedback to the students, in real-time [4–6]. It includes lessons in English, Mathematics, and Science. The system provides various interactive modes, such as videos and animations. Amrita Learning maintains the model of a student, based on the student's prior knowledge, skills, and learning speed. Based on this student models, the learning exercise questions are crafted around these models and presented to the students, concomitant with his or her skill levels.

In a previous paper, students' conceptual knowledge was investigated based on their interactions with the Amrita Learning system, using the Feature-Aware Student knowledge Tracing (FAST) algorithm [7]. The present study builds upon the previous work, focused on the comparison of the performance between the FAST algorithm and Hidden Markov models.

The outline of this paper is as follows: Sect. 2 summarizes previous work in this domain; Sect. 3 describes methodology, Sect. 4 reports the results and Sect. 5 ends with a discussion and conclusion.

2 Literature Survey

The three common knowledge tracing methods are Bayesian knowledge tracing, Hidden Markov Model, and Bayesian knowledge tracing with features. Bayesian Knowledge Tracing (BKT) is a popular approach to model students learning. BKT has a long track record of effective deployments of in ITSes, in connection with mastery learning. Researchers have compared BKT with extension and other models.

Corbett and Anderson compared BKT with Hidden Markov Model-Item Response Theory (HMM-IRT) and learned that HMM-IRT outperforms BKT [8]. Chen Lin found that Intervention BKT, which is a special case of input-output HMM, predicts better than the traditional BKT model because it involves a smaller number of parameters [9]. LR-DBN (Logistic Regression Dynamic Bayes Net) trains and updates subskills together, using logistic regression [10].

FAST is an extension of Knowledge Tracing, which uses logistic regression to model general features in Knowledge Tracing [11]. It was found that FAST outperforms LR-DBN and Item Response Theory (IRT) [11]. Klingler et al. [12] compared FAST with IRT and Latent Factor Knowledge Tracing (LFKT) model and reported that FAST achieved the best performance.

3 Methodology

3.1 Dataset

The current study used the dataset of 2402 students in first-standard (i.e. Grade 1), drawn from 28 CBSE schools in India, who use Amrita Learning ITS regularly for 20 min, twice a week. The Mathematics module of Amrita Learning is constituted of 113 lessons for grade 1. A total of 211,275 recorded responses was acquired over an academic year.

3.2 Bayesian Knowledge Tracing

BKT models student knowledge during the interaction with the ITS. Each interaction with the ITS system is referred to as learning step. Figure shows the representation of the basic BKT model. The latent state of the students knowing the skill in the t th learning step is represented by X_t . The observed variable Y_t denotes whether the answer at t th step is correct or not (incorrect). In the standard BKT algorithm, there are only skill-specific parameters:

- init $P(L_o)$, the probability of a student knowing the skill prior to start of a lesson,
- transit $P(T)$, the probability of a student learning the skill after the completion of the lesson,
- guess $P(G)$, the probability of a student being able to answer questions in the lesson without having the skill and
- slip $P(S)$, the probability of a student knowing the skill but answering the question incorrectly [9] (Fig. 1).

The BKT algorithm updates the probability of the student learning, guessing or slipping by using formulas of conditional probability, throughout the observational sequence.

The conditional probabilities are computed, (whether or not the student had tried the skills correctly) using Eqs. (1) or (2). The mastery of the skill can be found, using the conditional probability represented by Eq. (3). To estimate the probability of student l applying the skill s correctly for the given opportunity one uses Eq. (4).

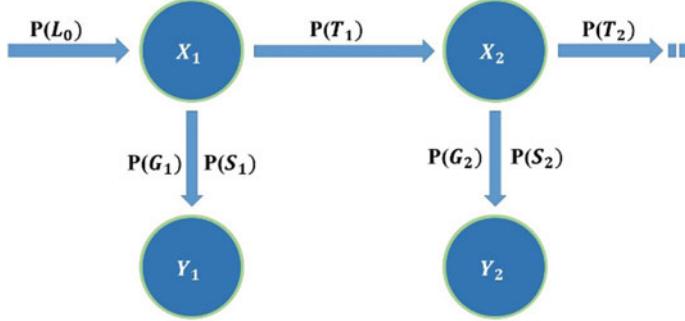


Fig. 1 Standard BKT model (adapted from [11])

$$P(L_t|Y_t = \text{correct}) = \frac{P(L_t)(1 - P(S))}{P(L_t)(1 - P(S)) + (1 - P(L_t))P(G)} \quad (1)$$

$$P(L_t|Y_t = \text{wrong}) = \frac{P(L_t)(P(S))}{P(L_t)(P(S)) + (1 - P(L_t))(1 - P(G))} \quad (2)$$

$$P(L_{t+1}) = P(L_{t+1}|Y_t) + (1 - P(L_{t+1}|Y_t))P(T) \quad (3)$$

$$P(C_{t+1}) = P(L_{t+1})(1 - P(S)) + (1 - P(L_{t+1}))P(G) \quad (4)$$

This simple model can be extended to include more variables that the unknown variables or the observational variables depend on (see Figs. 2 and 3). Such additional variables are called features.

Z_t represents the set of emission features at the t th learning opportunity. Emission features are the “features” which affect the emission probability (guessing or slipping). Transition features are the features that may affect the transition probabilities. Nevertheless, models have been proposed for each of these specific cases.

Z_t represents the set of features which is involved in performing the t th step.

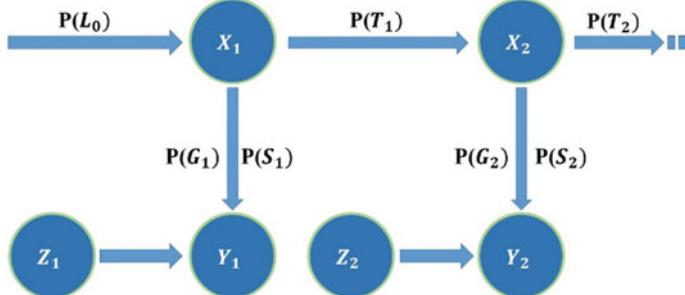


Fig. 2 HMM extension to model emission features (adapted from [11])

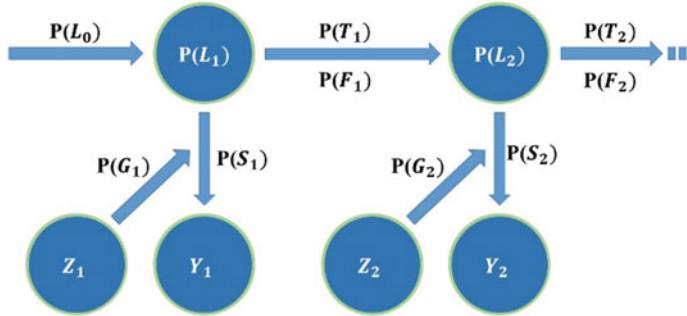


Fig. 3 FAST model for multiple subskills (adapted from [11])

3.3 Hidden Markov Models

Hidden Markov Models (HMM) are probabilistic models in which the system is modelled with hidden or unobservable Markov process states. The term “states” refers to the hidden states and the term “observations” refers to the observed states.

Standard BKT model can be extended to allow general features as shown in Fig. 2. The model assumes as checkpoints, when each time a resource is used or a question from the lesson is asked. Between two consecutive questions, the student may access a large number of resources which may increase the student knowledge regarding the lesson. The HMM model fits the learning probability rates for each of the available resources.

The model is fit using the Expectation–Maximization (EM) algorithm. In the k th step of the algorithm, the E step takes as an estimate x_t^k for the latent variable’s expectations

$$x_t^k = E(X_t | Y_t, P(Lo) = \pi_L^k, P(T) = \pi_T^k, P(G) = \pi_G^k, P(S) = \pi_S^k)$$

where the estimates $\pi_L^k, \pi_T^k, \pi_G^k, \pi_S^k$ are found in the M step by maximizing the likelihood for fixed values x_t^{k-1} for the latent variable X_t .

3.4 Feature-Aware Student Knowledge Tracing (FAST)

FAST is an extension of knowledge tracing, which allows features in the transitions as well as the emissions. Unlike most other knowledge tracings, FAST uses logistic regression parameters instead of conditional probability tables.

Using FAST, arbitrary features as given in Fig. 3, can be combined into a knowledge tracing model, with minimal computational complexity, using linear regression terms instead of calculating conditional probability tables which grow exponentially with the number of features.

Thus, FAST is a general model; it can model multiple subskills as well; such that they do not depend equally on a question that requires them, the opposite of which is an assumption for the HMM model. The learning, forgetting, guessing, or slipping probabilities are fit using a modification of the EM algorithm called the Expectation–Maximization with Features (EM with features) algorithm. The EM algorithm consists of E step and M step, where E step deals with mastering the skills at each step and the probability is deduced. M step involves a drastic change, where, instead of the probability tables, the parameters are modelled using logistic regression, which is continuous and hence, observable during training and testing.

We assume a linear relationship between the features Z_t , and the log-odds of the parameters. This linear relationship can be written in the following mathematical form:

$$\log_e \frac{p}{1 - p} = \beta_0 + \beta_1 Z_t$$

We learn β_0 and vector β_1 from the given data, by training through a weighted regularized logistic regression.

4 Comparison Methodology

Accuracy is a statistical measure that is defined as the quotient of correct predictions made by a classifier divided by the number of predictions made by the classifier.

Root Mean Square Error (RMSE) is the measure of the difference between the values predicted by the model and the actual values [15]. It is the standard deviation of the squares of the residuals. The smaller the RMSE, the more precise the prediction.

The ROC curve gives the relationship between true positive rates and false positive rates as a function of the threshold used for classification. The AUC is considered to be the area under the ROC curve [16], indicates how well the model can distinguish between classes.

Cross-validation is splitting one's training set into subsets and then training/fitting the machine learning to model a subset, which is used to predict the remaining subsets. The model performance is assessed using a metric depending on what is being predicted. In the present study, accuracy RMSE and AUC were estimated for all methods using cross-validation.

Table 1 Model comparison

| | Accuracy | RMSE | AUC |
|------|----------|------|------|
| FAST | 0.69 | 0.47 | 0.66 |
| HMM | 0.72 | 0.43 | 0.70 |

5 Results

Both algorithms slightly improve over Bayesian knowledge tracing without feature, where the AUC = 0.64. As seen from Table 1, the cross-validation accuracy, RMSE, AUC are all marginally better for HMM than FAST. The differences, however, are relatively small.

6 Discussion and Conclusion

In many Indian schools, a teacher has to handle a large group of students with different learning capabilities. This makes it a challenging task for teachers to gauge where students lack in their study skills. Teachers should have a clear plan understanding of the concepts that need to be taught in greater detail, for each student. Generating reports that show which students need additional training in certain concepts helps teachers set students up for the CBSE exams. The goal is to test whether a student has mastered the NCERT concept and to model and generate reports of marginal probabilities. Such reports can be based on knowledge of tracing models.

For the current study, we compared the performance of three knowledge tracing models: the standard BKT, the FAST algorithm, and the Hidden Markov model. The accuracy, RMSE, and AUC values show that HMM slightly outperforms FAST and BKT. One reason may be that FAST uses an approximation of the full posterior distribution calculated by HMM. The prime advantage of the FAST algorithm is its minimal drain on computational time.

In future, we plan to compare HMM with other algorithms such as deep knowledge tracing, Intervention BKT. In the future, we hope to expand the comparison of the algorithms on a larger variety of student data, from different grades. On incorporation of this algorithm into the ITS, the feedback from teachers is inevitable, for the appraisal of the utility of the proffered methods.

One limitation of the study is that it only used the first-grade student data. Another limitation is that although the number of students used for the study is large but follow up for the student being for a shorter period of time.

References

1. V. Prain, P. Cox, C. Deed, J. Dorman, D. Edwards, C. Farrelly, B. Waldrip, Personalised learning: lessons to be learnt. *Br. Edu. Res. J.* **39**(4), 654–676 (2013)
2. Pupil-teacher ratio by level of education, UNESCO. <http://data UIS.unesco.org/index.aspx?queryid=180>
3. K. VanLehn, The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**(4), 197–221 (2011)
4. P. Nedungadi, M.S. Remya, Incorporating forgetting in the personalized, clustered, bayesian knowledge tracing (pc-bkt) model, in *2015 International Conference on cognitive computing and information processing (CCIP)* (IEEE, 2015), pp. 1–5
5. G. Gutjahr, K. Menon, P. Nedungadi, Using an intelligent tutoring system to predict mathematics and english assessments, in *2017 5th IEEE International Conference on MOOCs, Innovation and Technology in Education (MITE)* (IEEE, 2017), pp. 135–140
6. R. Raman, P. Nedungadi, Adaptive learning methodologies to support reforms in continuous formative evaluation, in *2010 International Conference on Educational and Information Technology* Vol. 2 (IEEE, 2010), pp. V2–429
7. M. Haridas, N. Vasudevan, S. Gayathry, G. Gutjahr, R. Raman, P. Nedungadi, Feature-Aware knowledge tracing for generation of concept-knowledge reports in an intelligent tutoring system, in *2019 IEEE Tenth International Conference on Technology for Education (T4E)* (IEEE, 2019), pp. 142–145
8. S. Schultz, I. Arroyo, Tracing knowledge and engagement in parallel in an intelligent tutoring system, in *Educational Data Mining* (2014)
9. C. Lin, S. Shen, M. Chi, Incorporating student response time and tutor instructional interventions into student modeling, in *Proceedings of the 2016 Conference on user Modeling Adaptation and Personalization* (2016), pp. 157–161
10. Y. Xu, J. Mostow, Comparison of methods to trace multiple subskills: is LR-DBN best? *Int. Educ. Data Mining Soc.* (2012)
11. J. González-Brenes, Y. Huang, P. Brusilovsky, General features in knowledge tracing: applications to multiple subskills, temporal item response theory, and expert knowledge, in *The 7th International Conference on Educational Data Mining* (pp. 84–91). University of Pittsburgh (2014)
12. S. Klingler, T. Käser, B. Solenthaler, M. Gross, On the performance characteristics of latent-factor and knowledge tracing models. *Int. Educ. Data Mining Soc.* (2015)
13. M.V. Yudelson, K.R. Koedinger, G.J. Gordon, Individualized bayesian knowledge tracing models, in *International conference on artificial intelligence in education* (Springer, Berlin, Heidelberg, 2013), pp. 171–180
14. M.J. Beal, Z. Ghahramani, C.E. Rasmussen, The infinite hidden markov model, in *Advances in neural information processing systems* (2002), pp. 577–584
15. T. Meyer, Root mean square error compared to, and contrasted with, standard deviation. *Surveying Land Inf. Sci.* **72**(3), 107–108 (2012)
16. C. Cortes, M. Mohri, Confidence intervals for the area under the ROC curve, in *Advances in neural information processing systems* (2005), pp. 305–312

ABCADF: Deploy Artificially Bee Colony Algorithm for Model Transformation Cohesive with Fitness Function of Adaptive Dragonfly Algorithm



Pramod P. Jadhav and Shashank D. Joshi

Abstract Model transformation (MT) is the key factor in the software engineering field. To get the project successful most of the time the developer demand the transformation of the one model to another, to fulfill the requirement of the developer and make the project reliable model transformation is important. Agenda of this paper is to convert the class diagram (CLD) to the relational schema (RS). MT frame the protocol for transformation. In this work model transformation is achieved through Artificial Bee colony (ABC) and adaptive dragonfly algorithm for model transformation along with fitness function of test-driven development (TDD). Finally, for the result and analysis, three CLD models are transformed to RS and compare to existence optimization algorithms like DF, ADF, PSO, etc.

Keywords Model transformation · Bee algorithm · Fitness function · UML class diagram · Relational schema · Dragonfly algorithm

1 Introduction

Different properties of the model transformation are used to satisfy the user requirement, along with some static and dynamic regression techniques, synchronization of function, and specification of the simple and graph model transformation [1]. Higher-Order Transformation is considered for the next step and advancement of the model transformation [2]. In these overall scenario object-oriented modeling (OOM) notation is consider for the standardization of the unified modeling language (UML) [3]. Unified Modeling Language (UML) is one of the useful languages in the data modeling, these models are used in the various software project as a flow of project, and same UML diagram is used for input and output of model transformation (MT).

P. P. Jadhav (✉)

G H Raisoni Institute of Engineering and Technology, Wagholi, Pune, Maharashtra, India

e-mail: pramod.jadhav@raisoni.net

S. D. Joshi

Bharati Vidyapeeth Deemed to be University, Pune, Maharashtra, India

This paper focus on the model and its different aspect of model transformation, it may require some concept of model driving engineering (MDE) and object-oriented modeling concept. Work also contains an adaptive dragonfly algorithm (ADF), and TDD based fitness function. These concepts are associated with Artificial Bee colony algorithm unified with adaptive dragonfly algorithm (ADF), [4–6] fitness function makes the result very effective and used for further research. Here behavior of the bee colony has been identified and accordingly, algorithm is designed. The weight of the algorithm and find the suitable block can be further used as input of the next phase. And finally, after the whole process class diagram (CLD) can be converted into Relational schema (RS) diagram which is the main aim of this research work [7, 8].

The major contribution of the paper is:

Proposed Artificial Bee Colony algorithm for model transformation cohesive with fitness function of ADF: The proposed work is considered with the Adaptive Dragonfly (ADF) algorithm enabled with the Artificial Bee colony algorithm using fitness function and block codes, for model transformation of CLD to RS model [4, 5, 7].

The paper is organized as:

Section 1 introduces the paper core idea, Sect. 2 consider the existing works and made a literature review, Sect. 3 explains the proposed work of this research, Sect. 4 discuss the results of the proposed system and compare the result with existing work. And finally, put up the conclusion of research work.

2 Motivation of the Work

This section explains the literature review of the existing Model Transformation concept and elaborates the core idea about the model transformation.

2.1 Literature Survey

Advantage of this method in Software Company to produce more accurate and quality output towards the model transformation and related system [3, 9]. design a mapping model along with the Fuzzy technique to form a fuzzy relational database and produce a fuzzy UML data model. It includes different fuzzy formulae to make a steady fuzzy module and further, it used in different model-based structure for the transformation. The major limitation of this approach was to make a complicated structure and relate with some faulty database, which may affect the output model [10] developed a model transformation that converts source CLD model to a modified CLD target model. This method should improve the maintenance phase in the targeted system and also concentrate on behavior of the schema required for the model transformation [11, 12]. Same approach can be modified with the different algorithms and try to find out

the best approach among them. Along with the dragonfly algorithm other supporting algorithms like ant colony optimization, fractional weight age base approach, and whale optimization-based approach [4, 6], can be rectified for better result [5, 7]. Other approaches are also trying to prove the better result as compare to the existing one.

3 Proposed Bee Colony Algorithm for Effective Model Transformation

Bee colony introduces for the behavior of the bee in daily routine. Here bees routine is an observer and follows the algorithmic steps. In the artificial bee colony concept, there are three types of bees are present. (a) Employed bee (b) Onlooker bee and (c) Scout bee. Employed bee tries to search the food source and then come to the hive for planning to get that food. Meanwhile, they share the information of the food source to onlooker bee, while onlooker bee is doing analysis for a good food source and select the best food from available data share by an employed bee. High-quality food is selected or targeted from the onlooker bee [13–15]. Once the food source is targeted then employee bee are select some of the bees to get that selected food, called scout. The main swarm in this concept is employed bee and onlooker bee. The number of optimized solutions in this algorithm is equal to the number of onlookers or employed bee available in the hive. Generally, it is based on a random solution which depends upon the total population in the hive. Let $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,N}\}$ represents the random solution and D represent the total dimension. Every employed bee X_i produces a new nominee solution V_i in the community of its existing location as follows:

$$j = x_{i,j} + \Phi_{i,j} * (x_{i,j} - x_{k,j}) \quad (1)$$

where X_k is arbitrarily selected candidate solution ($i \neq k$), j is an arbitrary dimension index selected from the set $\{1, 2, \dots, N\}$, and $\phi_{i,j}$ is an arbitrary number within $[-1, 1]$. If the fitness value of V_i is superior than parental X_i then update X_i with V_i employed bees are search the food and after completing the search process they share search information with onlooker bee [1, 8]. The sharing of this food information through waggle dance. Onlooker bees select the food source from deep analysis of information and select the best one by considering the amount of the food. This is a probabilistic selection from available information, below are the function through which random selection is done [14].

$$P_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (2)$$

where f_i is the fitness function of i th solution, the maximum solution of containing higher probability for food selection [15]. If the position cannot improve within a specific period of time then (threshold time) then it will confirm that an ample amount of food is available. Then scout bee discovers new food sources to be replaced with X_i

$$x_{i,j} = lg_i + \Phi_{i,k} * (ug_j - lg_j) \quad (3)$$

where $\Phi_{i,k}$ is the random solution with the value $[0, 1]$. And the ug and lg are the upper and lower boundary of the j th dimension. Employed bee share the information through waggle dance; waggle dance indicates the correct food information to the onlooker bee. Onlooker bee are decided and select better food source [2, 13].

Waggle dance indicates the angle between food and sun indicated as α , one bee coming from the food to angle zero, and the same another bee on the other side, between two bees there is a spiral shape having specific dimension. This activity is called the waggle dance.

3.1 First Module of Proposed ABCADF Algorithm

Our main goal is to transform the class diagram (CL) to the Relational Schema (RS) model. In the proposed methodology, a class diagram is an input through which different block sets are considered which is useful to select the best block, and then it is used in the adaptive dragonfly algorithm (ADF). To find the best and correct solution ADF is used along with fitness function. Fitness function is supportive of the algorithm and overall proposed system. This is considered as the first module of the system. This module provide the best solution to the second module of this system. Proposed system has divided into two phases, first phase is training set data, and the second phase is testing set data. In the second phase, the data coming from the first phase is considered as an input of second phase and then second phase is executed, in the second phase, Artificial Bee Colony (ABC) algorithm is used along with the ADF which gives a better solution. Generally, ABC is an optimization algorithm and used to find the optimal solution in the available population. Here all types of bees are considered as a total population, in this system also take care of the position update and Bee velocity for the better solution. Phase II is the testing phase which is the final stage of this work. Algorithm of the ABC and ADF works together and transforms the class (CL) diagram model to the Relational Schema (RS) model (Fig. 1).

- (a) **Block sets scenario.** Block set are represented as B . and indicated as $B = \{b_1, b_2, b_3, \dots, b_s, \dots, b_t\}$, in this b_s is the selected block among the total block represented as b_t . Block set limit can be indicated as $1 < b.s < b_t$, selected block set will be used for further process.

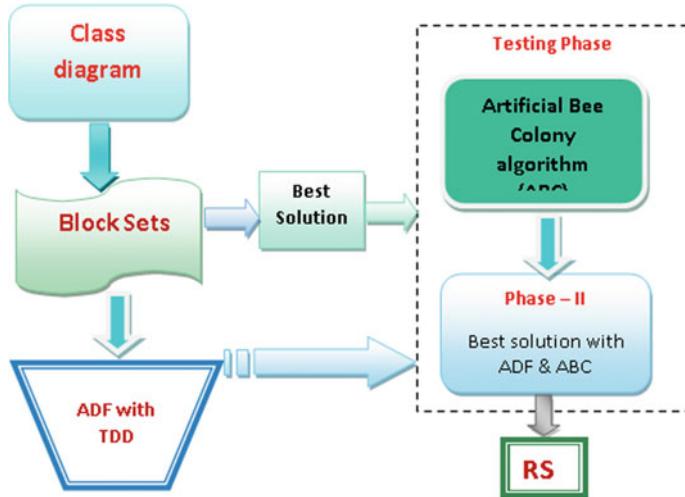


Fig. 1 The architecture of proposed ABCADF based model transformation

$$B = \{b_1, b_2, b_3, \dots, b_s, \dots, b_t\} \quad (4)$$

- (b) **Fitness function:** Here fitness function is made by considering the internal coherence, Adequacy, External coherence, along with Association of the transformations.

$$\text{Fitness} = Fc_1 + Fc_2 \quad (5)$$

where the fitness Fc_1 and Fc_2 are indicated by using the following equation,

$$Fc_1 = \sum_{l=1}^8 a_l * (In_l + Ex_l) \quad (6)$$

where a_l is adequacy factor, In_l is an internal coherence, and Ex_l consider as external coherence factor for l th construct. The TDD fitness is as follows: [7]

$$Fc_2 = \frac{\text{Count of satisfied test cases}}{\text{Total Count of test cases}} \quad (7)$$

- (c) **Position update of ADF:** in the case of the adaptive dragonfly algorithm, dragonfly moves from one position to another position with their specific velocity. Velocity of dragonflies can be intended by considering the element such as alignment, Cohesion, separation, attraction towards the food, and disruption from the enemy.

$$\Delta P(t+1) = (fF_q + gG_q + hH_q + iI_q + jJ_q) + \lambda_2 \cdot \Delta V(t) \quad (8)$$

where $\Delta P(t+1)$ is a velocity update of the ADF algorithm which will consider for further calculation. And f, g, h, I and j refer to the weight for updating the value of separation, alignment, cohesion, attraction, and distraction, respectively. Also, F_q, G_q, H_q, I_q and J_q represent the cohesion, distraction, attraction, alignment, and separation, of the q th dragonfly which apprise the velocity. The term λ_2 specifies that weight apprising along with amount of iteration. ADF algorithm, the more cultured functioning of ADF contemplates the value of $f = g = h = i = j = \lambda_1$. For every iteration,

$$\Delta P(t+1) = \lambda_1(F_q + G_q + H_q + I_q + J_q) + \lambda_2 \cdot \Delta V(t) \quad (9)$$

where, λ_1 and λ_2 denotes to apprising weight element, which is subject to an overall amount of iteration. The equation of adaptive weights are altered as follows:

(d) ***Proposed ABCADF algorithm for position update:***

$$\text{Fitness} = P_i = \frac{Fc_1 + Fc_2}{\sum_{j=1}^{TS} f_i} \quad (10)$$

Fitness function will generate the fitness value of ADF function and the value from the artificial bee colony algorithm is assigned to the fitness function. Fitness = $Fc_1 + Fc_2$ is assigned to the P_i and then $\sum_{j=1}^{TS} f_i$ is considered for the total fitness. Equation (10) will consider the total fitness evaluation in the ABCADF algorithm.

Whereas the above equation can be modified by considering the ADF position update value.

$$x_{i,j} = +\Phi_{i,k} * (ug_j - lg_j) \quad (11)$$

Above equation indicate the value of $x_{i,j}$ which containing the ADF position update term from adaptive dragonfly $\lambda_1(F_q + G_q + H_q + I_q + J_q) + \lambda_2 \cdot \Delta V(t)$ —(9) as a $\Delta P(t+1)$. Value of Eq. (9) is replaced with the $\Delta P(t+1)$ and find the whole value of Eq. (11).

$$x_{i,j} = +\Phi_{i,k} * (u_j - l_j) \quad (12)$$

Equation (11) is reformed to the value of the $x_{i,j}$ and will get the Eq. (12) which is the final equation of this algorithm which get better result towards the model transformation. u_j and l_j Is the upper and lower bound term which is also helpful for the correct value of the bee position. Here random value is considered for the final equation.

Algorithm 1 Algorithmic step of ABCADF for the transformation from CLD to RS

| Sl. no | Proposed ABCADF |
|--------|--|
| 1 | Input: Available Population P_i |
| 2 | Output: Best solution P_i |
| 3 | Start |
| 4 | Initial Population |
| 6 | { |
| 7 | For ($s = 1; s < i_{\max}$) |
| 8 | For all the solution |
| 9 | Compute the fitness using equation (10) |
| 10 | If (solution s has neighbor) |
| 11 | Change the velocity using equation (09) |
| 12 | Change the position using equation (12) |
| 13 | Else |
| 14 | Change the position using equation (3) |
| 15 | End if |
| 16 | End for |
| 17 | } |
| 18 | End |

3.2 Second Module of ABCADF Method

Second module is the testing module which will work after the first module is executed and then output of the first will consider the input of the second module and then selected block is used in the testing purpose. Solution Blocks is selected based on the proposed algorithm from solution encoding. Proposed Artificial bee colony (ABC) algorithm is used in both the module indicated in the proposed architecture. After successfully testing both the module result will generate in terms of relational schema (RS) model which ultimate aim of this research work.

4 Results and Discussions

This segment indicates the result of the proposed work which also provide a comparison with existing algorithm. The result is generated using the ABCADF algorithm. Proposed algorithm encompasses the CLD model as input diagram and produces the relational schema (RS) as an output model. Here in this work AD, PSO, ADF, algorithm are considered for the comparison purpose. Finally, performance measure parameter such as Automatic correctness (AC) and Fitness measure are comprised for accurate evaluation

4.1 Database Description

In this research work, three input model encompasses for the comparison. All three input models are having a separate number of instances like classes, association, and aggregation. The following table indicate the number of classes, aggregation, and association. In the above section, it is already discuss that DA, PSO, ADF, and ACADF are used for comparison and further calculation. Block will be selected among the total block set of example base. After Applying, the proposed algorithm selected block is considered for further processing and generates the output based on the best solution selected through algorithm.

Database of CLD 1 model: first CLD having total 8 instances, having 3 classes, 3 aggregation, and 2 association.

Database of CLD 2 model: second CLD having 9 instances, which include 2 classes, 4 aggregation, and 3 association.

Database of CLD 3 model: third CLD having 11 instances, which include 4 classes, 3 aggregations, and 4 association.

4.1.1 Evaluation Metrics

Proposed ABCADF methods is compared with existing algorithm and quantity the performance by considering AC and fitness function. Section 3 discusses the fitness function based on the evaluated equation, and find the best fitness function by encompassing the artificial Bee Colony (ABC) and ADF algorithm.

Automatic Correctness: After completing the process result is generated. Comparison and performance can be measure based on the Automatic Correctness (AC). AC is fluctuating from 0 to 1 and the resultant graph will display within a specified range.

4.2 Algorithm Used for Comparison

The existing algorithms used for evaluation are, practical swarm optimization PSO [3], Dragonfly algorithm [7], Adaptive Dragonfly algorithm (ADF), and the Ant Colony unified with Adaptive dragonfly algorithm (ACADF). All these algorithm are shown in a table and graph along with performance measure AC and fitness function.

4.2.1 Analysis for CLD 1 with Comparison

Let us see the comparative analysis of CLD 1 with the proposed ABCADF algorithm with existing methods like DA, PSO, ADF, ACADF. This analysis is based on the performance criterion AC and fitness function. Graph show two-part Fig. 2a is for

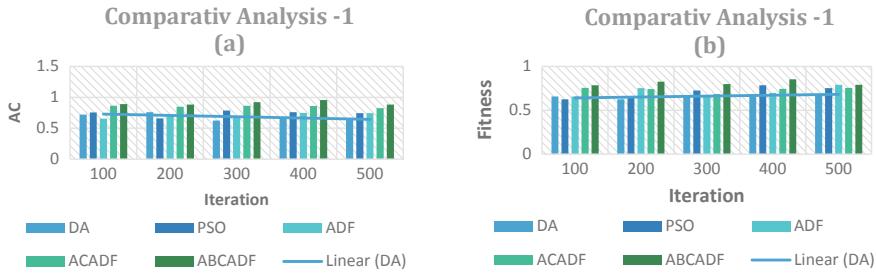


Fig. 2 **a** Comparative analysis, **b** comparative analysis

the AC with comparison and Fig. 2b is for the fitness function. For analysis here, 100 iterations to 500 iteration are considered for the analysis. In Fig. 2a, comparison can takes place between DA, PSO, ADF, ACADF, and ABCADF and its 100 iteration value is 0.7212, 0.7568, 0.6578, 0.8647, and 0.8924, respectively. comparison can takes place between DA, PSO, ADF, ACADF, and ABCADF and its 500 iteration value is 0.6645, 0.7452, 0.7451, 0.8245, and 0.8834. In Fig. 2b fitness function of the 100 iteration are 0.6584, 0.6254, 0.6578, 0.7548, 0.7854 and 500 iteration value are 0.6821, 0.7524, 0.7894, 0.7562, 0.7895, respectively, in above algorithm sequence.

4.2.2 Analysis for CLD 2 with Comparison

In case of comparative analysis 2 of the CLD 2 compare proposed ABCADF algorithm with existing methods like DA, PSO, ADF, and ACADF. This analysis is again based on the performance criterion AC and fitness function. For analysis here, 100 iterations to 500 iterations are considered for the analysis. In Fig. 2a i.e. AC measures, comparison can takes place between DA, PSO, ADF, ACADF, and ABCADF and its 100 iteration value is 0.7548, 0.7412, 0.7125, 0.6987 and 0.7896, respectively. Comparison can takes place between DA, PSO, ADF, ACADF, and ABCADF and its 500 iteration value is 0.7562, 0.7541, 0.6985, 0.7512, and 0.7985, respectively. In Fig. 2b fitness function of the 100 iteration are 0.7856, 0.7415, 0.7516, 0.7489, 0.8213, and 500 iteration value 0.7521, 0.8524, 0.7584, 0.7896, 0.8954, respectively in above algorithm sequence.

4.2.3 Analysis for CLD 3 with Comparison

Comparative analysis 3 of the CLD 3 compare proposed ABCADF algorithm with existing methods like DA, PSO, ADF, and ACADF. This analysis is also based on the performance criterion AC and fitness function. First let us discuss the analysis for 100 iteration for AC measures of DA, PSO, ADF, ACADF, and ABCADF and its value 0.7562, 0.7568, 0.6524, 0.7642, and 0.7952, respectively. And for 500 iteration value, respectively 0.6985, 0.6983, 0.7524, 0.7985, and 0.8032, respectively. In Fig. 2b

Table 1 Comparative analysis

| Input | Performance metrics | Comparative methods | | | | |
|-------|---------------------|---------------------|--------|--------|--------|-----------------|
| | | DA | PSO | ADF | ACADF | Proposed ABCADF |
| CLD 1 | AC | 0.6645 | 0.7452 | 0.7451 | 0.8245 | 0.8834 |
| | Fitness | 0.6821 | 0.7524 | 0.7894 | 0.7562 | 0.7895 |
| CLD 2 | AC | 0.7562 | 0.7541 | 0.6985 | 0.7512 | 0.7985 |
| | Fitness | 0.7521 | 0.8524 | 0.7584 | 0.7896 | 0.8954 |
| CLD 3 | AC | 0.6985 | 0.6983 | 0.7524 | 0.7985 | 0.8032 |
| | Fitness | 0.7562 | 0.7546 | 0.6542 | 0.6254 | 0.7892 |

fitness function of the 100 iteration are 0.6589, 0.7532, 0.7546, 0.7569, 0.7989, and 500 iteration value 0.7562, 0.7546, 0.6542, 0.6254, 0.7892, respectively, in above algorithm sequence.

4.3 Comparative Discussion

Here comparison can take place along with existing algorithms as discussed in the above section like DA, PSO, ADF, ACADF, and ABCADF, these comparisons can possible with the help of the performance measure like AC and Fitness function. Above section In graph 100–500 iteration are considered for the comparison. And in the explanation of these graph iteration, 100 and 500 are explained, for the overall comparison iteration 500 is consider and depicted in Table 1.

Above table indicate that all existing methods are considered for the 500 iterations in the performance measure of AC and fitness factor of all the algorithm. Proposed algorithm are showing better facts and figure as compared to the existing one. In this research artificial Bee Colony adaptive with dragonfly algorithm is performing better by considering the AC and fitness factor.

5 Conclusion

In this research work, the proposed artificial Bee Colony adaptive with the dragonfly algorithm (ABCADF) is used for the model transformation. Basically, the ABC is the optimization algorithm that will helpful for block set selection, as well as better performance for the CLD to RS model transformation. In the first phase of this architecture, ABC and ADF will consider for the block selection and model transformation. While in the second phase testing is done for the proposed algorithm and get the output as RS model. After the comparing existing algorithm, section comparative discussion indicates that the proposed algorithm shows show the better performance, with the values of the CLD 1, CLD2, CLD 3, followed by its AC and

Fitness function as 0.8834 0.7895 0.7985 0.8954 0.8032 0.7892, respectively. As a limitation, this system will decrease the performance when the number of iteration is increased. Overall proposed ABCADF algorithm performs better than the existing one and it is more suitable for model transformation.

References

1. M. Kessentini, H. Sahraoui, M. Boukadoum, B.O. Omar, Search-based model transformation by example, *Softw. Syst. Model.* **11**(2), 209–226 (2012)
2. L. Samimi-Dehkordi, B. Zamani, S. Kolahdouz-Rahimi, EVL+ Strace: a novel bidirectional model transformation approach, *Inf. Softw. Technol.* **100**, 47–72, (2018)
3. J.S. Cuadrado, E. Guerra, J. de Lara, Static analysis of model transformations. *IEEE Trans. Software Eng.* **43**(9), 868–897 (2017)
4. P.P. Jadhav, S.D. Joshi, WOADF: whale optimization integrated adaptive dragonfly algorithm enabled with the TDD properties for model transformation. *Int. J. Comput. Intell. Appl.* **18**(4), 1950026 (21 p) # c World Scientific Publishing Europe Ltd. <https://doi.org/10.1142/s1469026819500263>
5. P.P. Jadhav, S.D. Joshi, ACADF: ant colony unified with adaptive dragonfly algorithm enabled with fitness function for model transformation, in *International Conference on Communication and Cyber Physical Engineering (ICCCE-2019)* Feb 1–2, 2019, Pune, India. 2nd edn
6. P.P. Jadhav, S. Joshi, Fractional weightage based objective function to hybrid optimization algorithm for model transformation. *Evol. Intell.* ISSN: 1864-5909 <https://doi.org/10.1007/s12065-018-0179-8> Springer © Springer GmbH Germany, part of Springer Nature 2018
7. P.P. Jadhav, S.D. Joshi, ADF: adaptive dragonfly optimization algorithm enabled with the TDD properties for model transformation. *Int. J. Database Theory Appl.* **11**(4), 41–58 (2018). <http://dx.doi.org/10.14257/ijdta.2018.11.4.04>
8. B. Basturk, D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, in *IEEE Swarm Intelligence Symposium 2006*, Indianapolis, Indiana, USA (2006)
9. Z.M. Ma, F. Zhang, L. Yan, Fuzzy information modeling in UML class diagram and relational database models. *Appl. Soft Comput.* **11**(6), 4236–4245 (2011)
10. P.P. Jadhav, S.D. Joshi, A.M. Bagade, A conceptual study of test case design by Investigating the various principles and aspects. *Int. J. IT Eng. Impact Factor IJITE* **04**(10), 33–43 (2016) ISSN: 2321–1776
11. M. Fleck, J. Troya, M. Kessentini, M. Wimmer, B. Alkhazi, Model transformation modularization as a many-objective optimization problem. *IEEE Trans. Softw. Eng.* **43**(11), 1009–1032 (2017)
12. A.M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. Maga.* 1556-603X <https://doi.org/10.1109/mci.2006.329691>. IEEE Computational Intelligence society
13. Y. Xu, P. Fan, L. Yuan, A simple and efficient artificial bee colony algorithm. in *Hindawi Publishing Corporation Mathematical Problems in Engineering*, vol. 2013, Article ID 526315, 9 p. <http://dx.doi.org/10.1155/2013/526315>
14. A. Baykaslu, L. Özbakir, P. Tapkan, Artificial bee colony algorithm and its application to generalized assignment problem swarm intelligence: focus on ant and particle swarm optimization, ed. by T.S.C. Felix, M.K. Tiwari (ITECH Education and Publishing, Vienna, Austria, 2007) p. 532. ISBN 978-3-902613-09-7
15. H.A. Abbass, MBO: marriage in honey bees optimization a haplodetrosis polygynous swarming approach, in *CEC2001 Proceedings of the Congress on Evolutionary Computation*, Seoul, Korea (2001), pp. 207–214

Performance Comparison of Markov Chain and LSTM Models for Spectrum Prediction in GSM Bands



Sandeep Bidwai , Shilpa Mayannavar , and Uday V. Wali

Abstract A critical component of Cognitive Radio (CR) is an ability to predict availability of unused RF slots at a given time and location. Secondary users can use such unused slots without any licensing requirements as long as they retreat when the primary user wishes to occupy the band. With increasing demand on data bandwidth, frequencies in basic GSM bands often go under-utilized, and hence are excellent candidates for CR. RF channel utilization is largely random and hence difficult to predict. Use of machine learning techniques could alleviate some of the uncertainty in prediction. In this paper, we compare the performance of a Markov chain based model with a machine learning LSTM model for use in CR, in terms of accuracy. We have analyzed clusters of five channel-pairs for a given time period, considering the frequency planning guidelines of GSM.

Keywords Cognitive radio · Spectral estimation · Markov chains · LSTM

1 Introduction

1.1 Cognitive Radio (CR)

Spectrum is a limited resource constrained by the increasing demand for wireless communication. Therefore, innovation in improving the spectrum utilization requires

S. Bidwai ()

KLE Dr M S Sheshgiri College of Engineering and Technology, Belagavi, Karnataka 590008, India

e-mail: profssb@gmail.com

S. Mayannavar · U. V. Wali

S G Balekundri Institute of Technology, Belagavi, Karnataka 590010, India
e-mail: mayannavar.shilpa@gmail.com

U. V. Wali

e-mail: udaywali@gmail.com

immediate attention. Cognitive Radio (CR) was introduced to alleviate underutilization of certain frequency bands. CR provides legal access to unlicensed users (secondary users) if the licensed users (primary users) do not use the allocated bands, as long as the secondary users vacate the bands when primary users start transmitting. Therefore, CR requires implementation of spectrum sensing and innovative spectrum usage prediction methods. Further, it is interesting to note that analog TV bands became unused because of the popularity of digital TV and now basic GSM bands are being underutilized. Loads handled by 4G and 5G can not in any way be handled by GSM900 and E-GSM900 because of bandwidth limitations, even when protocols like LTE are implemented. Therefore use of CR can improve the spectrum utilization in these bands [1]. Prediction of spectrum utilization is complex and hence needs support of advanced statistical techniques like Markov chains and Monte Carlo methods for sampling. On the other hand, machine learning techniques may be able to improve prediction accuracy by their ability to extract non-linear information from the training data.

The states of channels are observed over the L consecutive time period in which the time interval is different. We have recorded the GSM traffic in various regions like Belagavi city, Pune city, Sangli city and rural area like ashta. The GSM band analysis is carried out based on occupancy and un-occupancy [2]. It was found that Uplink of the GSM band i.e. 890–915 MHz is mostly remaining vacant whereas downlink (935–960 MHz) is always occupied. In many research works till today, it was found that the assumptions about the Markov chain existence for PUs for channel utilizations were not validated. In this research work, we validated the Markov chain existence in the PU channel utilization using real time measurements collected in GSM band (890–960 MHz). Further the detection process of Cognitive Radio is inclined to errors, we probabilistically model the errors and then formulate the spectrum sensing paradigm as a hidden Markov model that predicts the true state of the channel. The occupancy of our proposed method in predicting true states of the channel is substantiated using extensive simulations.

The occupancy of frequencies at a time of instance can be considered as a state that can be either free (unoccupied by PU) or busy (Occupied by PU). The states of the sub-bands are observed over L consecutive time periods where each time period is of a min 3 s in our case. In this paper, we focus on GSM band (890–960 MHz). Existing research work reported in [3–7] assumes existence of Markov Chain in RF transmissions. In [8], LSTM model is designed for channel prediction in GSM band in [9] whereas the comparative study of Deep learning algorithms like LSTM, Auto Encoder and MLP are present in [10] for cognitive radio system. In [11], three step ahead spectrum prediction frame work was designed based on neural network which is optimized by genetic algorithm to avoid the local optimization problem. The new method of wideband spectrum sensing based on channel clustering and prediction in mentioned in [12]. In this research paper, the detection channel is selected and detected for each cluster and then the state of estimated channel in the cluster is predicted with the help of Hidden Markov Model to get the states of all wide band channels. In [13] a Convolutional Neural Network (CNN) based Deep learning algorithm for spectrum sensing is presented and proved that that model is perform

well compared to Estimator-Correlator detector and Hidden Markov Model (HMM) in terms of correct detection accuracy. In cognitive Radio, it has been suggested in [14, 15] that the temporal relation between primary users states can be represented by the Markov process or Semi-Markov process and in [16], Hidden Markov model has been applied in spectrum sensing to harness such chronological relation to improve the spectrum sensing performance.

1.2 Global System for Mobile (GSM) Communication

The cellular concept is a system with many low power transmitters, each providing coverage to only small portion of the service area. Each base station is allocated a portion of the number of channels available to the entire system and nearby base station are assigned different group of channels so that the interference between base stations is minimised channels assignment in case of GSM900 and E-GSM900 shown below (see Fig. 1).

Channel utilization in GSM bands is dependent on time, frequency of use and selected base station. We have labelled the GSM channels from 890 to 915 MHz as channel number 1–125 (uplink channels) and from 935 to 960 MHz as Some illustrative examples from our data set are indicated above (see Fig. 2). Respective

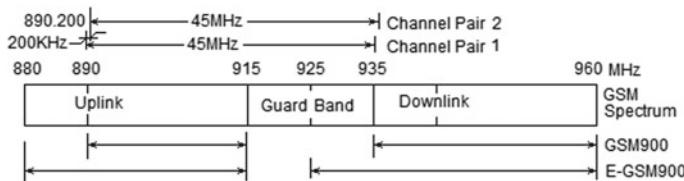


Fig. 1 Channel assignment in GSM/E-GSM

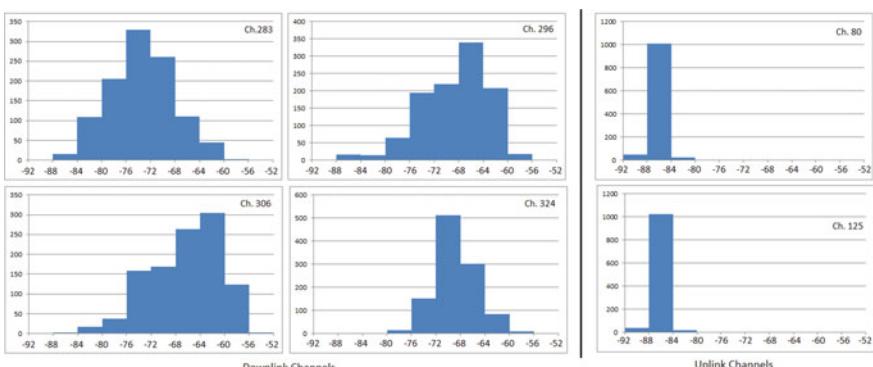


Fig. 2 Histogram indicating signal strength in typical downlink and uplink channels

Table 1 Absolute radio frequency channel numbers (ARFCN)

| | | | |
|-----------|---|---|------------------------------------|
| GSM 900 | $\text{FI}(n) = 890 + 0.2*n$ | $1 \leq n \leq 124$ | $\text{Fu}(n) = \text{FI}(n) + 45$ |
| E-GSM 900 | $\text{FI}(n) = 890 + 0.2*n$ $\text{FI}(n) = 890 + 0.2*(n - 1024)$ | $0 \leq n \leq 124$ $975 \leq n \leq 1023$ | $\text{Fu}(n) = \text{FI}(n) + 45$ |

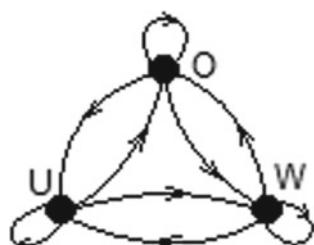
dataset is made available on GitHub [17]. Channel numbers in GSM bands are assigned according to Absolute Radio Frequency Channel Number (ARFCN), given in Table 1.

1.3 Markov Chains (MC) and LSTM Model

Markov Chains are a classical way to model stochastic time series, such as seen in telecommunication channels. Channel usage, as mentioned earlier, depends on various factors which needs to be used in channel prediction. However, Markov models require knowledge of only current state and hence computationally very efficient. The status of next channel can be predicted based on the past channel states using HMM [18]. A Markov process can be expressed as

$$P(q_{n+1} = S_{n+1} | q_n = S_n) \quad (1)$$

The process of building a Markov model consists of creating probability map for every state of the system from the training data. This can be easily accomplished by binning the input data into states of the system. In case of a RF transmission system, the signal strength indicates the state of transmission system: if the signal is good, transmission exists; if the signal is below a threshold, the channel is free; otherwise the state is unknown. Therefore the model consists of three states: Occupied (O), free (White space, W) or Unknown (U). (see Fig. 3). Further, as the predicted value of the system depends on present state, a state transition matrix indicating the probabilities of transition from current state to next state. These probabilities are used to predict the next possible state of the system. Given any particular state, the next state can be any one of the three possible states. Therefore, each row of the transition matrix will have a cumulative probability of 1. This will allow us to sample the next state based

Fig. 3 Three state markov model

on the current state. However, it is possible to consider the transition probability as a whole and estimate the next transition, instead of the state. It is also possible to extend Markov model to consider the states in previous several instances, and then use a Reinforcement Learning (RL) algorithm to predict the channel occupation.

At the simplest level, estimation of channel occupancy for a given length of time can be made using the transition matrix, by selecting one row that corresponds to current state. Generating the samples that meet the given probability distribution can be done in several ways. One easiest ways is to use the CDF of a row from the probability matrix, that corresponds to present state. Projecting a set of uniformly distributed samples along the y-axis on to the CDF curve will yield the samples that comply with the given probability distribution function. This is a standard technique in sampling systems and easy to implement. However, we need to understand that only the probabilities of samples can be predicted but not the exact sequence. Estimating the exact sequence requires advanced methods like Monte-Carlo methods or the use of Metropolis algorithms.

Complexity of building a useable model depends on the accuracy requirements. We have modeled a simple 3 state Markov model, choosing thresholds on incoming signal: Signals below -80 dB is considered as **White-space** (unused frequency band) while signals above -70 db are considered as **Occupied** and the range in-between is considered as **Unknown**. Transition probability matrices for these three states are derived from the observed data and used for prediction.

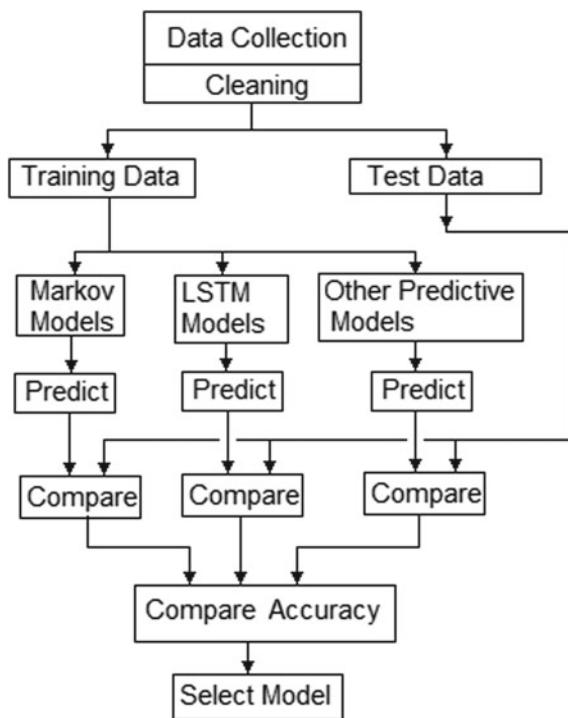
Sample function is used to generate random sampling. `np.argmax(probas)` gives maximum probability index. Markov predict function takes current state.

Long Short Term Memory (LSTM) is a popular artificial neural network (Recurrent) for time series prediction among the researchers [19]. In principle, LSTM overcomes the vanishing/exploding gradient problems by allowing the data to be accessible after long cycles of input data. This allows establishing otherwise difficult temporal relations between events that are separated by large time gap. Excellent discussion on LSTM is available in [19, 20].

2 Methodology

Three major components of the work carried in this research are (1) Data collection and Preparation, (2) data modelling and (3) Model evaluation. (see Fig. 4). Data collection is done using a spectrum analyzer and capturing its data using a program. The data contains several types of errors which are then cleaned using another small program. One round of scanning the entire GSM band takes minimum 3 s, after which it is repeated. The scanning process captures data in all 500 GSM channels, including downlink, uplink and control channels. Data is available as CSV or XLS format. These datasets are available on GitHub. Details of the data collection process are available in [2]. Captured data is filtered to match frequencies used by a known base station, which are then taken for further analysis. About 70% of the data in each channel is used for training the models while the remaining 30% is used for testing.

Fig. 4 Methodology used for the channel occupancy prediction



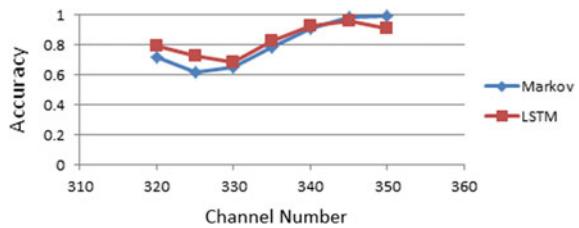
This data is then applied to various models, which in turn will predict the spectrum status in next steps matching the 30% test data. Accuracy is calculated on test set.

For Markov models, transition matrix is generated from the training set. For LSTM, the data is applied as it is but after cleaning. After the training is completed each model generates its prediction, whose statistical properties are compared with that of the test data set. Accuracy and loss for each model is computed and compared with other models. Overall procedure is indicated in Fig. 4.

3 Implementation

The Primary User data is collected from various urban area and rural locations like Belagavi, Pune, Sangli and Ashta. All the algorithms used in this work are implemented in Python using standard TensorFlow, Keras library. Markov models are built using hmm functions. In LSTM implementation three parameters to configuration are number of LSTM units, number of epochs and size of loopback, varying one parameter at a time. First, by keeping the number of LSTM units constant i.e. 128, keeping the constant hyper parameters of Adam optimizer and lookback = 20, we

Fig. 5 Downlink prediction accuracy comparison (channel versus accuracy)



change the number of epochs and batch size equally and observe the change in accuracy of the model. Second way is keeping number of epochs/batch size constant i.e. 100, hyper parameters of Adam optimizer constant. By varying the lookback, we observed the accuracy and loss of the model. Third way is to keep epochs/batch size constant i.e. 100, lookback = 20 and constant Adam optimizer hyper parameters and changing the number of units of LSTM. The observed accuracy of above methods is as shown in Fig. 5.

3.1 Data Collection

In predicting task, we need primary user data showing the activity of PU in certain area around the GSM Base station [2], respectively with the help of specialised software. It contains historical record of GSM traffic signals in that area. There are two types of activities recorded. One is instantaneous presence of PU or absence of PU. The dataset is provided pre-processed before applying it to the neural network or Markov Model. A special program is created to clean the dataset and make a standard format that suits to the models. The main objective of the data preparation is to process the data into desirable format which the model requires. The GSM trace dataset is arranged in such a way that it should be compatible for LSTM model. The channel selection is made randomly. For the selected channels the average probability of occurrence is calculated. In Markov model the transition probability is shown in transition matrix. This is second step of data preparation. Data processing aims to convert the cleared data into desired format. The same type of data set is used for both models. Following subsections (a) and (b) describe the data processing step for LSTM and Markov Model.

3.2 Data Processing for LSTM

As the look back size is set to 20, the dataset of size 20 will be initializing. Then 21st element in the data set would be generated in the next step. So likewise, the window of 20 elements would go on shifting from top to bottom. Each data instance was segmented into shorter segments by applying a sliding window of window size of

Table 2 Arrangement of input sequence for LSTM model with 20 loop back

| | | | | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----|------------|------------|------------|------------|------------|
| $P^{(1)}$ | $P^{(2)}$ | $P^{(3)}$ | $P^{(4)}$ | | ... | $P^{(18)}$ | $P^{(19)}$ | $P^{(20)}$ | $P^{(21)}$ | |
| | $P^{(2)}$ | $P^{(3)}$ | $P^{(4)}$ | $P^{(5)}$ | ... | $P^{(18)}$ | $P^{(19)}$ | $P^{(20)}$ | $P^{(21)}$ | $P^{(22)}$ |

20 and step size 1. Figure 4 shows that data processing step of a instance of LSTM method. Given a data instance, we denoted each activity in that data instance with $P^{(i)}$ where $i = 1, 2, 3, \dots, m$. by applying sliding window using window size 20 and step size 1, we are able to split the data instances into shorter segments as input data and take the next activity as the corresponding target. For example, the first segment of the data instance is a vector of length 20 which consists of $P^{(1)}$ to $P^{(20)}$ and it's corresponding target is $P^{(21)}$ (Table 2).

4 Results and Discussions

Very interestingly, both Markov model and LSTM were able to make the prediction on the 30% samples of the total data set with high degree of accuracy. We have compared the actual test data set and the predicted data set statistically rather than as a time sequence. As the RF signal is truly random, exact prediction is nearly impossible. However, the channel can be characterized by its statistical properties like distribution of power over a period or over a frequency range. Therefore, we have modelled our experiments to predict the statistical properties of channel rather than the actual signal itself. To illustrate this further, assume that we have 100 samples taken over a period of time. Let 50 of these samples be above -40 db while 20 of these are below -90 db. We say that the channel occupancy is identified as occupied:unknown:free::0.50:0.30:0.20. Actual time domain signal may have all the ‘occupied’ signals in any particular way within the 100 observed samples. This single step will reduce the computational complexity to a large extent.

It may be seen from Fig. 5 below that LSTM offers slightly improved accuracy over Markov model but differences are not statistically significant. Therefore, we conclude that both methods give approximately identical results. It may also be noted that the changes in prediction accuracy vary over the channels to a considerable extent. We attribute these changes in accuracy to the quality of data or to the nature of data in these channels and not dependent on the method used for prediction.

Visual representation of the channel prediction and the test data is given in Fig. 6a. The wavy line indicates the actual signal while the blue lines indicate the estimated state values, corresponding to the training and test data sets. As the predictions are made on the states rather than the strength of actual RF signal in each band, computational load is considerably reduced. The results are also more easy to compare and quantify. Similar match was also observed for Markov models. Figure 6b shows how the mean square error in prediction reduces over the number of epochs. This

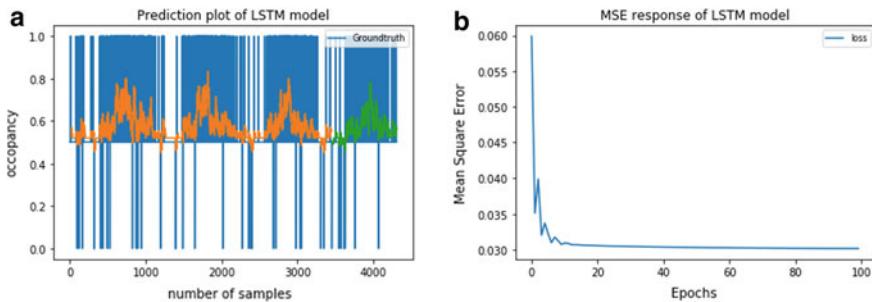


Fig. 6 **a** Comparison of predicted channel occupation, **b** variation of MSE with respect to epochs

is typical of an LSTM system. On the other hand, Markov models require only one iteration over the applied data set. This actually indicates better strength of Markov models over LSTM models.

5 Conclusions

Both Markov and LSTM models generate fairly identical estimates. This may be considered as a good result in itself, it also hints at improving the current models with other modern machine learning and deep learning methods. Some of these methods currently being tested are Convolution Neural Networks (CNN), Auto Encoder (AE) and Multi-Layer Perceptron (MLP). It is also possible to improve the quality of samples drawn by using other methods, which needs to be explored.

References

1. B. Fette, *Cognitive Radio Technology, Communication Engineering Series*, (Elsevier, Burlington, MA01803, USA 2006)
2. S. Bidwai, U.V. Wali, B. Shirgapur, S. Bidwai, Detecting white spaces for cognitive radio, in *International Conference on Smart Electronics Systems-2016*, KLE Dr MSS CET, Belagavi, Int. J. Technol. Sci. **3**(4), 57–60 (2016)
3. T.W. Rondeau, C.J. Rieser, T.M. Gallagher, C.W. Bostian, Online modeling of wireless channels with hidden markov models and channel impulse responses for cognitive radios, in *IEEE MTT-S Digest* (2004), pp. 739–742
4. K. Kim, I.A. Akbar, K.K. Bae, J.-S. Um, C.M. Spooner, J.H. Reed, Cyclostationary approaches to signal detection and classification in cognitive radio, (IEEE, 2007), pp. 212–215
5. C.-H. Park, S.-W. Kim, S.-M. Lim, M.-S. Song, HMM based channel status predictor for cognitive radio, in *IEEE Proceedings of Asia-Pacific Microwave Conference* (2007)
6. I.A. Akbar, W.H. Tranter, Dynamic spectrum allocation in cognitive radio using hidden markov models: Poisson distributed case, in *IEEE Southeast Conference* (2007), pp. 196–201
7. Q. Zhao, A. Swami, A decision-theoretic framework for opportunistic spectrum access. *IEEE Wireless Comm. Mag.* **14**(4), 14–20 (2007)

8. C. Ghosh, et al. Markov chain existence and hidden markov models in spectrum sensing, in *2009 IEEE International Conference on Pervasive Computing and Communications*, Galveston, TX, USA. (2009). <https://doi.org/10.1109/percom.2009.4912868>
9. S. Bidwai, N. Joshi, S. Bidwai, LSTM model for channel occupation prediction in GSM band, in *3rd International Conference in Electrical, Electronics, Communication, Computers and Optimization TECHNIQUES-2018*, 14–15 Dec. 2018 held at GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka. <https://doi.org/10.1109/ICE-ECCOT43722.2018.9001366>, pp. 331–335. Date added to IEEE Xplore: 20th Feb, 2020, Publisher: IEEE. Electronic ISBN:978-1-5386-5130-8, Print on Demand (PoD) ISBN: 978-1-5386-5131-5
10. S. Bidwai, N. Joshi, S. Bidwai, U. Wali, Deep learning predictive models for cognitive radio system, in *3rd Int conf on Data Engineering & communication systems*, RNSIT, Bengaluru, Karnataka, India, 19–20 Dec 2019, Int. J. Innov. Technol. Exp. Eng. (IJITEE) **9**(2S), 491–496 (2019)
11. K. Lan, H.S. Zhao, J. Zhang, C. Long, M. Luo, A spectrum prediction approach based on neural networks optimized by genetic algorithm in cognitive radio networks, in *10th International Conferences on Wireless Communications, Networking and Mobile Computing—(WiCOM 2014)* IET Digital Library, Beijing, China (2014)
12. H. Wang, et al., Wideband spectrum sensing method based on channels clustering and hidden markov model prediction, *Information* **10**, 331 (2019). <https://doi.org/10.3390/info10110331> www.mdpi.com/journal/information, published on 25 Oct (2019)
13. J. Xie, et al., Activity pattern aware spectrum sensing: a CNN-based deep learning approach. *Commun. Lett.* <https://doi.org/10.1109/lcomm.2019.2910176>. IEEE
14. S. Geirhofer, L. Tong, B.M. Sadler, Cognitive radios for dynamic spectrum access-dynamic spectrum access in the time domain: modeling and exploiting white space, *IEEE Commun. Mag.* **45**(5), 200
15. Y. Saleem, M.H. Rehmani, Primary radio user activity models for cognitive radio networks: a survey. *J. Netw. Comput. Appl.* **43**, 1–16 (2014)
16. H. Eltom, S. Kandeepan, et al., HMM based cooperative spectrum occupancy prediction using hard fusion, in *Proceedings of Communication Workshops, International Conferences Communication (ICC)*, Kaula lumpur, Malaysia, May 2016
17. GitHub link <https://GitHub.com/profssbssb/GSM-Datasets>
18. C.-H. Park, S.-W. Kim, S.-M. Lim, M.S. Song, HMM based channel status predictor for cognitive radio. 1–4 (2008). <https://doi.org/10.1109/apmc.2007.4554696>
19. S. Hochreiter, Long short-term memory, *Neural Comput.* **9**(8):1735–1780 (1997)
20. F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM. *Neural Comput.* **12**(10), 2451–2471 (2000)

Prediction of Network Attacks Using Connection Behavior



N. Aakaash, K. Akshaya Bala, Veerabrahmam Pranathi,
and Meenakshi S. Arya

Abstract An intrusion detection system (IDS) is created as it is important to configure the real working environment to analyze all the probabilities of attacks, which is not cost efficient. A network intrusion detection software shields a computer network from unauthorized users, including perhaps insiders. The incoming link is tested for any network attack and the type of attack. The existing IDS systems just predict whether the link is attacked or not providing no information about the type of attack. This paper proposes an ensemble learning voting classifier based technique for network intrusion detection. The method analyses the links to predict whether the network has been attacked or not. In addition, the proposed method also specifies the type of attack when applicable. Various machine learning algorithms are applied and compared to predict DOS, R2L, U2R, Probe and overall attacks accurately and the classifier with best accuracy is used for designing the voting classifiers. The efficiency of the proposed algorithm is evaluated using parameters like precision, recall and F1 score.

Keywords Intrusion detection · Ensemble classifier · Voting based classifier · Performance metrics

N. Aakaash · K. Akshaya Bala · V. Pranathi · M. S. Arya (✉)
SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India
e-mail: raina.arya@gmail.com

N. Aakaash
e-mail: akaash1998@gmail.com

K. Akshaya Bala
e-mail: akshayabala2203@gmail.com

V. Pranathi
e-mail: pranathi12361@gmail.com

1 Introduction

As the networked system are becoming wider and businesses continue to grow, most of their private data is going online, as a result of which the count and sophistication of network attacks and security breaches is also increasing drastically. Supposedly, there exist two sorts of organizations within the USA: the ones that have already been hacked and the ones that do not know that they have been hacked. For the maintenance of the framework and to secure private assets, companies are counting on network intrusion detection systems (NIDS) [1] to systematically observe the network traffic and report doubtful or abnormal behavior. Historically, most of the NIDS operate in one of the following two styles: Anomaly detection or Misuse detection. Misuse detection examines for particular signatures of known defamatory conduct, whereas an anomaly-based detection tries to create a model which includes normal network traffic patterns [2] and then detects abnormalities from those. Anomaly intrusion detection offers the intriguing ability in detecting odd attacks even before they have been categorized by the security analysts and being capable of detecting differences on the existing attack methods.

In order to create information for the intrusion detection system [3], it is needed to configure real working environment to analyze all the probabilities of attacks, which is not cost efficient. Data validation, data preprocessing and feature engineering are the phases involved. The data analysis phase (data validation, preprocessing, feature engineering) methodically discovers the patterns in the assembled information and projects them to the defined problem. It is a procedure of analyzing the data, modeling and transforming of data and deciding on how to organize, classify, interrelate, compare and display the data. Data quality targets mainly on the accuracy and reliability of data collected and used in an evaluation [4, 5]. Image processing, Web site analysis, medical applications, remote sensing, etc., have standard and legitimate ground truth databases for analysis. Likewise, most of the computer NIDS uses the KDD Cup99 for the classification analysis of network traffic. In the proposed system, KDD Cup99 dataset will be used for all experimentation purposes.

This paper is organised as follows: Sect. 2 presents the literature survey of the various algorithms and techniques available in the problem domain concerned. Section 3 describes the algorithms used for measuring the performance with respect to each attack. Section 4 discusses the various parameters used for determining the efficacy of the algorithms. Section 5 presents the output for each machine learning module [DoS, R2L, U2R, Probe, overall attack] and compares the accuracy of each algorithm for predicting the attack. Section 6 presents the proposed method and the results are discussed in Sect. 7. Conclusion and the future scope is discussed in Sect. 8.

2 Literature Survey

Intrusion detection system (IDS) is described as a software application which supervises the network or system activities and discovers if there is any malignant action. At present, hackers utilize various kinds of attacks for accessing important data. A large number of intrusion detection strategies, techniques and algorithms help to identify those attacks. The paper provides a detailed account about intrusion detection, kinds of intrusion detection strategies, various kinds of attacks, tools and methods for intrusion detection and the challenges in designing and deploying these systems [3]. Data cleaning involves the removal of null values, invalid and unwanted data. Data cleaning plays a major role in every process like validation and preprocessing as it helps to generate errors and develop solutions for larger databases, the significance of the same with respect to Intrusion Detection Systems is described in [4].

Real-world networks show cost of connections, where link costs frequently signify physical data. In [6], a new method suggests a sequence of new centrality indices for links in line graph. Every network has a link prediction issue based on studying the ‘proximity’ of nodes in a network. In an existing warning application [1], precise prediction of DoS attacks is the major aim in the network offense and defense work.

Trust for data might be set up in a way that it may be coming from certified source or that it has been approved by authenticating body, and so, there is constantly a sense of security and confidence. In the enormous and intricate social network framed utilizing the cyberspace or media transmission technology, the ID or forecast of any sort of socio-technical attack is constantly problematic. This challenge makes a chance to search various procedures, ideas and algorithms used to distinguish these sorts of networks based on specific examples, properties, structure and patterns in their linkage. It tries to find the private information in a vast social network by compacting into small networks and then examining using Apriori and Viterbi algorithms, respectively [7]. In the future, the prediction displays the security status of the direct network, and security managers can have corresponding actions to increase security of the network based on the outcomes.

Additionally, we study three factors that have large impact factors of the cyber-attacks [8]. The DoS attacker purposely jams the communication path by releasing noises for estimating the failure performance. When the attacker receives the ACK real-time information, then we can communicate the energy allocation problem of the dynamic attack and change it to locate the ideal solution [9]. Software-defined networks (SDNs) [10] block the connections in the switch level of such networks in order to manage the rise in the count of attacks.

Widely known ML techniques such as Naïve Bayes, Bayesian network (Bayes net) and decision tree (DT) have been utilized to discover the host that would be attacked on the basis of historic data. Experimented outcomes demonstrate that the accuracy of 91.68% is achieved through Bayesian network. Lastly, the efficacy of the proposed system is shown through a graphical example [11].

3 Algorithms

In the proposed system, various algorithms are tested to find out which one predicts a particular attack (DoS, R2L, U2R and Probe) with the best accuracy.

3.1 Logistic Regression

The aim of logistic regression is to discover the best fitting model to compare the relation between the dichotomous characteristics of a response variable. The dependent variable is a binary variable which is referred as 0 (no, failure) or 1 (yes, success).

$$\log(p \text{ attack}) = \log(p \text{ attack}/1 - p \text{ attack}) = \beta_0 + \beta_1 X_1 + \dots + \beta_j X_j$$

where p attack is the probability that a network has been attacked, β_0 is the intercept, $\beta_1, \beta_2, \dots, \beta_j$ are the coefficients and X_1, X_2, \dots, X_j are the variables.

Steps:

- Divide the problem of classification of good connection behavior into $n + 1$ binary classification problems (+1 because the index starts at 0).
- Predict the probability the observations which are in that single class.
- Prediction = max (probability of the classes)

3.2 K-Nearest Neighbor (KNN)

K-nearest neighbor is a supervised learning technique that saves all instances corresponding to the training data points in n-dimensional space.

Steps:

- Load the given dataset and initialize ‘ k ’ value to determine the number of neighbors.
- Calculate the distance between trained network connection behavior and new connection behavior.
- Sort the collection of distances in ascending order of connection behaviors by the distant space and selects the first ‘ k ’ entries from collection of network connections.
- So, selected ‘ k ’ entries are given and return the mode of the ‘ k ’ labels.

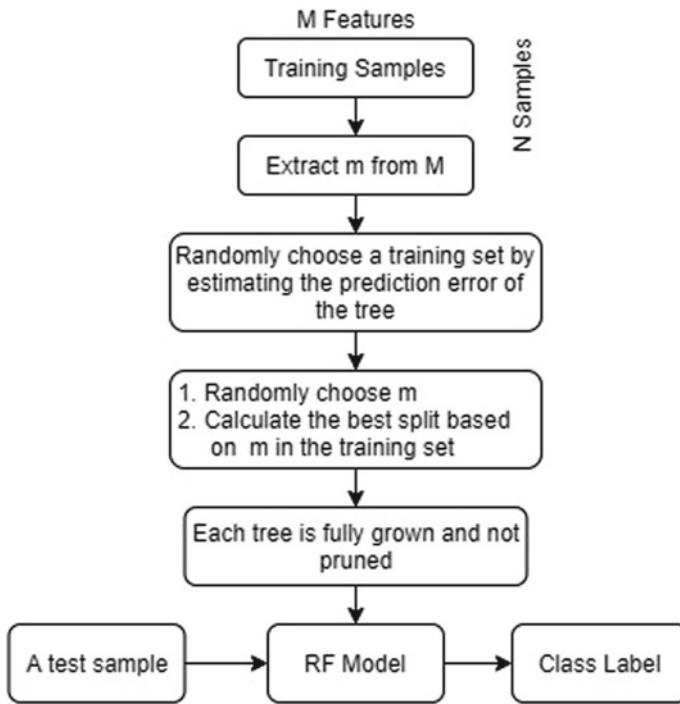


Fig. 1 Performance of random forest

3.3 Random Forest

Random forest is also known as random decision forests. It is one of the ensemble learning algorithms utilized for classification and regression including those that are operated by building a multitude of decision trees at training time and outputs the class which may be either mean prediction (regression) of the individual trees or mode of the classes (classification) (Fig. 1).

3.4 Naïve Bayes

A Naive Bayes classifier is a supervised learning algorithm based on the Bayes' Theorem, which expects that features are statistically not dependent.

Bayes' Theorem

$$P(a|x) = \frac{P(x|a)P(a)}{P(x)}$$

where,

- $P(a|x)$ is the posterior probability, $P(x|a)$ is the probability of data ‘ x ’ given that the hypothesis ‘ a ’ was true, $P(a)$ is the prior probability of class, $P(x)$ is the probability of the data.

3.5 Decision Tree

Decision tree algorithm is also a supervised learning technique. It can be utilized to solve classification and regression problems. A tree structure is used to find out the problem solution, where each leaf node represents a class label and the internal node of the tree represents the attributes. Using the decision tree, any Boolean function can be represented on discrete attributes.

Steps:

- Find out the best connection behavior (such as protocol type, source byte size, destination byte size and service type) and place it on the tree’s root node.
- Split the training set into subsets. Each subset of training dataset should have a consistent value for the given connection behaviors.
- Find the leaf nodes of all the branches by repeating step I and II on every other subset.

Assumptions made using a decision tree:

1. Firstly, the entire training set is taken as the root.
2. Feature values are preferred to be categorical.
3. Based on the attribute values, there is a recursive distribution of records.
4. Statistical methods are used to order attributes as the internal node.

3.6 Support Vector Machine

Support vector machine is a supervised learning technique that analyzes the data utilized for regression and classification. SVM model is an illustration of the data values as points in space, mapped so that the data values of different classifications are isolated by a fine space beyond what many would consider possible.

Steps:

- Represent different classes of network attack types in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner to minimize

the error and to divide the given datasets into classes to find a maximum marginal hyperplane (MMH).

- Generate hyperplanes iteratively that segregates the classes of network types in best way.
- Choose the hyperplane that separates the classes correctly.

4 Parameters

Various parameters of the classifiers are considered to calculate the accuracy of each algorithm.

Precision: Precision is a classifier's ability to not label a negative instance as a positive one. It is the ratio of TP to the sum of TP and FP.

TP—True Positive

FP—False Positive

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

Recall: Recall is a classifier's ability to find all the instances that are positive. It is the ratio of TP to the sum of TP and FN.

FN—False Negatives

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

F1 Score: A weighted harmonic mean of recall and precision so that the best score is 1 and the worst is 0 is called the F1 score. By the thumb rule, the weighted average of F1 should be used to compare the classifier models and not the global accuracy.

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

Specificity: Specificity (SP) is estimated as the number of precise negative predictions by the total negatives. The best specificity is 1, whereas the worst is 0.

$$SP = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Sensitivity: Sensitivity (SN) is computed as the number of precise positive predictions by the total positives. The best sensitivity is 1, whereas the worst is 0.

$$SN = \frac{TP}{TP + FN} = \frac{TP}{P}$$

5 Performance Measurement of ML

DOS:

| Parameter | LR | DT | RF | SVC | NB | KNN |
|-------------|------|------|------|------|------|------|
| Precision | 0.79 | 0.81 | 0.79 | 0.79 | 0.79 | 0.80 |
| Recall | 0.98 | 0.96 | 0.98 | 0.99 | 0.99 | 0.79 |
| F1 score | 0.88 | 0.88 | 0.87 | 0.87 | 0.87 | 0.80 |
| Sensitivity | 0.98 | 0.96 | 0.97 | 0.99 | 0.99 | 0.79 |
| Specificity | 0.18 | 0.27 | 0.18 | 0.09 | 0.09 | 0.38 |
| Accuracy | 79.0 | 78.8 | 78.8 | 77.2 | 77.3 | 78.0 |

R2L:

| Parameter | LR | DT | RF | SVC | NB | KNN |
|-------------|------|------|------|------|------|------|
| Precision | 0.62 | 0.63 | 0.62 | 0.77 | 0.77 | 0.63 |
| Recall | 0.94 | 0.90 | 0.87 | 0.93 | 0.04 | 0.88 |
| F1 score | 0.75 | 0.74 | 0.73 | 0.75 | 0.08 | 0.73 |
| Sensitivity | 0.93 | 0.89 | 0.87 | 0.93 | 0.04 | 0.87 |
| Specificity | 0.16 | 0.23 | 0.24 | 0.17 | 0.98 | 0.24 |
| Accuracy | 62.8 | 62.7 | 62.3 | 62.3 | 42.4 | 48.6 |

U2R:

| Parameter | LR | DT | RF | SVC | NB | KNN |
|-------------|------|------|------|------|------|------|
| Precision | 0.83 | 0.83 | 0.83 | 0.83 | 1 | 0.83 |
| Recall | 1 | 1 | 1 | 1 | 0.22 | 1 |
| F1 score | 0.91 | 0.91 | 0.91 | 0.91 | 0.36 | 0.91 |
| Sensitivity | 1 | 1 | 1 | 1 | 0.21 | 1 |
| Specificity | 0 | 0 | 0 | 0 | 1 | 0 |
| Accuracy | 83.3 | 83.4 | 83.4 | 83.4 | 34.2 | 83.4 |

Probe:

| Parameter | LR | DT | RF | SVC | NB | KNN |
|-----------|------|------|------|------|------|------|
| Precision | 0.82 | 0.82 | 0.82 | 0.82 | 1 | 0.82 |
| Recall | 1 | 1 | 1 | 1 | 0.2 | 1 |
| F1 score | 0.90 | 0.90 | 0.90 | 0.90 | 0.34 | 0.90 |

(continued)

(continued)

| Parameter | LR | DT | RF | SVC | NB | KNN |
|-------------|------|------|------|------|------|------|
| Sensitivity | 1 | 1 | 1 | 1 | 0.20 | 1 |
| Specificity | 0 | 0 | 0 | 0 | 1 | 0 |
| Accuracy | 81.8 | 81.8 | 81.8 | 81.9 | 35.6 | 81.8 |

6 Proposed System

Machine learning supervised [6] classification algorithms will be utilized to give the network connection dataset and hail out patterns, which would help in foreseeing if the connection is attacked or not. The dataset for the current research was build by combining datasets from multiple sources to create a simplified dataset, and afterward, various algorithms would be employed to extricate patterns and to acquire results with the greatest accuracy. The intrusion [12] detector learning goal is to assemble a predictive model (i.e., a classifier), which is qualified to find if the link has been attacked and the type of network attack. The objective is to examine the ML-based techniques for improved packet connection transfers forecasting by prediction of results in the best accuracy from the ensemble learning voting classifier technique (Fig. 2).

Four types of attacks are considered in general: DoS, R2L, U2R and Probe. The KDD Cup99 dataset is used to train the system of when and how an attack occurs. The dataset consists of various connection behaviors and an attack class. The data is split into an 8:2 ratio for training and testing, respectively. Using feature engineering, the features are acquired from the raw data by mining. The accuracy of the prediction of each attack using various classifiers is compared. The ensemble voting classifier is used finally to predict the most accurate result.

The input given to the system will be source file size, destination file size, protocol type, flag type and the service. Thus, the system will output the type of attack occurred along with its category (Fig. 3).

Data preprocessing and cleaning: The prepared dataset is preprocessed which involves tasks such as removal of null values, checking for duplicate values and visualising the nature of data and correlation among various parameters through plots and graphs.

Training and Test Dataset Creation: The preprocessed dataset is partitioned into two datasets: training [80%] and testing [20%].

Classifier building: The various algorithms discussed earlier viz. K-nearest neighbor, logistic regression, random forest, support vector machine, decision tree and Naïve Bayes are applied on the training dataset to check the efficacy of each algorithm for predicting various types of attacks. On the basis of the accuracy of these algorithms, a voting classifier is built to find whether the connection is attacked or not and its type.

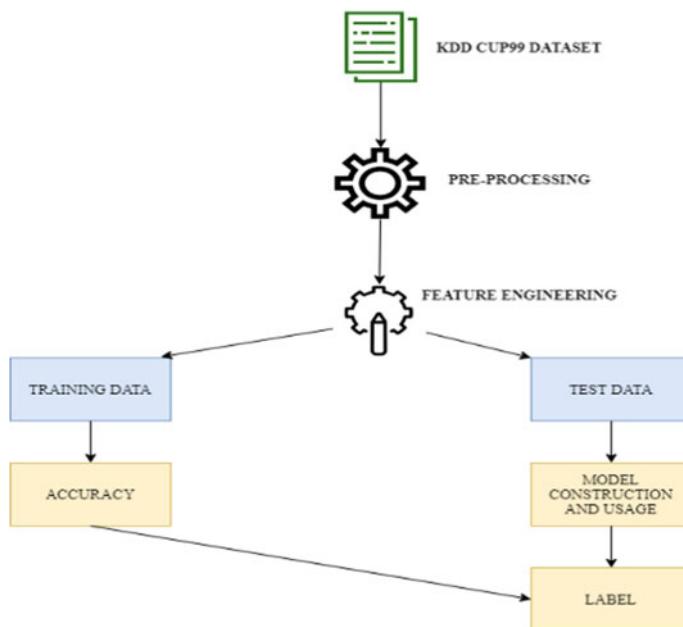


Fig. 2 Proposed system architecture

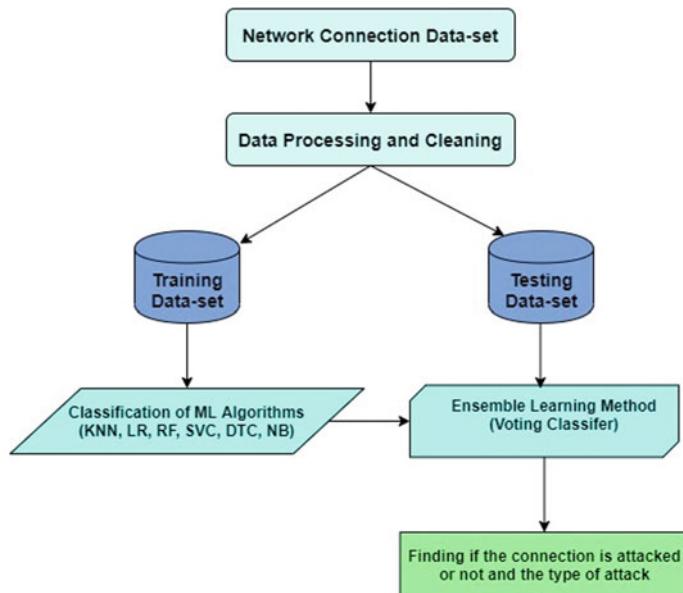


Fig. 3 Data flow of the system

6.1 Modules

Data Validation: The KDD Cup99 dataset is taken and the null attributes are dropped. Various types of attacks are classified using one-hot encoding (Fig. 4).

Voting for attacks: Various machine learning classifier methods are compared, and the one which has the best accuracy in predicting the attack is considered.

Calculation of maximum votes: An ensemble learning voting classifier is utilized to predict the type of attack on the basis of the maximum votes.

Predicting the label: Different connection behaviors are given as an input, and the type of network attack is predicted as the output.

7 Results

The output of each module which compares the accuracy of each algorithm while predicting the attack is given in a bar graph format.

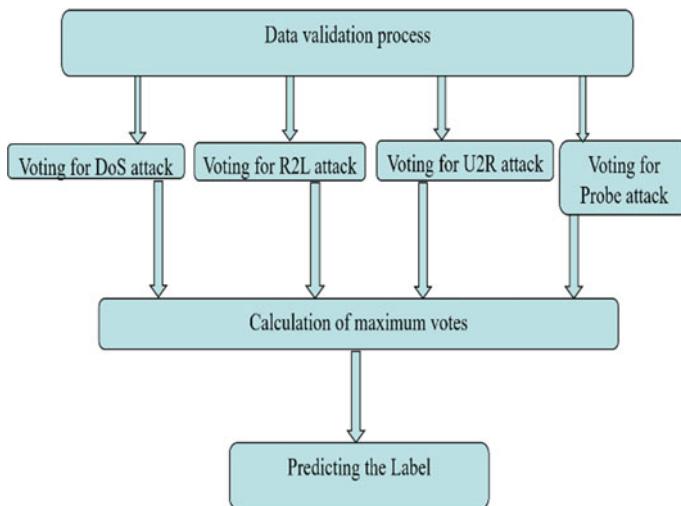
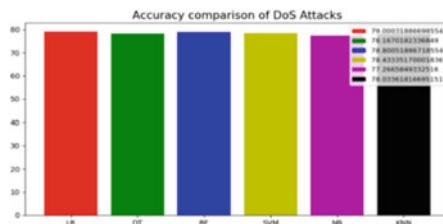


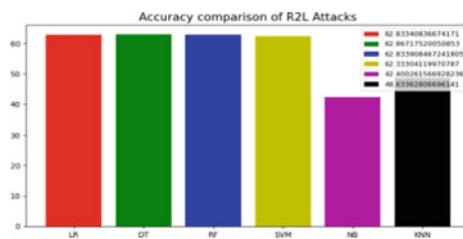
Fig. 4 Various modules involved in the system

7.1 Dos Attack Prediction



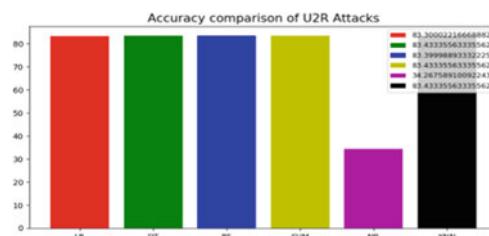
Logistic regression has delivered the highest percentage of accuracy—79%. Random forest delivers the second highest percentage of accuracy—78.8%.

7.2 R2l Attack Prediction



Decision tree has delivered the highest percentage of accuracy—62.86%. Random forest and logistic regression have delivered the second highest percentage of accuracy—62.83%.

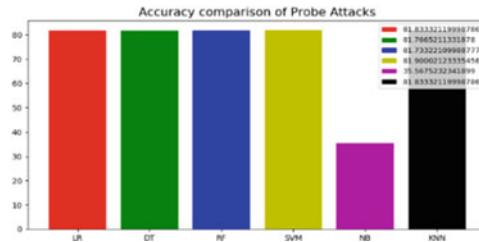
7.3 U2r Attack Prediction



Support vector machine, K-nearest neighbor and decision tree have delivered the highest percentage of accuracy—83.43%.

Random forest delivers the second highest percentage of accuracy—83.33%.

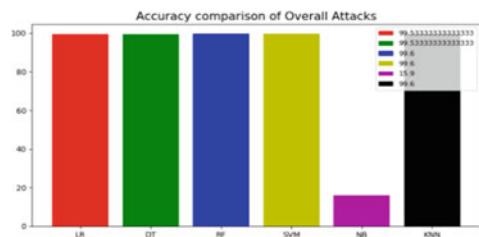
7.4 Probe Attack Prediction



Support vector machine has delivered the highest percentage of accuracy—81.9%.

Random forest and K-nearest neighbor have delivered the second highest percentage of accuracy—81.83%.

7.5 Overall Attack Prediction



In overall attack, random forest, support vector machine and K-nearest neighbor deliver the highest accuracy—99.6%.

8 Conclusion and Future Scope

This paper proposed a framework of network attack prediction classifier. The analytical process began with data preprocessing, exploratory data analysis and model

construction and usage. The performance metrics for each of the algorithm was calculated for each kind of attack [DoS, R2L, U2R, Probe] to find out which algorithm shows the highest accuracy and works the best. The proposed algorithm performs very well as various algorithms were compared and the algorithms with the highest accuracy for overall attacks were used for building an ensemble voting classifier to achieve an accuracy of 99.6%. From the developed model, it can be inferred that the ensemble learning based machine learning techniques are useful in the prediction of models that can help the network sectors reduce the long process of analysis and eradicate any human error.

In the current research, the existing datasets were used for development of the system. The algorithm can further be applied to real-time network data for Intrusion Detection in live networks. Neural networks or deep learning networks can also be applied to predict the network intrusions.

References

1. Z. Abedjan, X. Chu, D. Deng, R.C. Fernandez, I.F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, N. Tang, Detecting data errors: where are we and what needs to be done? *VLDB* **9**(12), 993–1004 (2016)
2. F. Chiang, R.J. Miller, Discovering data quality rules, in *VLDB* (2008)
3. M. Tiwari, R. Kumar, A. Bharti, J. Kishan, Intrusion detection system. *Int. J. Techn. Res. Appl.* **5**(2), 38–44 (2017)
4. G. Webb, J. Vreeken, Efficient discovery of the most interesting associations. *ACM TKDD* **8**(3), 1–31 (2014)
5. P.C. Arocena, B. Glavic, G. Mecca, R.J. Miller, P. Papotti, D. Santoro, Messing up with bart: error generation for evaluating datacleaning algorithms. *VLDB* **9**(2), 36–47 (2015)
6. C. Fu, M. Zhao, L. Fan, X. Chen, J. Chen, Z. Wu, Y. Xia, Q. Xuan, Link weight prediction using supervised learning methods and its application to yelp layered network. *IEEE Trans. Knowl. Data Eng.* (2018)
7. P. Ranjan, A. Vaish, Apriori Viterbi model for prior detection of socio-technical attacks in a social network. *IEEE Trans. DOI* (2014)
8. J. Wu, L. Yin, Y. Guo, Cyber attacks prediction model based on Bayesian network. *IEEE Trans. Parallel Distrib. Syst.* (2012)
9. H. Zhang, Y. Qi, J. Wu, L. Fu, L. He, DoS attack energy management against remote state estimation. *IEEE Trans. Control Netw. Syst.* **5**(1), 383–394 (2016)
10. S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, B. Yang, Predicting network attack patterns in SDN using machine learning approach, in *IEEE Conference on Network Function Virtualization and Software Defined Networks* (2016)
11. Y. Guan, X. Ge, Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks. *IEEE Trans. Sign. Inf. Process. Over Netw.* **4**(1), 48–59 (2017)
12. Z.-T. Li, J. Lei, L. Wang, D. Li, A data mining approach to generating network attack graph for intrusion prediction. *IEEE Trans. Fuzzy Syst. Knowl. Disc.* (2007)

Multi-Face Recognition Using CNN for Attendance System



Prasanth Vaidya Sanivarapu

Abstract There has been a great technological development in face recognition in the last two decades. Machines these days are verifying the identity automatically for secure transactions, for access control of buildings, surveillance and security tasks, etc. Attendance is recorded everywhere like library, schools and colleges. The professor calling out student names and recording attendance is the traditional approach. The proposed approach deviates from such systems by introducing a novel approach that uses image processing for taking attendance. The face recognition algorithm is utilized to identify the faces for accurate attendance. Student database consisting of images, register numbers and names is collected. Real-time class images and Avengers dataset are considered as sample images for identifying the person. Student faces are trained with the features extracted from them. In recognition of face, local binary pattern histogram algorithm (LBPH) classifier is utilized. If the system recognizes faces, the attendance gets marked immediately; else, it shows unknown.

Keywords CNN · Multiple face recognition · Attendance system · Face recognition

1 Introduction

The face of any person is his/her identification. Computer algorithms use facial recognition technique to verify or identify an object or a person through images. Recognizing multiple faces in real time is very challenging due to the complex computation of the processing and obtaining acceptable level of accuracy. The technology of face recognition is used in identifying a person from an image or a video's video frame. It is also described as a biometric artificial intelligence-based system that can

P. V. Sanivarapu (✉)
Department of Computer Science and Engineering,
MLR Institute of Technology, Hyderabad, Telangana, India
e-mail: vaidya269@gmail.com

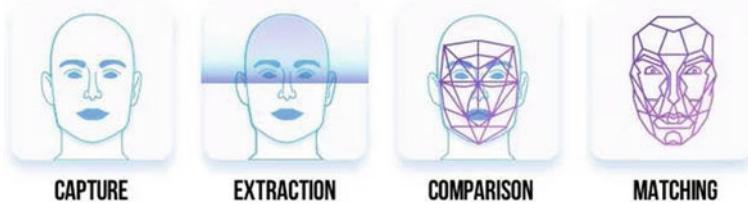


Fig. 1 General face recognition system procedure

uniquely identify a person by pattern analyzing based on the person's facial shape and textures [1]. In the proposed system, all the faces present in the image are recognized and labeled. If any unknown or undesired person is recognized, then the system alerts using alarm and all the details of people present in the image are updated in Excel sheet including time stamp (for any future use). This feature has many real-time applications such as recognizing a criminal or prohibiting the entrance of unknown people in a restricted area. The general face recognition system procedure is depicted in Fig. 1.

Images are classified and clustered by similarity and object recognition performed upon by using CNN that is deep artificial neural network [2]. Optical character recognition (OCR) is performed by convolutional networks to computerize text and make possible the natural language processing on handwritten and analog documents [3, 4]. CNNs have a great efficacy toward face recognition domain in identifying objects, humans and so on [5]. The convolutional networks do not regard to images like people do which is why one must understand what an image means as it is given as input to and processed by a convolutional network. Convolutional networks interpret images as three-dimensional objects or volumes and not as flat canvases measured by only height and width as color images possess RGB encoding which helps the network in perceiving the images as people do [6]. Convolutional network acquires a color image as a cuboid whose depth is three layers referred to as channels, one for each letter in RGB and whose height and width are measured by the pixel count [7].

2 Existing System

Omkar et al. [8] proposed a deep face recognition for recognizing faces in the large database. This kind of face recognition involves collecting a large-scale database, training and testing the face recognizer with the collected database. Deep face recognition is highly effective in the case when convolutional neural network (CNN) is used on a very large-scale training dataset. To minimize the manual automation, the dataset involved must have very less noise. When trained efficiently using CNN, desirable results can be achieved.

Bruce et al. [9] proposed an algorithm with individual dissimilarities in face perception and person recognition. They published a narrative view of face recognition. They stated that there are many ways to perceive and recognize faces. They mentioned two particular fields of application—the use and recruitment of “super-recognizer” (SRs) in forensic operations and passport scrutiny or other identity photos used for gaining permission to restricted places [10].

Gao et al. [11] proposed Semi-Supervised Sparse Representation-Based Classification for Face Recognition With Insufficient Labeled Samples. The goal of this face recognizer is to recognize faces accurately even when there is very less training data. A few labeled instances imply that it is difficult to remove these nuisance variables. To solve the problem, they implemented a method called semi-supervised sparse representation-based classification. This is based on new work on sparsity. The gallery dictionary consists of instances of each individual and a variation dictionary representing linear nuisance examples.

Ding et al. [12] proposed Multi-Directional Multi-Level Dual-Cross Patterns for Robust Face Recognition. Face recognition becomes a challenging task when there is deterioration of quality of face image and large variations of illumination, expression and pose. To be more accurate, we should develop an efficient face image descriptor and an extensive face representation scheme. They took four databases into consideration and checked for the accurate schemes. MDML-DCPs constantly achieved the best results on the four databases. This method is more suitable for practical applications.

Xie and Zisserman [13] proposed Multicolumn Networks for Face Recognition. This is a set-based face recognizer (two sets are to be recognized if they are of same person or not). Neural networks are used to calculate quality weight based on both content and visual quality, relative to each other over a given set. These quality weightings improve the verification performance. To increase the performance, we use more powerful backbone network for end-to-end training of multicolumn networks, and quality assessment should not only be based on the feature descriptor from the whole face, but it has to be focusing on more detailed facial parts.

Ramalingam and Mouli [14] proposed local directional number pattern reduced by dimensionality for face recognition. The descriptor choice plays a major role here. The descriptor used here is the local directional number pattern (LDNP). Resultant LDNP after assignment of 3-bit code is divided to form histogram-based descriptor. These bins are further manipulated to form the final descriptor. This face recognizer functions on standard databases.

3 CNN (Convolutional Neural Networks)

CNNs are image processing algorithms that use deep learning to perform both tasks, generative and descriptive, often using machine vision that includes image and video recognition, along with recommender systems and natural language processing (NLP) [15]. The layers of neurons are arranged in such a way as to cover the

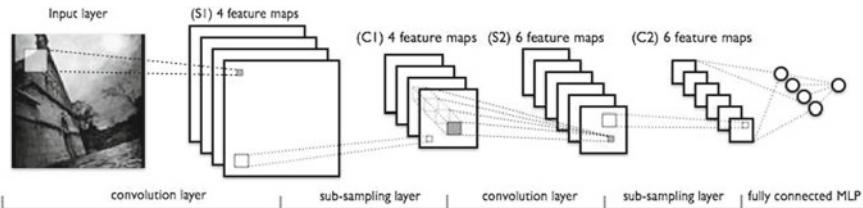


Fig. 2 Convolutional neural networks structure

entire visual field avoiding the piecemeal image processing problem of traditional neural networks [16]. CNN structure is shown in Fig. 2.

Initially, standard normalization is processed for the given input image. In the next stage, feature extraction and reduction of redundancy are done with the help of pooling layers. All the extracted features are combined and are further processed by sending them to fully connected CNN layers.

4 Proposed System

Multiple face recognition system (MFRS) in real time for attendance system is proposed using CNN. In MFRS, the person is identified from digital data (images, videos). CNN is utilized in the MFSR for identifying the faces. The efficiency of CNN in image recognition is one of the main reasons why it is used. In this system, all the faces present in the image are recognized and labeled. If any unknown or undesired person is recognized, then users are alerted using alarm and all the details of people present in the image are updated in Excel sheet including time stamp (for any future use). This feature has many real-time applications such as recognizing a criminal or prohibiting the entrance of unknown people in a restricted area. The proposed architecture is shown in Fig. 3. The proposed method is divided into two modules. The first module is multiple face recognition; in this module, multiple faces in an image are recognized using embedding model. The second module is alerting and result recording. In the second module, an alert is sent if an unknown person is identified in the input image and also the recognized data is stored systematically by their names, time and date of recognition and so on in the database, which can be used for analysis. The modules are described in detail in the following sections.

4.1 Multiple Face Recognition

The multiple face recognition module is divided into two stages. In the first stage, training of the faces is done using CNN. In the second stage, testing of the proposed system is done with test images to find the accuracy of the system. The detail description of the two stages is given below.

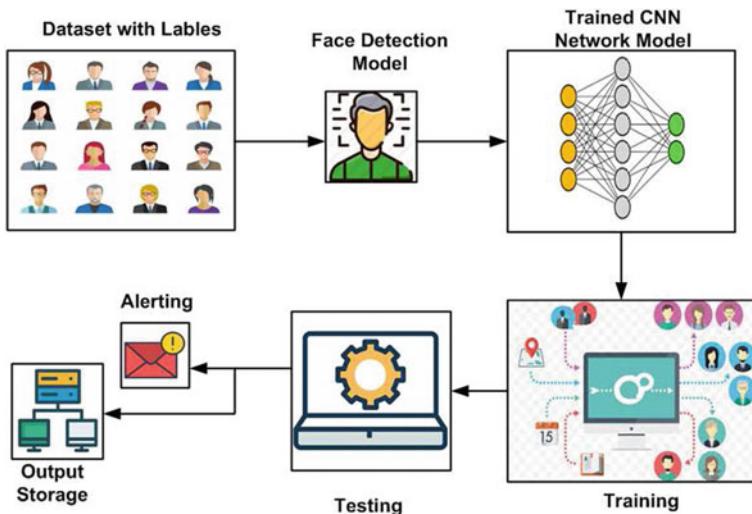


Fig. 3 Multiple face recognition system architecture

4.1.1 Training

The step-by-step process of training is provided in Algorithm 1.

Algorithm 1 Training Algorithm

Input: Individual images from dataset

Output: Trained system with embedding vector for each image

- 1: Read the images with unique identity label(names)
 - 2: Detect the faces using Haar cascade face detection model
 - 3: Select a model with pre-trained network
 - 4: Construct 128-d embeddings for each face in the dataset
 - 5: Trained system with embedding vectors
-

Firstly, the images present in the database are quantified. To improve accuracy of the system, trained network is considered which can create 128-d embeddings. The reason for considering the trained network is that it takes extra time to do in pre-training network from the scratch. As the trained networks are already available and are working accurately, these can be considered for easy processing. After training, the embedding vectors are utilized for comparing and recognition of the images. Here, two models are utilized: a face detection model for identifying the face in an image and another model, a face embedding model for transforming an image into meaningful vector for training the faces. Finally, in the face classification, a simple KNN model is utilized.

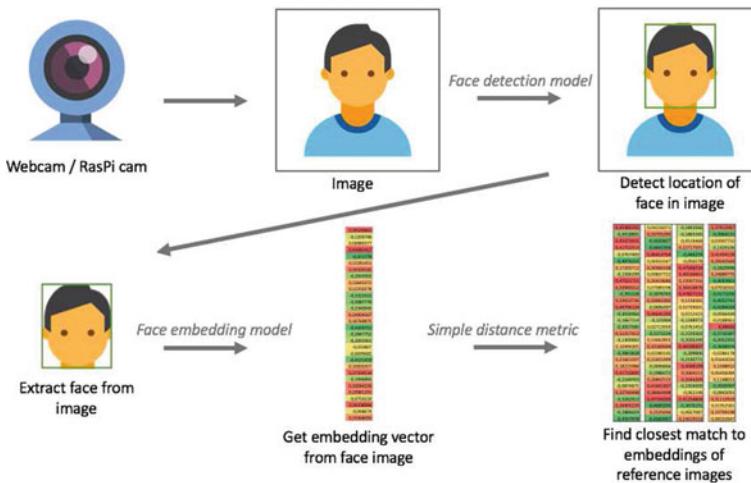


Fig. 4 MFSA testing phase

4.1.2 Testing

The testing of the proposed MFSA process is shown in Fig. 4. In this stage, stream of images is taken from webcam, and the face detector model will identify the faces in the image. The embedding vector of the reference image is compared with the vectors of the trained images to find the similarity. The output is displayed with boxes around the faces with labels. In testing stage, the same models (face detection model and face embedding model) used in training stage are utilized. The step-by-step process of testing is provided in Algorithm 2.

Algorithm 2 Testing Algorithm

Input: Stream of images from webcam or camera

Output: Recognized images with labels

- 1: Read the images from the webcam
 - 2: Detect the faces using Haar cascade face detection model
 - 3: Construct 128-d embedding for face in the input image
 - 4: Compare the vector embeddings of trained data
 - 5: Display boxes of multiple recognized faces with labels
-

After completion of the training stage, 128-d face embedding vectors for each image in the dataset is generated. The trained system will help to load the image, encodings and face names from the database, then detect all faces in the input image and compute their 128-d encodings and finally, display the processed image with box labeled.

4.2 Alerting and Result Recording

After completion of multiple face recognition, multiple faces are detected. If any of them is unknown or undesired, then the system will alert the users. This makes the system to be advantageous for security purposes. For future reference, to check the details of faces recognized, this system records the details along with a time stamp in the database.

5 Result Analysis

In the proposed system, the Avengers dataset is considered and trained six superheros (Evans, Hemsworth, Robert Dowrey Jr, Ruffalo, Scarlett, Clint) with five images each.

The algorithm is tested are shown from Fig. 5. From these images, it can be analyzed that all the superheroes are recognized with their real names and for those who are not trained are labeled as unknown. The recognized persons are stored with the name, date and time in the database, and the unknown faces are not stored. The data is stored in the database as shown in Table 1.



Fig. 5 Test images

Table 1 Multiple face recognition system database

| MFRS database | Test image-1 | Test image-2 | Test image-3 |
|------------------|---------------------|---------------------|---------------------|
| Date and time | 2020-02-13 16:12:28 | 2020-02-13 16:14:18 | 2020-02-13 16:16:12 |
| Recognized faces | Evans | Evans | Evans |
| | Hemsworth | Hemsworth | Hemsworth |
| | Robert Dowrey Jr. | Scarlett | Scarlett |
| | Scarlett | Ruffalo | Ruffalo |
| | Ruffalo | Clint | Clint |
| | Unknown | Unknown | Robert Dowrey Jr. |
| | Unknown | | |

6 Conclusion and Future Scope

The proposed approach deviates from such systems by introducing a novel approach that uses image processing for taking attendance. The face recognition algorithm is utilized to identify the faces for accurate attendance. In the future, this system has a scope of extension of the application to serve the requirements of security-based door unlocking system, attendance management system and surveillance. This application can further be automated to be triggered at regular intervals to automate the face recognition mechanism providing better security aspects.

References

- Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in *European Conference on Computer Vision* (Springer, 2016), pp. 499–515
- H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, *IEEE Trans. Med. imaging* **35**(5), 1285–1298 (2016)
- Y. Li, G. Wang, L. Nie, Q. Wang, W. Tan, Distance metric optimization driven convolutional neural network for age invariant face recognition. *Pattern Recognit.* **75**, 51–62 (2018)
- A. Singh, S.P. Vaidya, Automated parking management system for identifying vehicle number plate. *Indones. J. Electr. Eng. Comput. Sci.* **13**(1), 77–84 (2019)
- Y. Zhang, D. Zhao, J. Sun, G. Zou, W. Li, Adaptive convolutional neural network and its application in face recognition. *Neural Process. Lett.* **43**(2), 389–399 (2016)
- S.P. Vaidya, A blind color image watermarking using brisk features and contourlet transform, in *International Conference on Recent Trends in Image Processing and Pattern Recognition* (Springer, 2018), pp. 203–215
- B. Kwolek, Face detection using convolutional neural networks and Gabor filters, in *International Conference on Artificial Neural Networks* (Springer, 2005), pp. 551–556
- O.M. Parkhi, A. Vedaldi, A. Zisserman et al., Deep face recognition, in *BMVC*, vol. 1 (2015), p. 6
- V. Bruce, M. Bindemann, K. Lander, Individual differences in face perception and person recognition (2018)
- S.P. Vaidya, Multipurpose color image watermarking in wavelet domain using multiple decomposition techniques, in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (IEEE, 2018), pp. 251–255
- Y. Gao, J. Ma, A.L. Yuille, Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Trans. Image Process.* **26**(5), 2545–2560 (2017)
- C. Ding, J. Choi, D. Tao, L.S. Davis, Multi-directional multi-level dual-cross patterns for robust face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(3), 518–531 (2015)
- W. Xie, A. Zisserman, Multicolumn networks for face recognition. *arXiv preprint arXiv:1807.09192* (2018)
- S.P. Ramalingam, C.M.P.V.S. Sita, et al., Dimensionality reduced local directional number pattern for face recognition. *J. Amb. Intell. Hum. Comput.* **9**(1), 95–103 (2018)
- S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in *Twenty-ninth AAAI Conference on Artificial Intelligence* (2015)
- H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5325–5334

Simulating the Concept of Self-Driving Cars Using Deep-Q Learning



Akhilesh P. Patil, Pramod Sunagar, Karthik Ganesan, Biswajit Kumar,
and Kartik Sethi

Abstract Prior to the deployment of a critical software or functional hardware equipment, it is always suitable to test the working of the equipment in an environment that tests the capabilities of the functionality to the fullest. The equipment can be deployed in the real-world only on successful clearance granted to the test scenarios. Thus, an efficient foolproof simulated environment for any type of hardware is customary in a product development cycle. Thus, through this research, we aim at creating a suitable environment for a self-driving car which consists of simple test cases the agent will have to pass before it can be deployed in the real world. The outcome developed as the end-product of this research will help serve companies that are involved in the development of technologies for the purpose of building a full-fledged vehicle with autonomous capabilities. This product will not only help them to run the algorithms designed for technology but also help in designing suitable test case scenarios. The simulating agent can be visualized as a game-playing agent where an environment similar to that of a typical game comprises of the environment where the agent overcomes certain obstacles and challenges to complete a particular task.

Keywords Reinforcement learning · Deep Q-learning · Markov decision process · Action selection policies

A. P. Patil (✉) · P. Sunagar · K. Ganesan · B. Kumar · K. Sethi
Ramaiah Institute of Technology, Bangalore, Karnataka, India
e-mail: Apatil1997@gmail.com

P. Sunagar
e-mail: pramods@msrit.edu

K. Ganesan
e-mail: bksinha4497@gmail.com

K. Sethi
e-mail: karthik.sethi1997@gmail.com

1 Introduction

Reinforcement learning is different from other forms of learning such as supervised learning and unsupervised learning in the sense that there is an absence of a target variable upon which the system can be trained to satisfy the user goal. This method of learning is also called as “learning with a critic”. A critic differs from a teacher in that it only tells us how well we have been doing a task and not the purpose of the intended task itself. The feedback from a critic is scarce and when it comes, it comes late. This leads to the credit assignment problem. After taking several actions and getting the reward, we would like to assess the individual actions we did in the past and find the moves that led us to win the reward so that we can record and recall them later on. The solution to this problem is to model the agent using a Markov Decision Process. In certain applications, such as the self-driving agent, the agent does not know the entire state of the system exactly. It is equipped with certain sensors that feed input to the agent, which the agent then uses the probabilities calculated on the observations to model the state of the environment and take the necessary actions. Let us say, we have an autonomous agent that navigates within a particular area. The agent may not know its exact location in the area, or what else is there in the area. The agent may have a camera with which sensory observations are recorded. This does not tell the car its state exactly but gives some indication as to its likely state. For example, the car may only know that there is an obstacle to its right but may not be aware of the obstacles that may be encountered in the future.

Through this project, an autonomous agent within a software environment is simulated with the help of a deep Q-learning algorithm which is a domain of reinforcement learning. The primary purpose of the self-driving agent is to reach a goal state from a particular point in the environment within a specific boundary which in our case would be a road from the source to the destination. Initially, the agent exceeds its boundaries to appear to be in a random state of motion. In such situations, where the agent falls out of its boundaries, a negative reward is given to the agent so that the agent learns from its mistakes and does not perform similar fallacies in the future. At any given point of time, the environment is in a certain state. For example, a “safe state” for the self-driving agent would be a scenario where the agent proceeds in its path without collisions or causing hindrance to the other entities within the environment. The decision-maker in this case—the self-driving agent makes a possible set of legal actions such as taking the right deviation to avoid a collision or a deviation to avoid crossing the boundaries.

2 Related Work

Profound Reinforcement Learning requires an enormous measure of information to be fruitful. On account of reenactment conditions, the poor introductory achievement rate is satisfactory, yet in reality situation, this case may not hold. A paper referred

to in [1] utilizes Deep Q-gaining from Demonstrations (DQfD), which uses little arrangements of showing information to greatly quicken the gaining procedure even from moderately little measures of exhibit information. This calculation gives us the correct methodology for us to begin assembling a precisely reproduced condition.

An amusement playing specialist is regularly alluded to have the capacity of beating human capacities with regards to a mimicked situation. Consequently, building up an operator turns into an essential experiment. The paper referred to in [2], presents the main profound learning model to effectively gain control arrangements legitimately from high-dimensional tactile information utilizing support learning. The model is a convolutional neural system, prepared with a variation of Q-realizing, whose input is crude pixels and whose yield is an esteemed work evaluating future prizes. We apply our strategy to seven Atari 2600 recreations from the Arcade Learning Environment, with no alteration of the engineering or learning calculation. This methodology finds a decent application for the present task where we reproduce a specialist like a diversion playing operator.

In the paper referred to in [3], a multiagent-based profound deterministic arrangement inclination calculation for independent heading to control the consistent activities of vehicles was created. For this reason, a street test system condition in which multi-operators every now and again experience path changes.

It is required to dependably test a driverless self-ruling vehicle in a reproduced situation before it very well may be sent in reality. A paper referred to in [4], presents a way to deal with actualize a test system to test such vehicles. It incorporates an investigation of the best in class in driverless vehicle reproduction and talks about the particular destinations that this specific test system intends to accomplish so as to help to test the connections of various driverless autos in urban systems.

Programming for self-driving vehicles requires serious testing to dodge lethal mishaps and to permit the right task in certifiable conditions. Reenactment structures permit emulating the conduct of complex frameworks, for example, self-ruling vehicles utilizing rearranged models of this present reality. Subsequently, they are significant devices permitting to stretch out part and utilitarian tests to address interconnections between sensors, actuators, and controllers in virtual and predefined situations. In a paper, referred to in [5], a methodology that joins the advantages of both abnormal state and low-level test systems to execute segment and connector models. Vehicle and traffic specialists can pick the most appropriate dimension of detail for their application and incorporate true condition information from OpenStreetMap.

In a paper referred to in [6], a Deep Stochastic IOC RNN Encoder decoder structure, DESIRE, for the assignment of future forecasts of numerous communicating operators in powerful scenes. This venture utilizes PC vision to make forecasts of the event of future occasions inside a domain. The model initially acquires a differing set of theoretical future forecast tests utilizing a contingent variational autoencoder, which are positioned and refined by the accompanying RNN scoring-relapse module. Tests are scored by representing collected future prizes, which empowers better long haul key choices like IOC structures.

Information affiliation issues are a significant part of numerous PC vision applications, with multi-object following being a standout amongst the most unmistakable

models. A commonplace way to deal with information affiliation includes finding a chart coordinating or arrange stream that limits an entirety of pairwise affiliation costs, which are frequently either hand-created or learned as direct elements of fixed highlights. Alluding to paper in [7], we can see that it is conceivable to learn highlights for system stream based information affiliation by means of backpropagation, by communicating the ideal of a smoothed system stream issue as a differentiable capacity of the pairwise affiliation costs.

In paper [8], assessment of the exhibition of approach molding calculation utilizing 26 human instructors. We inspect if the calculation is reasonable for human-produced information on two distinct sheets in a pac-man area, contrasting execution with a prophet that gives investigate dependent on one known winning arrangement. Maybe shockingly, we demonstrate that the information produced by our 26 members yields stunningly better execution for the specialist than information created by the prophet. This may be on the grounds that people don't dishearten investigating various winning strategies. Moreover, we assess the effect of various verbal directions, and various translations of quiet, finding that the convenience of information is influenced both by what guidelines are given to instructors, and how the information is deciphered.

In paper [9], it exhibits the structure and acknowledgment of path keeping capacity of a self-governing electric go-truck. The necessity of the framework concerning this paper is exploring the vehicle on a shutdown track with street markings, in view of data from an optical camera with path identification capacities. To accomplish this assignment, two arrangements were utilized, a twofold circle control with feedforward load unsettling influence remuneration and a nonlinear strategy. The control calculations were planned and tuned in a Hardware-In-The-Loop structure. The nonlinear calculation was actualized on two diverse equipment gadgets and approved in CarSim—Matlab programming condition.

In paper [10] Q-learning (Watkins 1989) is a basic route for operators to figure out the proper behavior ideally in controlled Markovian areas. It adds up to a gradual strategy for dynamic programming which forces restricted computational requests. It works by progressively improving its assessments of the nature of specific activities at specific states. This paper shows and demonstrates in detail a union hypothesis for Q-learning dependent on that illustrated in Watkins (1989). We show that Q-learning merges to the ideal activity esteems with likelihood 1 insofar as all activities are over and again examined in all states and the activity esteems are spoken to discretely. We additionally sketch expansions to the instances of non-limited, however, retaining, Markov situations, and where many Q qualities can be changed every cycle, as opposed to only one.

In paper [11] SUMO an open-source traffic recreation bundle including the reenactment application itself just as supporting devices, chiefly for system import, and request displaying. SUMO examines an enormous assortment of research themes, primarily with regards to traffic the board and vehicular correspondences. We depict the ebb and flow condition of the bundle, its significant applications, both by research subject and by precedent, just as future advancements and augmentations.

3 Methodology

3.1 Initial Approach

We first discuss the role of reinforcement learning and the role it plays in Q-learning. An environment can be defined as a state-space consisting of agents that perform certain actions and change their states. When the agent progresses to a state, it is awarded certain rewards and negative rewards which may further decide the change of state of the agent. Hence, we can say that the agent learns through these rewards. The agent may need to perform these actions in sequential order so that a positive outcome is possible.

This leads us to formulate the Bellman equation in terms of State— s , Action— a , and Reward— r . Suppose the agent performs a series of actions and reaches a state of a positive outcome. Henceforth, the agent tracks back and visualizes the previous state. It marks the preceding state as a state of a positive outcome too, due to the fact that this state had led to another state of a positive outcome. Thus, we can observe here that the Bellman equation can be viewed as a dynamic programming problem and hence a recursive equation can be formulated.

$$V(s) = \max \{ R(s, a) + \gamma (V(s')) \} \quad (1)$$

where R is the reward, $V(s')$ is the new state and γ is the discounting factor.

The State Score— V reduces by a factor γ at every iteration indicating that one state of the environment is better than the other. Ultimately, the states closer to the final state will have a higher value of V , and the discounting factor reduces the score through the succeeding iterations as we move further away from the goal state.

3.2 Intuition Behind Q Learning

When an agent plans to search for a path from the start state to a goal state, it does so in either a deterministic approach or a non-deterministic approach. In the deterministic search, the agent is a hundred percent sure of going to a particular state. In the non-deterministic search, the agent looks for three or more different existing paths to reach the goal state. There may be a ten percent chance of taking the first path, an eighty percent chance of taking the second path, and so on. The agent decides by probabilistic measures to take the best possible path.

A Markov Process is one where the future outcome depends on only the current state and not the sequential occurring events. A Markov Decision Process is a non-deterministic approach that describes a mathematical framework that the agent will use to move to different states. This leads us to the change in Eq. 1 where the $V(s')$ will change to the expected value that the agent changes its state. Hence, we model the probabilities of the occurrence of a change of state.

$$V(s) = \max \left(R(s, a) + \gamma * \left\{ \sum_{s'} P(s, a, s') * V(s') \right\} \right) \quad (2)$$

We further add living penalties by changing the R that is the Reward part of the equation. The Q —in Q-learning defines the quality of the action. A lucrative action is chosen by the action. It is calculated as

$$Q(s, a) = R(s, a) + \gamma * \left\{ \sum_{s'} P(s, a, s') * V(s') \right\} \quad (3)$$

$V(s')$ can be written as $Q(s', a')$ which denotes the lucrative state the agent takes.

Before taking the action, we have the Q learning equation as $Q(s, a)$ as in (3). After the agent takes the action, we have the equation,

$$Q_{t-1}(s, a) = R(s, a) + \gamma * \max_{a'} (Q(s', a')) \quad (4)$$

This leads us to the equation of Temporal Difference as follows

$$T \cdot d = Q_t - Q_{t-1} \quad (5)$$

If we calculate the value of Q in terms of an instance in time t and a preceding instance in time $t - 1$ we get the equation,

$$Q_t(s, a) = Q_{t-1}(s, a) + \alpha * T \cdot d_t(a, s) \quad (6)$$

where alpha is the learning rate.

Substituting the value of Temporal Difference in the above equation we get,

$$\begin{aligned} Q_t(s, a) &= Q_{t-1}(s, a) \\ &+ \alpha * \{R(s, a) + \gamma * \max(Q(s', a')) - Q_{t-1}(s, a)\} \end{aligned} \quad (7)$$

Observing the equation, we can conclude that alpha should never be 1 or 0. If alpha is 1, the term Q_{t-1} cancels out and we are left with a similar equation and if alpha is 0 the new state will be the same as the previous state.

4 System Architecture

4.1 System Overview

In the proposed system design, we intend to use deep learning techniques to model the controls of automobile based on input sensors. This architecture intends to

replace rule-based approach to semi-autonomous systems by using the reinforcement learning techniques and the agent learns to drive in a fully automated way for the given constraints.

We have presented a AI-based model which will be simulating the working of a Self-Driving car, we would be building a Deep Q model for the simulation, basically, a Deep Q model is where an agent learns and understands the functioning of an object over time by itself, hence here the cars would be able to understand the road maps and various other vehicles around it to gain more understanding as to what has to be done and over time as the data will increase the understanding will improve and the car can go on roads without any human control and it would eventually become a safe way of transportation.

The purpose of this System architecture is to provide a description of how the new AI-based model will enable autonomous vehicles to learn and drive vehicles in an autonomous way. The system design provides the architecture upon which the AI base model is intended to be used on; hence the purpose of the system architecture is to provide a high-level overview of the Deep Q model and the Autonomous car responses.

We have used a part of Reinforcement Learning (RL) which is Deep Q-Learning to make our self-driving car. All RL algorithms have three basic key elements, State, Actions, and Rewards. This is how these elements are defined. The state the car is in consists of four variables as shown in Fig. 1.

The Sensor Red, Sensor Yellow, Sensor Blue refer to the left camera, center camera and the right camera respectively referred in the figure. The first three come from the three sensors present on the front of the car. Each sensor (Red, Yellow, and Blue), senses how many pixels of sand are within a 10-pixel radius of itself. So, this would mean that if there was an area of sand on the left of the car, the blue sensor would sense more sand than the yellow sensor. This allows for the car to determine where sand is, and therefore which direction to travel. The last two variables represent the orientation of the car. This is measured in degrees, where 0° would be pointing straight up. The negative value of the orientation is also added as it helped to improve performance while tuning. The three possible actions are to turn 20° clockwise, turn 20° anti-clockwise, or do not turn. The main rewards include -5 —if the car drives into an obstacle, -0.1 —if the car drives away from the objective, 0.1 —if the car gets closer to the objective. These are only the main rewards. We have defined other rewards based on the performance of the car. For example, we later realized that the car was hugging the edges of the map too closely, so We gave it a negative reward whenever it was 10 pixels from the edge. In practice, these rewards can be defined, however, you like, to gain the results you'd like to achieve.

4.2 Network Architecture

This is the architecture of the neural network: Input Layer: 5 Nodes (One for each state-input), Hidden Layer: 30 Nodes, Output Layer: 3 Nodes (One for each

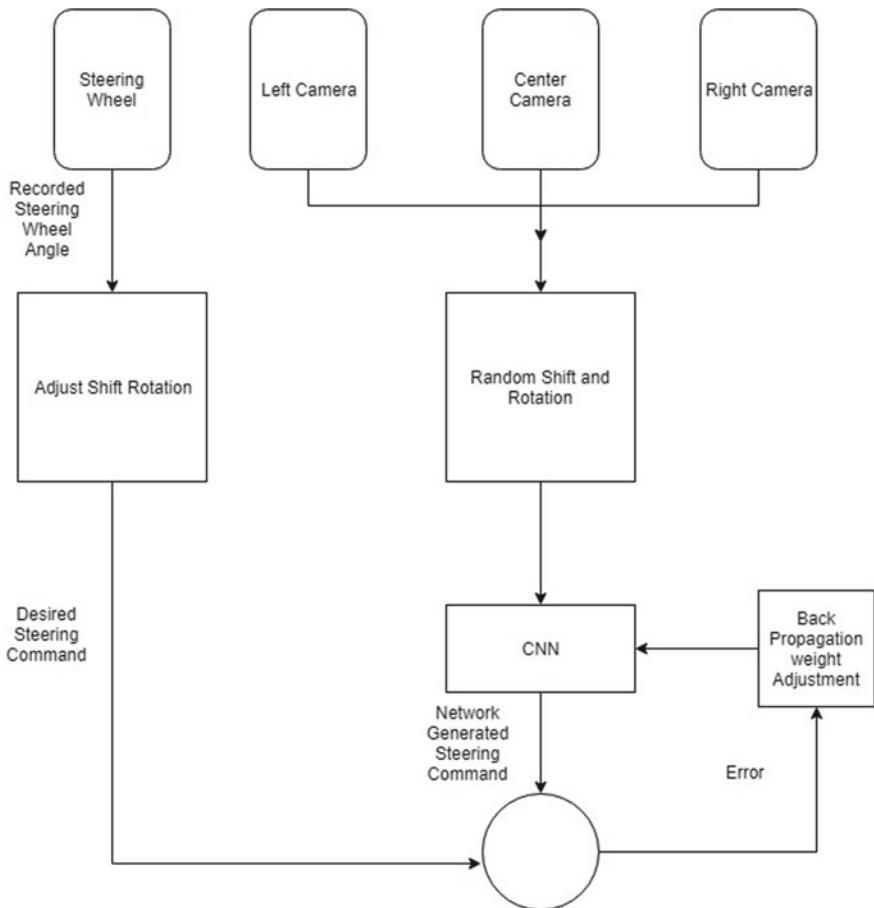


Fig. 1 Flow diagram of the system

action), Activation Functions: ReLU, Optimizer: Adam Optimizer. One hidden layer is normally enough for simpler problems such as this one. If any more were required then it would take longer to train and would not result in significant performance improvements (Fig. 2).

4.3 Deep *Q*-Training

The DeepMind system uses a deep convolutional neural network filled with layers of tiled convolutional filters to mimic the effects of receptive fields. Reinforcement learning becomes unstable or divergent when a nonlinear function approximator like neural network is used to represent *Q*. This instability comes from the correlations

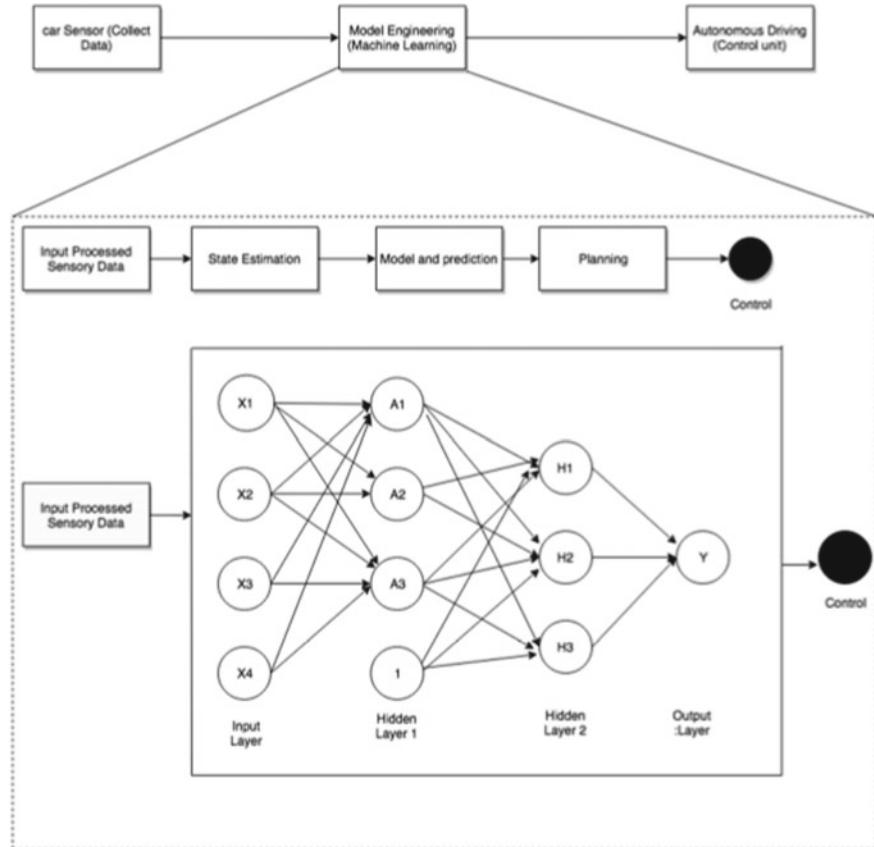


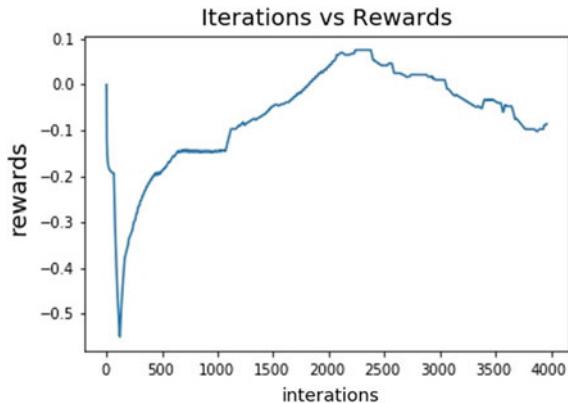
Fig. 2 Network architecture diagram

present in the sequence of observations. The fact that small updates to Q may significantly change the policy and the data distribution, and the correlations between Q and the target values.

5 Results

When it comes to reinforcement learning, we are interested in how quickly or how slowly a particular agent tries to learn from the different circumstances it is tested against. Through this measure, a baseline can be drawn explaining the different scenarios of testing. Some experiments were conducted in the simulating environment which included varying the gamma factor or the discount factor, testing the agent

Fig. 3 Graph for horizontal lines



along horizontal lanes, testing the agent along vertical lanes, and finally, tweaking the learning rate to 0.005. This is depicted in the following graphs as shown in Figs. 3, 4, 5 and 6.

The graph is shown in Fig. 3 explains the number of iterations vs the rewards when the agent is tested for horizontal lanes. It is observed that the agent takes a sharp rise in learning between iterations 0–500. Hence, we can tell that the learning agent can quickly adapt itself to learning horizontal lanes on the road. Between iterations 1500–2000, the positive rewards begin to set in which indicates that the agent has completely learned the pattern in the environment.

This graph shows the variations in the rewards with respect to the iterations for learning vertical lanes. The point where the agent starts learning is similar to Fig. 3. But, the point of positive rewards begins during iteration 2000–2500. We can further observe that there are a lot of spikes hovering around the slightly negative mark in the range of 3000–6500 iterations which shows that the agent finds it hard to understand the vertical lanes.

Fig. 4 Graph for vertical lines

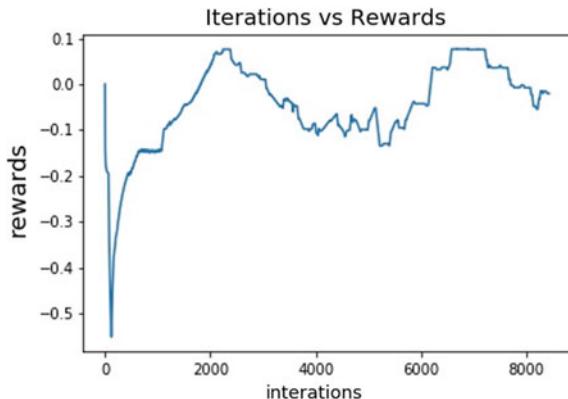


Fig. 5 Graph for learning rate of 0.005

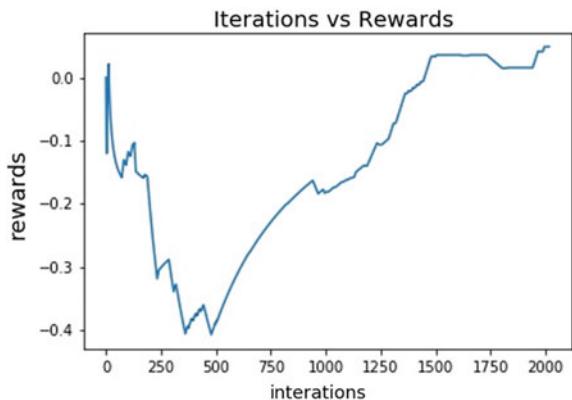
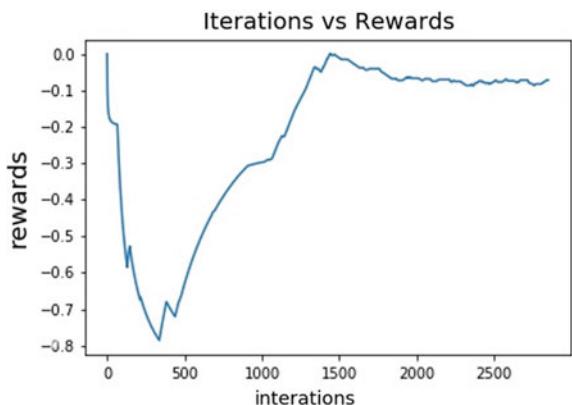


Fig. 6 Graph for discount factor of 0.6



The graph in Fig. 4 shows the variations in the rewards with respect to the iterations for learning vertical lanes. The point where the agent starts learning is similar to Fig. 1. But, the point of positive rewards begins during iteration 2000–2500. We can further observe that there are a lot of spikes hovering around the slightly negative mark in the range of 3000–6500 iterations which shows that the agent finds it hard to understand the vertical lanes.

The graph in Fig. 5 shows the change in rewards with iterations for a learning rate of 0.005. The slow learning rate indicates that it takes a lot of time for the agent to begin the process of learning, but once the point of positive approach is reached, the point of positive rewards is attained very quickly. We can observe that the positive approach point is at around 500 iterations and it attains the positive rewards very quickly between the range of 1250–1500 iterations.

The discount factor or the gamma factor as explained in Sect. 3. B was tweaked to 0.6. We can observe that the point of positive approach is around 300–500 iterations

Table 1 Quantification measures for iterations versus rewards

| | Point of positive approach | Point of attainment of positive reward | Point of asymptote |
|---------------------|----------------------------|--|------------------------------|
| Horizontal lanes | 0–500 iterations | 2000–2500 iterations | Greater than 4000 iterations |
| Vertical lanes | 0–500 iterations | 2500–3000 iterations | Greater than 8000 iterations |
| Learning rate—0.005 | 500–600 iterations | 1250–1500 iterations | Greater than 2000 iterations |
| Discount factor—0.6 | 250–500 iterations | 1250–1500 iterations | 1500 iterations |

and there is a gradual spike in the learning of the agent. Once it attains positive rewards at a very early stage of around 1300 iterations, the slope tends to stay at the positive stage indicating there is no further learning required unless a new scenario for the test is introduced.

6 Conclusion

The main objective of this research is to develop a suitable environment where the concept of self-driving cars can be tested on different scenarios so that when deployed in the real-world, there is minimum damage done. For the purpose of quantifying the results, we define certain measures from the graphs such as point of positive approach, point of attainment of positive reward, and point of the asymptote. The point of positive approach is the point where the agent starts receiving positive rewards and the slope starts ascending towards a positive reward side. The point of attainment of positive reward is the point at which the agent begins to learn the right things and hence tries to stay at this point to complete its learning. These quantifications are measured along with the number of iterations shown on the positive X-axis as seen in Figs. 3, 4, 5, and 6. To sum up, the three measures a table is shown below showing the point of a positive approach, the point of attainment of positive reward and point of asymptote (Table 1).

References

1. T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, G. Dulac-Arnold, J. Agapiou, J.Z. Leibo, Audrunas Gruslys; deep Q-learning from demonstrations; AAAI Publications, in *Thirty-Second AAAI Conference on Artificial Intelligence*
2. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller; Playing atari with deep reinforcement learning, [arXiv:1312.5602](https://arxiv.org/abs/1312.5602)

3. H. Yi, Deep deterministic policy gradient for autonomous vehicle driving. Int'l Conf Artif. Intell. [IICAI'18]
4. M.C. Figueiredo, R.J.F. Rossetti, A.M. Rodrigo, R.A. Braga, L.P. Reis, An approach to simulate autonomous vehicles in Urban traffic scenarios
5. F. Grazioli, E. Kusmenko, A. Roth, B. Rumpe, M. von Wenckstern, Simulation framework for executing component and connector models of self-driving vehicles, Software Engineering, RWTH Aachen University, German
6. N. Lee, W. Choi, P. Vernaza, C.B. Choy, P.H.S. Torr, M. Chandraker, DESIRE: distant future prediction in dynamic scenes with interacting agents, [arXiv:1704.04394](https://arxiv.org/abs/1704.04394)
7. S. Sachulter, P. Vernaza, W. Choi, S. Chandraker, Deep network flow for multi-object tracking, in *Proceedings of IEEE CVPR 2017*, Hawaii, USA
8. T. Cederborg, I. Grover, C. Isbell, A. Thomaz, Policy shaping with human teachers, in *International Joint Conference on Artificial Intelligence (IJCAI 2015)* (2015)
9. O. Toro, T. Becsi, S. Aradi, Design of a lane keeping algorithm of autonomous vehicle. *Periodica Polytechnica Transportation Engineering* (2015)
10. C.J. Watkins, P. Dayan. Q learning: technical note. *Mach. Learning.* **8** 279–292, (1992)
11. D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of sumo-simulation of urban mobility. *Int. J. Adv. Syst. Measure.* **5**(3, 4), 128–138 (2012)

Dynamic Cloud Access Security Broker Using Artificial Intelligence



Debayan Bhattacharya, Adeep Biswas, S. Rajkumar,
and Ramani Selvanambi

Abstract The Cloud technology is now a key component in almost every computing system. It is most widely used in commercial fields. Cloud storage can store almost any type of data. Thus the security of the cloud component is now a major issue. With the increasing amount of cloud users with various different types of requests, the threats faced by the platform are also changing. There is no static threat in the current scenario and thus the need for dynamic security access. This paper deals with this new type of threat which deals with several different constraints and threat levels to come out with an answer to either give access or not, to the cloud perimeter. This paper deals with the very first security layer of the cloud parameter and enhances its performance using the latest artificial intelligence technologies, neural network algorithms and helps in restricting data breach or unauthorized access in a very early stage, preventing an organization or the cloud vendor from huge data losses. The access granted is all labeled and can be further used for taking intelligent decisions to make the parameter more secured. This will have its application across a wide domain of systems and will help in safely storing data anywhere across any cloud platform.

Keywords Cloud security · Cloud perimeter · Neural network · Dynamic decision · Artificial intelligence

D. Bhattacharya (✉) · A. Biswas · S. Rajkumar · R. Selvanambi
Vellore Institute of Technology, Vellore, India
e-mail: srdebayan@gmail.com

A. Biswas
e-mail: adeep.biswas12@gmail.com

S. Rajkumar
e-mail: srajkumar@vit.ac.in

R. Selvanambi
e-mail: ramani.s@vit.ac.in

1 Introduction

Computer systems nowadays prefer having a storage which can be accessed anywhere across any location on mostly any device. The ease of access of the resources has considerably increased the performance and usability. But due to this uprising technology of storing resources and important data in cloud platforms there has been a steady increase in threats. Earlier mostly the individual systems with local storage were prone to attacks but due to the replacement or distribution of these different parts of the system new threats have come. The attacks are also now distributed and are carried on platforms which contain them. But as the resources are placed as different modules, attacking a particular component becomes very hard. If the attacker decides to pinpoint a particular resource and decrease its usability from the system then it has to first pass through the clouds security parameter. The security parameter is always well guarded by a security access broker. It needs to be fed with logic of what elements to allow to pass through it. The logic is of a very static nature and has a predefined decision state then can very quickly be overpowered by various types of intrusions, including various types of malwares and data packets. This makes the system very vulnerable to attacks and increases the risk factor of any business. With a dynamic cloud access security broker the decision scenario is constantly changing and the threat perception of the parameter is also increasing. Due to these the security access will be regulated by the data which has been generated from the neural network using the cutting edge artificial intelligence technologies. The decisions taken are also stored for upcoming threat scenarios and to come out with better test results. Many different parameters have been tested to get better result and to come out with a binary decision.

2 Literature Review

Cloud computing has gained in popularity as it can be used across many different domains and due to its scalability. The data is also available widely across all platforms [1]. The CASB works as a bridge between the user and the apps hosted in the cloud. It uses various proxies and API with regard to the application. The CASB widely helps in the encryption, automation and thus giving the user a very wide scope of browsing or searching through the platform to get a better result. It primarily acts as one of the most important control brain which helps in taking decisions for controlling the access to the perimeter [2]. There are some disadvantages of the cloud network which makes it very threat prone. Denial of service attacks, man in the middle attacks, malware injections, side channel attacks and authentication attacks are few of the most common attacks it faces. There are various different components in which the attack may happen with the most common being during the transactions, data storage and while the resources are being shared. There is a steady increase in the number of attacks due to the ever increasing use of cloud technologies in almost every domain

[3]. Cloud computing is booming in the recent times due to the increasing number of virtual data centers which is actually a very inexpensive option for the enterprise and its infrastructure. Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are being very widely used by the users. In return the users are completely free from the control of the computing infrastructure. They are also free from deploying or hosting their own applications. The cost is hugely saved as now they can only pay for the hardware or the software which they are using. This is very widely called as the new age pay as you go approach and is extremely popular [4].

The states of the data which are normally threatened can be widely divided into two types; The data at rest which consists mainly of the data stored in the cloud and the data in transit which is the data moving in and mostly out of the cloud. The three main pillars of Information Security, the confidentiality, integrity and availability based on the procedures and protection mechanisms. Out of these, the most significant fact is the matter which exposes the data of the above mentioned two states [5].

One of the major concerns of cloud computing is multi-tenancy. It is mainly in the scenarios where the users using the resources and sharing them run on a single server. It also occurs when both the attacker and the victim are on the same server using the same kind of hardware and operating system [6].

The data which is being stored in the public cloud will always be at risk. The cloud mostly has a centralized storage facility which is always a very lucrative target for hackers. The storage resources have a very complicated architecture. There is an extremely complex combination and use of hardware and software. Due to a very small breach there can be exposure of data [5].

The artificial neural networks are used in getting better results in taking decision of giving access to the cloud. These neural networks are made to simulate the exact conditions and duplicate the behavior of the various networks of neurons which exists in our brain. They consist of different layer of neurons which work as different modules. The neural network need to be first trained by a known dataset as an input and then it is to be tested with the output against the training data [7].

The cloud vendors must look after all the end user concerns of security and the storage of confidential data. The end users must always be updated with the performance reports, data backup and disaster recovery procedures for their cloud applications. There must always be symmetry between the control and the risk associated with the programs and operations [8].

The cloud access security brokers (CASB) are the main points through which security is enforced between the consumers and the service providers which generally have security controls to access the cloud services, normally the SaaS services. The CASBs may also have a control over the internal resources of the company. The main security controls are the authorization policy, authentication, prevention of intrusion, malware detectors and filters, security auditing and encryption [9].

Recent studies have shown that there are mainly three major vectors of attacks. They are network, hypervisor and hardware. They are mapped onto attacks which are of the nature of external, internal and the one who provides the cloud or an insider attack. The countermeasures and controls mainly include end to end encryption,

scanning for malicious activities, validation of cloud consumers, secure interfaces and APIs and business continuity plans [10]. Sometimes data is changed with real looking information's to make it more secured or to hide it from attackers; it is called data masking [11].

The charging model of the cloud provider is also immensely complex. The elastic resource pool which is accomplished by virtualization or multi-tenancy has made it more complex than normal data centers. Unlike the traditional data center, cloud providers cannot calculate their cost on the basis of consumption on the static computing [12].

3 Objectives

The CASB will look upon various different parameters, which mainly are:

1. Source port.
2. Destination port.
3. Protocol Name.
4. Device Id.
5. Event Category.
6. Relevance.
7. Credibility.
8. Severity.
9. Magnitude.
10. Event Count.
11. Time (Hours, Minutes, Seconds, am/pm).

To decide intelligently to grant access within the parameter or not, it will be powered by a neural network which can dynamically change its decision for the next requests (Fig. 1).

4 System Architecture

Our proposed system consists of two main modules- the cloud based broker module and the intelligence module. The intelligence module is made up of artificial neural networks for the categorization of the incoming cloud traffic into safe and malicious requests. The intelligence module consists of one input layer, five hidden layers and one output layer where in each layer the output is computed by the following formula in (1).

$$f = b + \sum_{i=0}^n x_i w_i \quad (1)$$

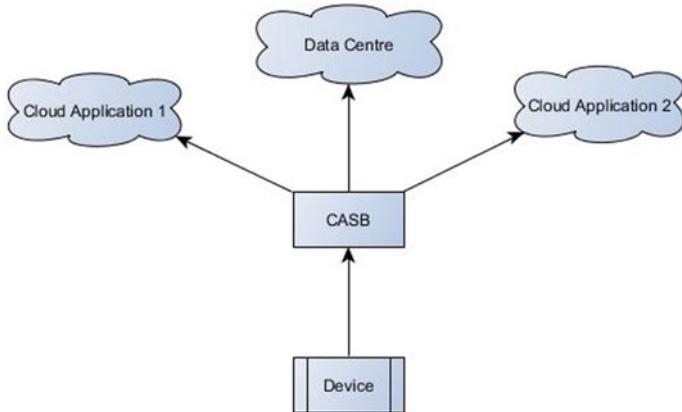


Fig. 1 The use of CASB in the system

Here f denotes the output of each layer, x denotes the input into each layer, w denotes the individual weights of each of the nodes present within the layer and b denotes the bias of each hidden layer.

The output layer gives 5 kinds of categories, namely

- Gateway Permit
- Gateway Deny
- Information Leak
- Access block
- Miscellaneous Exploit.

Softmax activation is used in the output layer due to which the output that is obtained is directly in the form of probabilities, sum of which is always equal to 1. In the rest of the layers, we use “RELU” activation as we observed that it was the most suitable for our numerical data. Other than this, in order to tackle the problems occurring due to over fitting, the concepts of weight regularization with a coefficient of 0.01 and a dropout rate of 0.3 is applied at each of the hidden layers. Finally while training our model; we applied Adam optimizer as it has a reasonably high convergence rate with low memory requirements.

5 Results and Discussions

For our experimental setup, we have taken the data from unrestricted access logs which consisted of around 1500 entries involving various conditions under which the rule based network security elements had to perform security controls of allowing or denying the incoming traffic. Based on this data, we were able to train our own intelligent gateway model in a supervised manner using feed-forward artificial neural

networks. As shown in the graph of Fig. 2, we were able to achieve a test accuracy of 97.32% and a validation accuracy of 86.26% which is at par with the latest state of the art cloud based security brokers. Another important point to be noted from the graph is that based on the network architecture, the accuracy increases significantly as the number of epoch increases, which shows that with increase in the amount of data collected by the continuous use of our dynamic cloud access security broker, the performance in terms of accuracy automatically increases with time. This is a major advantage of our proposed algorithm over the traditional rule based security gateways.

As shown in the graph of Fig. 3, the loss scores of training and validation of our model over increasing number of epochs is compared. The train and test loss scores remain almost equal, until the epoch value reaches around 500, where the train loss

Fig. 2 Graph for accuracy versus epoch

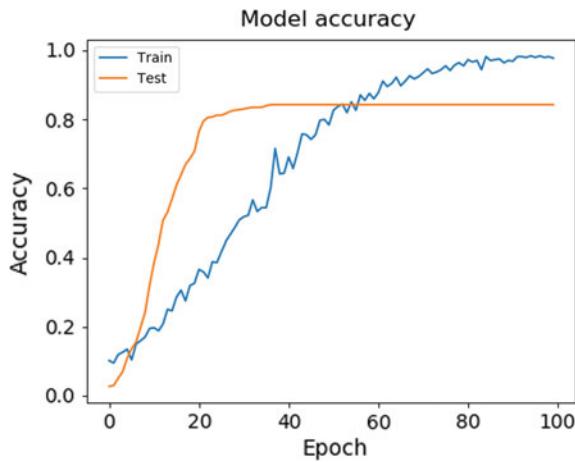
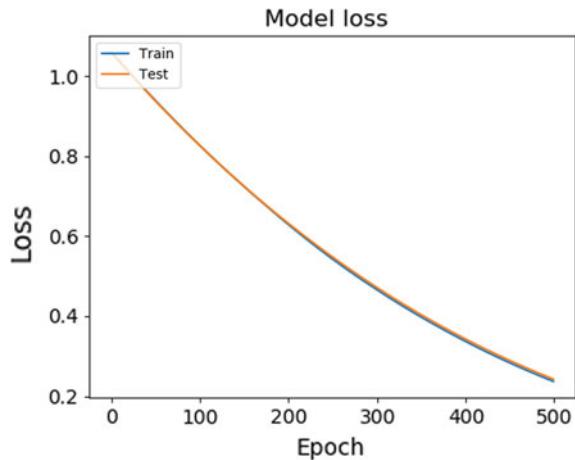


Fig. 3 Graph for loss versus epoch



becomes slightly lesser than the validation loss due to gradual over fitting of the network. Also, the general trend in both the cases is that the loss reduces as the epochs increase in an exponential fashion until ultimately the graph starts flattening out when the loss values start to become a comparatively small constant value. This is when our model reaches a stable state of minimum loss, and thus our model is able to achieve the best performance.

6 Conclusion

In the coming few years the use of cloud technology will increase tremendously and so will the security threat to it. Attackers nowadays no more use traditional attack methods to find loop holes but instead use machine learning and artificial intelligence to pose a security risk. Breaking any type of encryption is just a matter of time with the help of advance computers having massive processing powers. Thus our AI powered CASB technology will help in countering these types of threats. The main advantage of our proposed algorithm over other existing cloud security brokering protocols is that it is dynamic in nature and keeps on learning as and when more operational data is collected or available. This allows the security broker to continuously adapt by introducing new rules based on the recent trends observed in the transactional logs of the cloud system. Therefore, not only will it help in keeping the data in the cloud network safe but also it will help in the growing of businesses by mitigating all kind of risks. In future, our research work can further be extended by applying recurrent neural networks in our proposed system to improve the accuracy performances since RNNs can handle time dependencies better. But on the other side, RNNs are also computationally more expensive and hence it results in a tradeoff between accuracy and hardware processing power which is not suitable in all scenarios.

References

1. G. Gupta, P.R. Laxmi, S. Sharma, A survey on cloud security issues and techniques. *Int. J. Comput. Sci. Appl.* **4**(1), (2014)
2. C. Liu G. Wang, P. Han, H. Pan, B. Fang, A cloud access security broker based approach for encrypted data search and sharing, in *International Conference on Computing, Networking and Communications*, (IEEE, 2017), pp. 422–426
3. A.H. Bhat, S. Patra, D. Jena, Machine learning approach for intrusion detection on cloud virtual machines. *Int. J. Appl. Innov. Eng. Manage.* **2**(6), 56–66 (2013)
4. D. Bhambhani, T. Salman, M. Samaka, A. Erbad, R. Jain. Feasibility of supervised machine learning for cloud security. *Int. Conf. Inf. Sci. Secur. (ICISS)* 1–5 (2016). IEEE
5. R.V. Rao, K. Selvamani, Data security challenges and its solutions in cloud computing. *Procedia Comput. Sci.* 204–209 (2015)
6. A.W. Varsha, S. Gupta, Study of security issues in cloud computing. *Int. J. Comput. Sci. Mob. Comput.* **4**(6), 230–234 (2015)
7. S. Ezzati, H.R. Naji, A. Chegini, P. Habibimehr, Intelligent firewall on reconfigurable hardware. *Eur. J. Sci. Res.* **47**(4), 509–516 (2018)

8. V. Singh, S.K. Pandey, Research in cloud security: problems and prospects. *Int. J. Comput. Sci. Eng. Inf. Technol. Res.* **3**(3), 305–314 (2013)
9. E. Fernandez, N. Yoshioka, H. Washizaki, Cloud access security broker (CASB): a pattern for secure access to cloud services, in *Asian Conference on Pattern Languages of Programs, Asian PLoP*, vol. 15 (2015)
10. G. Ramachandra, M. Iftikhar, F.A. Khan, A comprehensive survey on security in cloud computing, in *The 3rd International Workshop on Cyber Security and Digital Investigation* (2017), pp. 1877–0509
11. J. Kumar, Cloud computing security issues and its challenges: a comprehensive research. *Int. J. Recent Technol. Eng.* **8**, 2277–3878 (2019)
12. S. Kumar, R.H. Goudar, Cloud computing—research issues, challenges, architecture, platforms and applications: a survey. *Int. J. Future Comput. Commun.* **1**(4), 356–360 (2012)

A Comparative VHDL Implementation of Advanced Encryption Standard Algorithm on FPGA



Darshit Suratwala and Ganesh Rahate

Abstract The advanced encryption standard (AES) offers more flexibility in terms of implementing it in software and on hardware. Since field programmable gate arrays (FPGA's) provide excellent performance in terms of parallel processing and simpler design cycles. This research intends to examine the required setup time, delay from clock to external outputs, logic and route utilization at input and output stage. The performance of AES algorithm is evaluated by simulating this symmetric algorithm on two different FPGAs using hardware description language (HDL). Xilinx ISE Design Suite is the software which helps in simulating, optimizing a synthesizable HDL code. Xilinx Spartan Family and Virtex Family devices are used for software evaluation, and the results of Xilinx Virtex are superior and efficient because of less delays.

Keywords AES · FPGA · Rijndael · Cryptography · Symmetric algorithm · Encryption · Decryption · Cypher · Decipher · VHDL

1 Introduction

Today, there is an emergence of secure data transmission network, which is the mainstream requirement for secure communication. In addition, hacking attempts have made the need for secure communication over the networks more important. This leads to encrypt the data and decrypt it using various encryption and decryption algorithms. Cryptography is a method to make the data appear random when transferred over an unsecured channel. It protects the confidential data from any unauthorized person to read or misuse it by altering the data.

D. Suratwala (✉) · G. Rahate

Electronics and Telecommunication Engineering Department, Pimpri Chinchwad College of Engineering, Pune, India

e-mail: darshit.ps@gmail.com

G. Rahate

e-mail: ganesh.rahate@gmail.com

In the year 2001, an organization called National Institute of Standards and Technology published an algorithm for cypher, called as advanced encryption standard (AES), also originally called Rijndael algorithm [1]. It falls under the category of symmetric or private-key algorithm where the sender and receiver cipher the data using a single private key. The AES algorithm is better in terms of securing the private networks from any possible hacks. But for any algorithm to perform, timing parameters are considered important. Therefore verifying the time-based constraints of AES on different FPGA platforms is necessary. Hence, it is used worldwide, and this made the data encryption standard (DES) obsolete. AES is used by all the sectors such as Ecommerce websites, online banking, and to prevent Cyber security crimes. In recent times, FPGAs have evolved and developed at an unprecedented rates. The main reason behind this is decrease in size of transistors and high clock speeds.

In this paper, the idea is to analyze and evaluate the published AES algorithm by NIST and implement its VHDL substitute on two different FPGAs using 128 bits of data blocks referred as “AES-128”. There is a noticeable difference observed in the timing summary at a common speed grade, hence lesser the time the faster the communication to achieve performance at high speeds. This will help adapting AES in various sectors such as banking, automotive, and home automation for enhanced security. AES has performed excellently well in terms of speed, security, and reliability.

The paper is divided into following sections as follows: In Sect. 2, types of ciphers along with AES are explained. In Sect. 3, the timing analysis of AES algorithm is on two different FPGAs. In Sect. 4, concluding remarks are presented.

2 Literature Survey

The literature survey is carried out to understand the types of ciphers, classification of cryptography, AES, and its features.

2.1 Types of Cypher

Cryptographic algorithms are categorized in two parts: Asymmetric and symmetric. These algorithms use different keys to cipher the encrypted data and decrypt it. Further, the algorithms are described in detail manner.

1. Asymmetric or Public-key Algorithm: In this algorithm, two keys are required, public as well as private. To encrypt, public-key is used and to decrypt, private-key is used. While communicating between a sender and receiver, the sender makes use of public key to encipher the data in such a way that receiver deciphers it using same key or private one. But the drawback of public key algorithms is that it uses more memory, and this results in slower execution speeds [2].

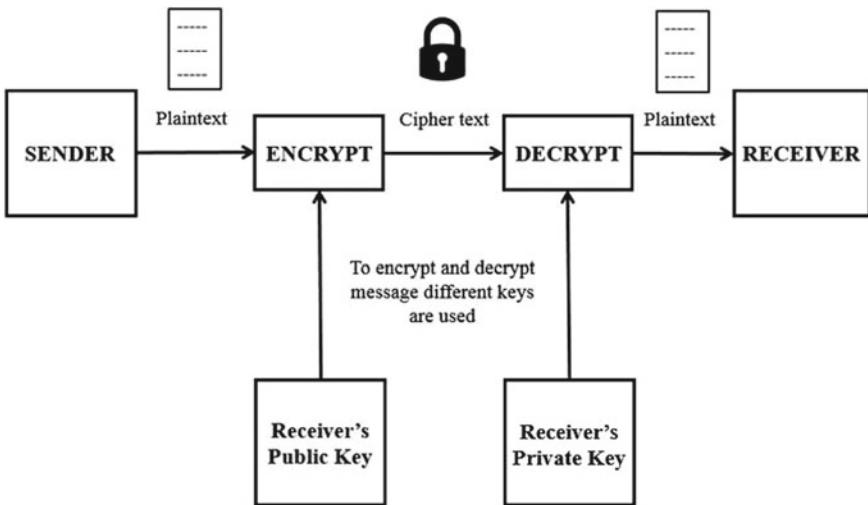


Fig. 1 Asymmetric algorithm for transmitting data

The above figure reciprocates asymmetric key algorithm that uses public and private keys for securing the message over a channel.

2. Symmetric or Private-Key Algorithm: In symmetric algorithm, the data is secured over a channel using a personal (private) key. Sender first sends the personal key to receiver and then the same key encrypts the data sent by the sender and decrypts the data at the receiver. Such algorithms require fewer memory and run faster compared to public-key algorithms. The promising symmetric cyphers are block ciphers which are AES and DES. Stream encoders typically are most well-liked in embedded domain because of being more efficient in terms of speed, size, and simplicity but not as safe as block ciphers [2]. Typical cases of stream cyphers are RC4, Salsa20 and Scream, etc. In private-key-enabled cryptosystems, the cipher data is sent on the channel even when there is a chance of a hack, but the key is not sent on the network with data; therefore, the possibility of data being deciphered is nil. The receiver gets authenticated only with the help of a password.

2.2 AES Algorithm

It is a type of symmetric algorithm. The encryption methodology of this algorithm states conversion of data to be sent into random sequence referred to as cypher-text. Now, converting the cypher-text into its original format is called decryption process in AES. There are three private keys which are 128, 192, and 256 bits long, and used to encrypt and decrypt the data by the AES [3].

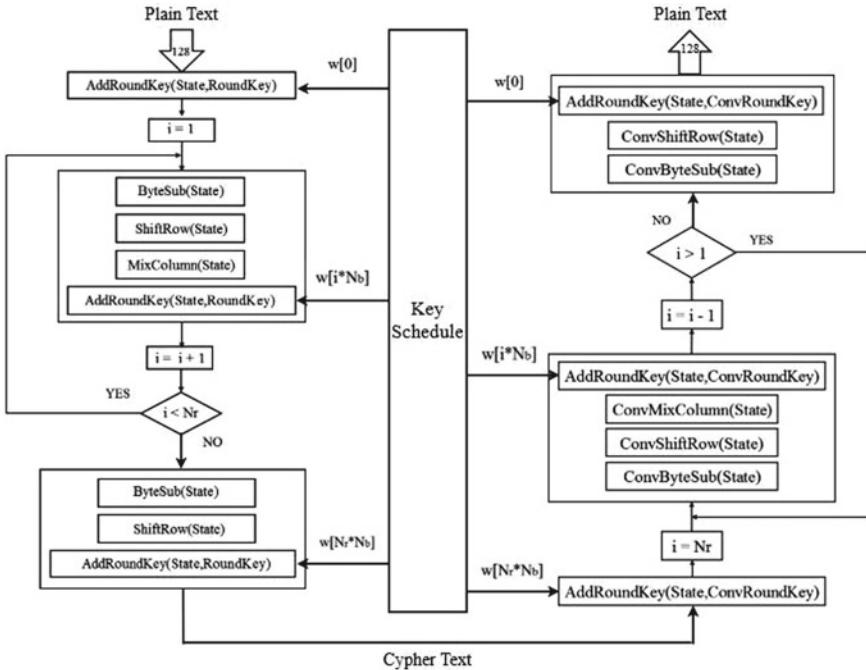


Fig. 2 Encryption and decryption process

Encryption Process: The process is made of four different transformations, which is applied for fixed iterations, also known as rounds, to the data that is to be encrypted. The key size will ensure the number of iterations or rounds, as shown in Fig. 2 [3].

1. **Bytes Substitution (ByteSub(State)) Transformation:** It is an irregular substitution of bytes that functions freely on every byte with the help of substitution table.
2. **Shift Rows (ShiftRow(State)) Transformation:** The final three rows of State, the bytes are shifted in cycles over various number of bytes. But the $r = 0$ (primary row) is not shifted.
3. **Mixing of Columns (MixColumn(State)) Transformation:** Galois field multiplication is base of this transformation. In a column, there are several bytes and each byte takes the place of another value that is a task of all 4 bytes of a given column. The Mix Columns() conversion works on the State column-by-column, making each column a 4 term polynomial.
4. **Addition of Round Key (AddRoundKey(State, RoundKey)) Transformation:** Bitwise XOR operation is used in adding round key to State. Key schedule generation block consists of Round key with Nb words.

Key Schedule Generation: Key of every round is made up of 128 bit array; it is formed as the product preceding key round, each round is changed by a constant and

Table 1 Key block

| AES type | Key length (Nk words) | Number of rounds (Nr) |
|----------|-----------------------|-----------------------|
| 128 | 4 | 10 |
| 192 | 6 | 12 |
| 256 | 8 | 14 |

series of Substitution-Box lookups for every key of 32 bit word. Original input by the user is the first key round. Each AES type has different sizes of key and different iterations, as presented in Table 1.

AES needs a primary set of Nb words (4) which is common for all types, and each of the Nr rounds requires Nb words of key data. The four-byte words are in linear array, called resulting key schedule.

Decryption Process: This method is a converse of encoding method. Transformations used in encoding method are reciprocally put into the present method. Therefore, in last round, values of information and key are nothing but the primary round inputs for decoding method and follow in reducing order as shown in Fig. 2 [3].

1. **Converse Byte Substitution (InvByteSub(State)) Transformation:** It is the converse of first step in encryption process, in which the converse Substitution-Box is pertain to every byte of the State.
2. **Converse Shift Rows (InvShiftRow(State)) Transformation:** It is the converse of the second step of encryption. The bytes present in last 3 rows of the State are shifted clockwise over different bytes, but $r = 0$, is not shifted. Row number is responsible for shifting.
3. **Converse Mixing of Columns (InvMixColumn(State)) Transformation:** It is the inverse of MixColumns. By operating on the State columns-by-columns, it makes every column a four-term polynomial.

Features of AES: The following features of AES makes it more affordable, reliable, and mainly secure algorithm.

1. Single cypher and decipher key.
2. Impossible to read the encrypted message through Brute force hacking because of 128-, 192-, and 256-bit key lengths.
3. Easily implementable on software and hardware.
4. Cost-effective because of no royalties.

3 Implementation and Results

VHDL is a hardware description language for very high speed integrated circuits, which is used due to its flexibilities to exchange among different environments. Implementable over other devices [4]. Work is based on Xilinx ISE Design Suite 14.7. It is used for checking the performance of the AES algorithm's results using tools available on ISE Design Suite. The cypher and decipher processes are simulated

Timing Summary:**Speed Grade: -3**

Minimum period: No path found
 Minimum input arrival time before clock: 3.475ns
 Maximum output required time after clock: 3.597ns
 Maximum combinational path delay: No path found

Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
 Total number of paths / destination ports: 224 / 16

Offset: 3.475ns (Levels of Logic = 4)
 Source: state<5> (PAD)
 Destination: b_0 (FF)
 Destination Clock: clk rising

Data Path: state<5> to b_0

| Cell:in->out | fanout | Gate Delay | Net Delay | Logical Name (Net Name) |
|--------------|--------|------------|-----------|---|
| IBUF:I->O | 32 | 1.222 | 1.656 | state_5_IBUF (state_5_IBUF) |
| LUT6:I0->O | 1 | 0.203 | 0.000 | Mram_state[7].GND_6_o_wide_mux_0_OUT1 (Mram_state[7].GND_6_o_wide_mux_0_OUT) |
| MUXF7:I1->O | 1 | 0.140 | 0.000 | Mram_state[7].GND_6_o_wide_mux_0_OUT_f7 (Mram_state[7].GND_6_o_wide_mux_0_OUT_f7) |
| MUXF8:I1->O | 1 | 0.152 | 0.000 | Mram_state[7].GND_6_o_wide_mux_0_OUT_f8 (state[7].GND_6_o_wide_mux_0_OUT<0>) |
| FDC:D | | 0.102 | | b_0 |

Total 3.475ns (1.819ns logic, 1.656ns route)
 (52.3% logic, 47.7% route)

Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
 Total number of paths / destination ports: 8 / 8

Offset: 3.597ns (Levels of Logic = 1)
 Source: b_7 (FF)
 Destination: b<7> (PAD)
 Source Clock: clk rising

Data Path: b_7 to b<7>

| Cell:in->out | fanout | Gate Delay | Net Delay | Logical Name (Net Name) |
|--------------|--------|------------|-----------|-------------------------|
| FDC:C->Q | 1 | 0.447 | 0.579 | b_7 (b_7) |
| OBUF:I->O | | 2.571 | | b_7_OBUF (b<7>) |

Total 3.597ns (3.018ns logic, 0.579ns route)
 (83.9% logic, 16.1% route)

Fig. 3 Timing summary of encryption and decryption using Spartan 6

using FPGA Xilinx family Spartan 6 and Virtex 7 devices. The synthesis is done at speed grade of -3.

3.1 Timing Summary for Xilinx Spartan 6

The implementation of VHDL code used for encryption and decryption on the Xilinx Spartan 6 FPGA using ISE design suite shows the timing analysis in Fig. 3. Before clock, minimum input arrival time obtained is 3.475 ns. It is also called the required

Timing Summary:**Speed Grade: -3**

Minimum period: No path found
 Minimum input arrival time before clock: 0.995ns
 Maximum output required time after clock: 0.511ns
 Maximum combinational path delay: No path found

Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
 Total number of paths / destination ports: 224 / 16

| Offset: | 0.995ns (Levels of Logic = 4) |
|--------------------|-------------------------------|
| Source: | state<5> (PAD) |
| Destination: | b_0 (FF) |
| Destination Clock: | clk rising |

Data Path: state<5> to b_0

| Cell:in->out | fanout | Gate Delay | Net Delay | Logical Name (Net Name) |
|--------------|--------|------------|-----------|---|
| IBUF:I->O | 32 | 0.000 | 0.657 | state_5_IBUF (state_5_IBUF) |
| LUT6:I0->O | 1 | 0.043 | 0.000 | Mram_state[7]_GND_6_o_wide_mux_0_OUT3 (Mram_state[7]_GND_6_o_wide_mux_0_OUT2) |
| MUXF7:I1->O | 1 | 0.172 | 0.000 | Mram_state[7]_GND_6_o_wide_mux_0_OUT_f7_0 (Mram_state[7]_GND_6_o_wide_mux_0_OUT_f7_1) |
| MUXF8:I0->O | 1 | 0.123 | 0.000 | Mram_state[7]_GND_6_o_wide_mux_0_OUT_f8_0 (state[7]_GND_6_o_wide_mux_0_OUT<0>) |
| FDC:D | | -0.001 | | b_0 |

Total 0.995ns (0.338ns logic, 0.657ns route)
 (34.0% logic, 66.0% route)

Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
 Total number of paths / destination ports: 8 / 8

| Offset: | 0.511ns (Levels of Logic = 1) |
|---------------|-------------------------------|
| Source: | b_7 (FF) |
| Destination: | b<7> (PAD) |
| Source Clock: | clk rising |

Data Path: b_7 to b<7>

| Cell:in->out | fanout | Gate Delay | Net Delay | Logical Name (Net Name) |
|--------------|--------|------------|-----------|-------------------------|
| FDC:C->Q | 1 | 0.232 | 0.279 | b_7 (b_7) |
| OBUF:I->O | | 0.000 | | b_7_OBUF (b<7>) |

Total 0.511ns (0.232ns logic, 0.279ns route)
 (45.4% logic, 54.6% route)

Fig. 4 Timing summary of encryption and decryption using Virtex 7

setup time. After clock, maximum output arrival time obtained is 3.597 ns. It is also called as delay from clock to external outputs. The total utilization of logic is 52.3% in 1.819 ns and 47.7% route in 1.656 ns at the input stage. The total utilization of logic is 83.9% in 3.018 ns and 16.1% route in 0.579 ns at the output stage.

3.2 Timing Summary for Xilinx Virtex 7

The implementation of VHDL code used for encryption and decryption on the Xilinx Virtex 7 FPGA using ISE design suite shows the timing analysis in Fig. 4. Before

Table 2 Difference in timing results due to change in RTL

| Parameters | Xilinx Spartan 6 | Xilinx Virtex 7 |
|--------------------------------------|-------------------|-------------------|
| Required setup time | 3.475 ns | 0.995 ns |
| Delay from clock to external outputs | 3.597 ns | 0.511 ns |
| Logic utilization at input stage | 52.3% in 1.819 ns | 34% in 0.338 ns |
| Logic utilization at output stage | 83.9% in 3.018 ns | 45.4% in 0.232 ns |
| Route utilization at input stage | 47.7% in 1.656 ns | 66.0% in 0.657 ns |
| Route utilization at output stage | 16.1% in 0.579 ns | 54.6% in 0.279 ns |

clock, minimum input arrival time obtained is 0.995 ns. It is also called the required setup time. After clock, maximum output arrival time obtained is 0.511 ns. It is also called as delay from clock to external outputs. The total utilization of logic is 34% in 0.338 ns and 66.0% route in 0.657 ns at the input stage. The total utilization of logic is 45.4% in 0.232 ns and 54.6% route in 0.279 ns at the output stage.

The summarized results in Table 2 shows that the change in RTL, i.e., moving from Xilinx Spartan 6 toward Xilinx Virtex 7 for refined and denser technology, the improvement in the delay is seen clearly. Therefore, AES will so be enforced with affordable efficiency on a Virtex 7 FPGA.

4 Conclusion

Compared to other algorithms, AES provides full-proof security because of the randomness. The permutation combinations of AES output are very much indistinguishable. The mathematics behind the algorithm makes the algorithm more sound and secure. Depending upon the number of rounds on the software implementations, for example, in this case, the implementation comprises of using one key block size combination (128-128) only for round 1 analysis. The only purpose of software implementation over here is to know the difference in their timing summaries. The change in RTL has clearly shown the timing delays between the Xilinx Spartan 6 and Xilinx Virtex 7 at speed grade of -3. The inclusion of setup time, delay from clocks to external outputs by the Xilinx Virtex 7 to implement AES algorithm is lesser comparative to Xilinx Spartan 6 as well as lesser logic utilization and more routing capability in lesser time have shown the optimized efficiency of Xilinx Virtex 7.

References

1. FIPS 197, Advanced Encryption Standard (AES), 26 Nov 2001. <https://csrc.nist.gov/publications/fips/fips197/fips>
2. A.K. Jadoon, L. Wang, T. Li, M. A. Zia, Lightweight cryptographic techniques for automotive cybersecurity. Hindawi Wirel. Commun. Mobile Comput. (2018). (Article ID 1640167)
3. A VHDL Implementation of the Advanced Encryption Standard-Rijndael Algorithm by Rajender Manteena. A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering Department of Electrical Engineering. College of Engineering University of South Florida. Date of Approval: 23 Mar 2004
4. R.N. Sklavos, O. Koufopavlou, Architectures and VLSI implementations of the AES-proposal. IEEE Trans. Comput. **51**(12) (2002)

Document Recommendation for Medical Training Using Learning to Rank



Raghvendra Rao, Suyash Choubey, Georg Gutjahr, and Prema Nedungadi

Abstract This paper describes a general approach where Learning-to-Rank (Ltr) algorithms are used to suggest training materials, such as videos, simulations, tutorials, and articles to students based on their mistakes while using an e-learning platform. In particular, the method is discussed and implemented for MedSim, computer-based medical simulations that replicate clinical scenarios for medical students. While interacting with a virtual patient, the student may ask questions, do medical examinations or therapy, and eventually arrive at a final diagnosis. The basic idea of the approach is to generate a query based on the mistakes that a student makes while interacting with a virtual patient. Based on this query, the Ltr algorithm then provides the student with suggestions for training material to review. We evaluated the approach with different Ltr algorithms and different queries and dataset sizes (Haridas et al. in Educ. Inf. Technol. 1–19, 2020 [1]). We studied the extensibility of the trained models by adding new training material to the test set. The accuracy is measured using NDCG@10. The results of the evaluation suggest that a training size of around 1000 is sufficient to obtain accurate results. The comparison of algorithms suggests that listwise algorithms perform better than pair-wise algorithms, also that in particular, LambdaMart performs very well on this problem.

Keywords Medical case search · Clinical decision support · Learning to Rank

R. Rao (✉) · S. Choubey · P. Nedungadi

Department of Computer Science, Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India
e-mail: raghavendarao234@gmail.com

S. Choubey
e-mail: choubeys.930@gmail.com

P. Nedungadi
e-mail: prema@am.amrita.edu

G. Gutjahr
AmritaCREATE, Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India
e-mail: georg.gutjahr@gmail.com

1 Introduction

Medical diagnosis is a very complicated task, and it is not easy to learn since each patient case is unique [2]. To learn the right way to diagnose a patient takes a lot of practice; even the doctors with years of experience may sometimes misdiagnose a patient.

MedSim.in [3, 4] is a website which provides articles and simulations of actual medical procedures which the medical students can use to learn the art of diagnosis using a virtual environment [5, 6].

This article considers the problem of suggesting relevant learning material to students based on the mistakes they made during their use of MedSim. The basic idea of the approach is to define a set of relevant questions (Sect. 4.1) to be asked to a virtual patient while diagnosing her and a relevance score between 0 and 3 for a few articles from MedSim. Based on which relevant questions get missed by the student in the simulation, a query (Sect. 4.2) is generated and all the relevant documents are recommended using Ltr algorithms (Sect. 2.2) [7, 8] so that the student can learn about the cases related to the missed questions.

1.1 MedSim

MedSim [3, 4] is an e-learning platform which reproduces real-life medical scenarios using computer-based simulations. It has data of virtual patients each representing a unique clinical scenario. A medical student can practice by diagnosing these virtual patients and gain clinical experience [5, 6].

Learning material in MedSim includes videos, animations, articles, and tutorial on topics such as blood pressure, body temperature and blood sugar measurement, and infectious diseases. In the following, the term *document* will be used to refer to any such material, as is commonly done in the information retrieval literature [9].

1.2 Paper Outline

The outline of this paper is as follows. Section 2 includes Learning to Rank (Ltr), its various types of algorithms, and their classifications. Section 3 includes related works and differences in our approach. Section 4 describes our approach to the problem and provide a methodology for applying Ltr in clinical decision making. Section 5 includes the evaluation of the proposed approach, its results, and inferences. Section 6 provides concluding remarks and summarizes our paper.

2 Learning to Rank

Learning to Rank (Ltr) [7, 10] is a field of machine learning for creating ranking models for information retrieval systems [9]. The training data is a lists of ranked documents. Information about this ranking is given by a partial order. There are multiple ways to specify this partial order. For example, a score can be defined by giving numeric values to the documents, where higher values indicate a more relevant document. Another possible is to have expert judgment can be used that defines a binary relation ('this document seems more relevant than that document'). The ranking model should sort documents in a list such that it 'resembles' in some way to the training data. The Ltr approach can be supervised or semi-supervised. Methods based on reinforcement learning are also possible.

2.1 Classification of Ltr Algorithms

The Ltr algorithms are classified into three categories based on the approach used, **Pointwise** (Example: SLR [11]), **Pairwise** [7] (Example: MART [12], RankBoost [8], RankNet [12]) and **Listwise Approach** [7] (Example: ListNet [12], LambdaMART [12]).

2.2 Ltr Algorithms

RankNet [12], **LambdaRank**, **LambdaMART** [13], **MART** [12], **RankBoost** [8], **Coordinate Ascent** [14], **ListNet** [7] and **Random Forest** [15] are the Ltr algorithms used in this paper.

More details on Ltr can be found in the book 'Learning to Rank for Information Retrieval' by Liu [9].

3 Related Work

Much of the previous work on Ltr was on new Ltr suitable algorithms (e.g., LambdaMart [13] and coordinate ascent [14]), on finding practical search scenarios for these algorithms by adding new feature sets, and on evaluating the performance on various search tasks and algorithms [16, 17]. Note that for some domains like medical and clinical applications, Ltr is applied based on the medical features of the patient only [18–20].

On the other hand, in the present work, the features are the mistakes of a medical student. Our work aims to apply LtR to search tasks performed by a medical practitioner for training himself/herself to learn about relevant questions to ask a patient based on its condition. Therefore, we use the MedSim [3] dataset, which provides virtual patients and relevant questions to ask. Based on the relevant questions missed by the student, we recommend related learning materials and documents to study.

4 Methodology

We created our model for a single virtual patient, and later, this could be extended for any number of virtual patients. Based on the relevant questions (Sect. 4.1) missed during the evaluation of the virtual patient by the medical student, a query (Sect. 4.2) is created. For each document from MedSim (1.1), the query created and the relevance of each question for the document in the case of the virtual patient is used to create a feature vector (Sect. 4.3) for that query-document pair. The feature vectors created for the documents will be used to rank them for the given query using the trained LtR models.

4.1 Relevant Questions

We assume a set of medical diagnosis questions as standard questions; i.e., any patient examination will mostly have a subset of N relevant questions out of these questions. In this experiment, the virtual patient is having 16 relevant questions (i.e., $N = 16$) which means the diagnosis of this virtual patient must contain these N questions.

4.2 Query

Based on the relevant questions (Sect. 4.1) missed by the medical student while the diagnosing of the virtual patient, a query will be created. This query will later be used for ranking the medical documents (Sect. 1.1) for the medical student to study from so that she could have a better understanding of the given case.

In our experiment as there are 16 relevant questions, there are 2^{16} possible queries based on whether a question is missed or not. As the number of possible queries is really large, we need our model to be able to be trained using a small set of these queries and still give significant results.

4.3 Feature Vector

For each document-query pair, a feature vector is to be generated. The feature vector in the proposed model has $2 \times N = 32$ binary features. The first N binary features are query-based features; i.e., they will represent which of the N relevant questions are missed while diagnosing. The last N binary features are static features, the i th feature is assigned ‘1’ if the document is relevant when the $(i - N)$ th relevant question is missed by the student and assigned ‘0’ if the document is irrelevant.

4.4 Relevance Score

Each document-query pair in the training dataset should have a relevance score. The score is assigned using the presumed relevance score of each document for each question. The relevance score is between 0 and 3. We assign 0 for least relevant documents and 3 for highly relevant documents. The relevance score of a query-document pair is computed by taking an average of all the scores of the document for each of the missed questions in the query.

4.5 Dataset

We select some random queries (Sect. 4.2) and a set of random documents (Sect. 1.1) corresponding each query for generating our dataset. Using the relevance score (Sect. 4.4) and feature vector (Sect. 4.3) for all document-query pair, a dataset is created. The format of the dataset for a document-query pair with relevance score ‘R’ and query ID ‘Q’ and value of first feature as ‘f1’, second feature ‘f2’ and so on, has the following form:

```
R qid:Q 1:f1 2:f2 3:f3 ...
```

This is based on the standard format acceptable by the RankLib [21]. An example with 3 relevant questions, where questions 1 and 3 are missed and the document is relevant for question 2 and 3, and where the relevance score is 2, would look like the following:

```
2 qid:1 1:1 2:0 3:1 4:0 5:1 6:1
```

The dataset created will be used to train models using different LtR algorithms (Sect. 2.2) and will compare the performance of each trained model.

5 Evaluation

5.1 Experimental Setup

We generate datasets as described in Sect. 4.5 of sizes 100, 1000, and 10,000, respectively, using 12 medical documents (Sect. 1.1) and 16 relevant questions (Sect. 4.1). We use these datasets to train models using different LtR algorithms (Sect. 2.2).

We create our first test dataset of size 3000 which will be used to test the performance of each trained model. We then compared the performance of each algorithm and the effect of training dataset size.

Our model should be extensible, which means adding new documents to rank should not require retraining the entire model. We test the extensibility of our model by creating another test dataset (second dataset) by adding using 10 new documents (22 documents in total) of size 3000 and again compared the performance of each model.

5.2 Evaluation Measure—NDCG@10

Normalized discounted cumulative gain (NDCG) [22] is a popular method of measuring the ranking quality. When a highly relevant document appears lower in the ranked result, the relevance quality gets degraded and should be penalized. The DCG score is then calculated by summing over decreasing relevance values of the documents, where these values are decreased according to according to the positions of the documents in the ordered list.

In more detail, the DCG score for ranking n documents, where relevance score of the i th document is r_i , is defined as

$$\text{DCG}@n = \sum_{i=1}^n \frac{r_i}{\ln(i+1)}. \quad (1)$$

NDCG@n refers that only the top n -ranked documents are used for calculating the score. In this experiment, we are using n as 10.

Comparing the ranking performance from one query to the next cannot be consistently achieved using DCG alone. We might find that some queries are harder than others and might produce lower DCG scores. This problem can be solved by scaling the results based on the best result seen, that is calculating the normalized DCG:

$$\text{NDCG}@n = \frac{\text{DCG}@n}{\max \text{DCG}@n}, \quad (2)$$

where the maximum in the denominator is taken over all possible rankings.

5.3 Results

When No New Document Is Used While Testing: Table 1 shows the performance of different LtR algorithms when trained on datasets of different sizes while ranking the medical documents. Tables 1 and 2 show the NDCG@10 value after the ranking, the improvement in NDCG@10 score above the baseline of unordered documents in test set.

In this case, the first test dataset (Sect. 5.1) is used for evaluation, which means no new document is introduced in the test set. Hence, the models are tested on the same documents they were trained with.

Table 1 Performance of LtR algorithms when no new document is introduced

| LtR algorithms | Dataset size | | | | | |
|------------------------------------|--------------|--------------|----------|--------------|----------|--------------|
| | 100 | | 1000 | | 10,000 | |
| | NDCG@ 10 | Improvement | NDCG@ 10 | Improvement | NDCG@ 10 | Improvement |
| Multiple Additive Regression Trees | 0.88 | +0.21 (+32%) | 0.91 | +0.24 (+36%) | 0.91 | +0.24 (+36%) |
| Rank Net | 0.82 | +0.15 (+23%) | 0.85 | +0.18 (+27%) | 0.85 | +0.18 (+27%) |
| Rank Boost | 0.84 | +0.18 (+26%) | 0.85 | +0.18 (+27%) | 0.85 | +0.18 (+27%) |
| Coordinate Ascent | 0.84 | +0.18 (+26%) | 0.85 | +0.18 (+27%) | 0.85 | +0.19 (+28%) |
| Lambda Mart | 0.85 | +0.19 (+28%) | 0.91 | +0.24 (+36%) | 0.91 | +0.25 (+37%) |
| List Net | 0.84 | +0.17 (+25%) | 0.85 | +0.18 (+27%) | 0.85 | +0.18 (+27%) |
| Random Forests | 0.88 | +0.21 (+32%) | 0.89 | +0.22 (+33%) | 0.89 | +0.22 (+33%) |

Table 2 Performance of LtR algorithms on ranking new documents

| LtR algorithms | Dataset size | | | | | |
|------------------------------------|--------------|--------------|----------|--------------|----------|--------------|
| | 100 | | 1000 | | 10,000 | |
| | NDCG@ 10 | Improvement | NDCG@ 10 | Improvement | NDCG@ 10 | Improvement |
| Multiple Additive Regression Trees | 0.85 | +0.13 (+17%) | 0.86 | +0.14 (+18%) | 0.86 | +0.14 (+18%) |
| Rank Net | 0.82 | +0.12 (+16%) | 0.84 | +0.13 (+18%) | 0.84 | +0.13 (+18%) |
| Rank Boost | 0.84 | +0.12 (+17%) | 0.85 | +0.13 (+18%) | 0.85 | +0.13 (+18%) |
| Coordinate Ascent | 0.84 | +0.12 (+17%) | 0.85 | +0.13 (+18%) | 0.85 | +0.13 (+18%) |
| Lambda Mart | 0.85 | +0.13 (+18%) | 0.85 | +0.13 (+18%) | 0.85 | +0.13 (+18%) |
| List Net | 0.84 | +0.13 (+17%) | 0.85 | +0.13 (+18%) | 0.85 | +0.13 (+18%) |
| Random Forests | 0.86 | +0.13 (+18%) | 0.87 | +0.14 (+19%) | 0.87 | +0.14 (+19%) |

After New Documents Are Introduced in the Test Set: Table 2 shows the performance of different LtR algorithms when trained on datasets of different sizes while ranking the medical documents. In this case, the second test dataset is used for evaluation, which means 10 new documents are introduced in the test set.

6 Discussion

We presented a model for applying LtR techniques to recommend training material to medical students. The results show that the model is capable of producing an improvement up to 36% above baseline of unordered documents with a NDCG@10 score of 0.91.

From the results, we can draw some conclusion on the performance of the LtR algorithms. In all the algorithms, there is not much improvement on changing the training set size from 100 to 10,000 as seen in changing the training set size from 100 to 1000. As discussed in above (Sect. 4.2), there are 2^n possible queries. A training size of about 1000 is enough for satisfying all the queries.

From Table 1, we can observe that MART gives NDCG@10 score of 0.88 when trained on a small training set. This score subtly increases to 0.91 on increasing the training set from 100 to 10,000. Whereas, LambdaMART shows considerable improvement in NDCG@10 score on increasing the training set. Its NDCG@10 increases from 0.84 to 0.91.

Pairwise approach works well when the model is trained on small train sets with no new documents used in the test (first test set). Whereas when we increase the train set size or include new documents in the test set (second test set), listwise approach outperforms the pairwise approach algorithms. Hence, listwise approach algorithms will be a better choice for applying this methodology.

7 Conclusion and Future Work

We presented a model for applying LtR techniques to evaluate not only the correctness of diagnosis, but to also recommend essential documents in ranked order to learn. The results show that our model is capable of producing an improvement up to 37% above the baseline. Future works include applying similar model to other applications like clinical searches, general health, etc. The technique can be generalized to improve ranking documents and search results for those tasks.

Future works include automatically generating the features by using information retrieval techniques on the medical documents rather than doing manual entry as done here. Instead of just using missed questions explicitly, NLP methods can be used to classify the asked question into the category of any of the relevant questions making the model more practical. Right now a small number of documents (13 and 23) are

used while training and testing the models. In future, new documents could be added to the test set to check the extensibility of the model to a further extent.

While the approach is used for a medical learning platform, it could equally well be applied to other types of pedagogical software applications like teaching languages or mathematics to provide a more effective, personalized learning experience.

References

1. M. Haridas, G. Gutjahr, R. Raman, R. Ramaraju, P. Nedungadi, Predicting school performance and early risk of failure from an intelligent tutoring system. *Educ. Inf. Technol.* 1–19 (2020)
2. R. Schilling, Every Patient Is Unique. <http://www.askdray.com/every-patient-is-unique/>
3. P. Nedungadi, R. Raman, The medical virtual patient simulator (MedVPS) platform, in *Intelligent Systems Technologies and Applications. Advances in Intelligent Systems and Computing*, vol. 384, eds. by S. Berretti, S. Thampi, P. Srivastava (Springer, Cham, 2016). https://doi.org/10.1007/978-3-319-23036-8_5. https://link.springer.com/chapter/10.1007%2F978-3-319-23036-8_5
4. P. Nedungadi, R. Shine, R.G. Raghu Raman, The medical virtual patient simulator (MedSIM)—an initiative under eHealth Digital India. *CSI CommunICatIonS* (2012)
5. H.A. Demba, P. Nedungadi, R. Raman, Olabs of digital India, its adaptation for schools in côte d’ivoire, West Africa, in *Information and Communication Technology for Intelligent Systems*, ed. by S.C. Satapathy, A. Joshi (Springer, Singapore, 2019), pp. 351–361
6. P. Chandrashekhar, M. Prabhakaran, G. Gutjahr, R. Raman, P. Nedungadi, Teacher perception of Olabs pedagogy, in *Fourth International Congress on Information and Communication Technology* (Springer, 2020), pp. 419–426
7. Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, H. Li, Learning to rank: from pairwise approach to listwise approach, in *ICML ’07* (2007)
8. Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.* **4**, 933–969 (2003)
9. T.-Y. Liu, Learning to rank for information retrieval. *Found. Trends Inf. Retr.* **3**(3), 225–331 (2009). March
10. H. Li, A short introduction to learning to rank. *IEICE Trans.* **94-D**, 1854–1862 (2011)
11. A. Pramanik, H. Aggarwal, M. Jacob, Deep generalization of structured low rank algorithms (Deep-SLR) (2019)
12. C.J.C. Burges, From RankNet to LambdaRank to LambdaMART: an overview, Technical Report MSR-TR-2010-82 (2010)
13. Q. Wu, C. Burges, K. Svore, J. Gao, Adapting boosting for information retrieval measures. *Inf. Retr.* **13**, 254–270 (2010)
14. D. Metzler, W. Bruce Croft, Linear feature-based models for information retrieval. *Inf. Retr.* **10**(3), 257–274 (2007)
15. Y. Zhou, G. Qiu, Random forest for label ranking. *Expert Syst. Appl.* **112**, 99–109 (2018). Dec
16. T. Qin, T.-Y. Liu, X. Jun, H. Li, LETOR: a benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.* **13**(4), 346–374 (2010). August
17. C. Macdonald, R.L.T. Santos, I. Ounis, On the usefulness of query features for learning to rank, in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM ’12*, New York, NY, USA (Association for Computing Machinery, 2012), pp. 2559–2562
18. J. Palotti, L. Goeuriot, G. Zucco, A. Hambury, Ranking health web pages with relevance and understandability, in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’16*, New York, NY, USA (Association for Computing Machinery, 2016), pp. 965–968

19. L. Soldaini, N. Goharian, Learning to rank for consumer health search: a semantic approach, in *Advances in Information Retrieval*, ed. by J.M. Jose, C. Hauff, I.S. Altingovde, D. Song, D. Albakour, S. Watt, J. Tait (Springer, Cham, 2017), pp. 640–646
20. M. Alsulmi, B. Carterette, Improving medical search tasks using learning to rank, in *2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, May 2018, pp. 1–8
21. V. Dang, M. Zarozinski, RankLib. <https://sourceforge.net/p/lemur/wiki/RankLib/>
22. H. Valizadegan, R. Jin, R. Zhang, J. Mao. Learning to rank by optimizing NDCG measure, in *Advances in Neural Information Processing Systems*, vol. 22, ed. by Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, A. Culotta (Curran Associates, Inc., 2009), pp. 1883–1891

An Algorithmic Game Theory Approach for the Stable Coalition and Optimum Transmission Cost in D2D Communication



Mahima Chaudhary, Anjana Jain, and Shekhar Sharma

Abstract In the proposed work, the problem of stabilizing coalition along with optimizing transmission cost for mode selection in D2D communication is considered. We considered a scenario of multiple D2D and cellular links in a single cell, and the D2D link can communicate through any of the three modes of transmission: cellular, reuse and dedicated modes. The proposed solution is based on a coalition game among D2D links for selecting the transmission modes. The joining or leaving of a coalition of a D2D link will be done based on individual transmission costs. The transmission cost is the function of transmission power and the cost of sub-channel occupancy. For stabilizing this coalition game and to get optimized transmission cost, a matching game is implemented between the coalitions and D2D links. A stable coalition means no D2D link can change its transmission mode and have lower transmission cost without making others worse off. We have also presented two cases along with one challenging study case, where we have performed our study with five D2D links and three modes of transmission in a single cell and showed the stable matching with optimized transmission cost.

Keywords Device-to-device (D2D) communication · Transmission cost · Coalition game · Matching game · Mode selection

M. Chaudhary (✉) · A. Jain · S. Sharma

Shri Govindram Seksaria Institute of Technology and Science, Indore, India
e-mail: mahimac388@gmail.com

A. Jain
e-mail: jain.anjana@gmail.com

S. Sharma
e-mail: shekhar.sgsits@gmail.com

1 Introduction

Device-to-device (D2D) communication enables direct communication between two mobile users without traversing the base station (BS). It can occur in both cellular spectrum (inband) and unlicensed spectrum (outband) [1]. ‘For providing very high data rates (multi-Gbps) to mobile devices in a 5G cellular network, mmWave communication is a promising technology’ [2]. To communicate directly in short-range proximity, mmWave can be applied to D2D-enabled wireless devices. The advantage of D2D communication is that it increases spectral efficiency and reduces delay [1]. There are three modes in which D2D user equipment (UE) can operate:

- (i) **Cellular mode**-D2D user equipment transmits the data through BS, similar to the cellular UE in this mode.

For transmitting the data, this mode requires more resources, but interference management is easier.

- (ii) **Reuse mode**-Reuse mode increases the spectral efficiency, by reusing some of the radio resources; D2D UE will transmit data directly to each other.

While using cellular mode, it can interfere with other D2D users and cellular users, but by using this mode, high spectral efficiency can be achieved.

- (iii) **Dedicated mode**- In this mode, D2D user will transmit their data directly through a dedicated portion of the spectrum. While using this mode, interference can be completely avoided because of the reserved resources for D2D communication but it has very poor spectrum efficiency.

In [5], they have considered a scenario of single and multiple cells only for one D2D link and cellular link, where they proposed the mode selection algorithm in which while satisfying the SINR constraint of the cellular network, the highest sum-rate is achieved.

In [6], they proposed an algorithm of mode selection and power control to maximize the sum-rate of cellular and D2D communication, for one D2D link and cellular link in a single cell.

In [7], mixed integer non-linear program with SINR constraint for cellular and D2D communication was proposed for sum-rate maximization in a single cell but for multiple cellular and D2D links.

In [8], for minimizing the transmission power subject to rate constraint, a Heuristics algorithm was proposed for multiple D2D links and cellular links in a single cell. But for D2D communication, they have only used cellular mode and dedicated mode.

All the above work assumes to achieve an optimal solution that UEs will always cooperate, but UEs are self-interested and can have their own communication requirements, and so their assumption may not hold for every scenario.

So, here, we are considering, depending on UEs requirement, different transmission modes can be used, and the links which are using the same modes can cooperate. That is why here coalition game theory can be implemented to examine the formation of cooperative games.

In paper [3], they used the discrete-time Markov chain (DTMC) to stabilize the coalition. They proposed an algorithm in which the D2D link will randomly select the coalition then puts a check whether the transmission cost is low and if it is low, then there are no further checks for other coalitions to get the lower transmission cost, but in our work, we are using matching game to stabilize the coalition wherein the preference profile for D2D links, they will prefer only that coalition first where they are getting lowest transmission cost.

The rest of the paper is organized as follows. Section 2 describes the coalition formation for optimum transmission costs. Section 3 presents the matching game formulation. Section 4 presents the case study. Section 5 presents the challenge. Section 6 presents the conclusion.

2 Coalition Formation for Optimum Transmission Cost

A coalition is a group of D2D links that are working cooperatively to select sub-channels and the corresponding mode of transmission. The D2D link will only leave a coalition and join another if it is getting lower transmission costs than its current coalition. And for each D2D link, the coalition game theory is used to select their transmission mode.

We considered a cell P cellular and Q D2D links sharing K sub-channels. The connection between transmitter of D2D user equipment and the receiver of D2D UE is a D2D link $i \in \mathcal{L} = \{1, \dots, L\}$ and the connection between a cellular UE and base station (BS) is a cellular link $j \in \mathcal{M} = \{1, \dots, M\}$ and has a rate requirement of R_j . Assuming that all the sub-channels have the same bandwidth BHz. For avoiding interference among the D2D links in the same coalition, orthogonal channels are used by the links for transmission.

For a coalition game model, we consider Q D2D links as players, and the coalition of players is denoted by $G_c, G_r, G_d \subseteq \mathcal{L}$, where G_c, G_r , and G_d are coalitions of D2D links using the cellular mode, the reuse mode, and the dedicated mode, respectively. Each link can use only one mode at a time, so

$$G \cup G_r \cup G_d = \mathcal{L} \text{ and } G \cap G' = \emptyset \text{ for any } G, G' \in \{G_c, G_r, G_d\} \text{ and } G \neq G'.$$

In each coalition $G \in \{G_c, G_r, G_d\}$, sub-channels are chosen by the members of coalition such that the sum of transmission power should be minimized while keeping rate requirement satisfied [3]:

$$\text{minimize } \sum_{k \in A_G} \sum_{i \in G} p_i^k \quad r \geq R_i, \quad \forall i \in G \quad (1)$$

where

A_G = set of available sub-channels for each coalition $G \in \{G_c, G_r, G_d\}$.
 p_i^k = transmission power of D2D link i on sub-channel k.

2.1 Transmission Cost

‘The transmission cost of the D2D link is a function of transmission power and the cost of channel occupancy’ [3].

Transmission cost of D2D link = $f(\text{transmission power, cost of channel occupancy})$

$$u(G, k) = \delta_i p_i^k(G) + \alpha_i c_i^k(G) \quad (2)$$

where,

p_i^k = transmission power of link i on subchannel k (in mW)

$c_i(G) \geq 0$: price of using subchannel k when D2D link i is in coalition $G \in \{G_c, G_r, G_d\}$

$c_i(G_c), c_i(G_r) \leq c_i(G_d)$: cost of channel occupancy for the dedicated be greater cellular and reuse mode coalition.

δ_i = positive weight constant of the transmission power.

α_i = positive weight constant of the price of subchannel occupancy.

2.2 The Condition Required for the Movement of D2D Link from One Coalition to Another

1. Each D2D link aims to minimize its transmission cost [3] from Eq. (2)

$$u_i(G, k) = \delta_i p_i^k(G) + \alpha_i c_i^k(G) \quad (3)$$

2. Each D2D link I will decide to leave its current coalition and join the new coalition only if it is getting lower transmission cost as compared to its current coalition and new coalition will only accept those D2D links for which other players are not getting higher transmission cost than the ones they are getting when they are in their current coalition [3].

$$u(G' \cup \{i\}) < u_i(G) \quad i \in G \quad (4)$$

$$u(G' \cup \{i\}) < u_i(G') \quad i \in G'$$

where,

$$G, G' \in \{G_c, G_r, G_d\}$$

$$G \neq G'$$

$$i \neq i'$$

3 Matching Game Formulation

'Matching [9] is implemented to optimally match resources and users have given their individual often different objectives and learned information.' It is performed between two sets of players with the help of the preference profile [4]. The preference profile is used by each player to rank the players of the opposite set. In this paper, we are using a many-to-one matching game for stabilizing the coalition game; here, in coalition set, one coalition can be matched to multiple players or D2D links, while in the D2D link set, every player has exactly one match. So, in the proposed matching game, the two sets are

- (a) Coalition $G \in \{G_c, G_r, G_d\}$

where,

G_c = group of D2D links using the cellular mode

G_r = group of D2D links using the reuse mode

G_d = group of D2D links using the dedicated mode.

- (b) D2D links or players $i_n = \{i_1, i_2, \dots, i_n\}$.

3.1 Preference Profile for D2D Links or Players I

The preference profile of D2D links will depend on transmission cost function from Eq. (2), i.e.,

$$u_i(G, k) = \delta_i p_i^k(G) + \alpha_i c_i^k(G)$$

Every link will calculate their transmission cost in each of three conditions and place only those coalitions in their preference profile for which transmission cost is lower as compared to the transmission cost in their current coalition.

Coalition with the lowest transmission cost will be preferred first. The mathematical representation of the above condition is given by Eq. (3)

$$u(G' \cup \{i\}) < u_i(G) \quad i \in G$$

where

$$G, G' \in \{G_c, G_r, G_d\}$$

$$G \neq G'$$

$$i \neq i'$$

3.2 Preference Profile for Coalition G

To form the preference profile of the coalition set, the coalition will put only one check on all the players and that is, which of the D2D links provide less or equal transmission costs to the players which are already present in that coalition as compared to the transmission cost that one's incurred when they are in their current coalition. It can be mathematically represented by Eq. (4).

So, the preference profile of a coalition will contain only those players for which the above condition is true.

Note The preference order of the coalition set preference profile is not going to affect the stability of the coalition as well as matching.

Only the preference order of the players will be considered.

4 Case Study

For the case study, we considered a single cell with five D2D links i_n and these D2D links can use any of the three transmission modes.

The D2D links or players transmitting in the same mode will form a coalition and as there are three transmission modes, there will be three coalitions

Coalition $G \in \{G_c, G_r, G_d\}$

where

G_c = group of D2D links using the cellular mode

G_r = group of D2D links using the reuse mode

G_d = group of D2D links using the dedicated mode

D2D links or players $i_n = \{i_1, i_2, \dots, i_n\}$.

So, the above two sets, coalition set and players set will form two sides for the many-to-one matching game to stabilize the coalition game theory along with optimizing the transmission cost. Assuming that the channel state information of all the links needs the information, the coordinator will distribute it.

4.1 Keeping the Size of Preference Lists Less or Equal to the Number of Players for Coalition Sets and the Number of the Coalition for a Player Set

Assume for this case the preference profiles for coalition and players as following (Tables 1 and 2).

After observing the preference profile of the coalition side, it is well known that for only players i_3 and i_4 , a group of D2D links already using cellular mode is getting either lower or equal transmission cost. Similarly, i_1 , i_2 , and i_5 for reuse mode coalition and i_1, i_2, i_3, i_4, i_5 for dedicated mode.

The preference profile of players shows that for i_1 player G_c is the coalition for which it is getting lowest transmission cost the G_r and then G_d , i.e., the order of transmission cost for i_1 player is

$$G_c < G_r < G_d$$

Similarly, the order of transmission cost for i_2 player is

$$G_d < G_c$$

Table 1 Preference profile of coalition for case study 4.1

| Coalition | Players or D2D links | | | | |
|-----------|----------------------|-------|-------|-------|-------|
| G_c | i_3 | i_4 | | | |
| G_r | i_1 | i_2 | i_5 | | |
| G_d | i_1 | i_2 | i_3 | i_4 | i_5 |

Table 2 Preference profile of players for case study 4.1

| Players | Coalitions | | |
|---------|------------|-------|-------|
| i_1 | G_c | G_r | G_d |
| i_2 | G_d | G_c | |
| i_3 | G_r | G_c | |
| i_4 | G_c | G_d | |
| i_5 | G_d | | |

The order of transmission cost for i_3 player is

$$G_r < G_c$$

The order of transmission cost for i_4 player is

$$G_c < G_d$$

The order of transmission cost for i_2 player is only

$$G_d$$

By following iterations, we are able to get stable many-to-one matching between coalitions and players:

$i_1 \xrightarrow{\times\!\times\!>} G_c$: i_1 player wants to join the G_c coalition but in the preference profile of G_c , i_1 is not present so, i_1 player will now try to join its second preference.

$i_1 \longrightarrow G_r$: As G_r player has i_1 in its preference profile, so i_1 will join G_r .

$i_2 \longrightarrow G_d$: i_2 will join the G_d coalition.

$i_3 \xrightarrow{\times\!\times\!>} G_r$: Player i_3 wants to join the G_r coalition but the preference profile of G_r did not have i_3 , so i_3 player will now try to join its second preference.

$i_3 \longrightarrow G_c$: As G_c player has i_3 in its preference profile, so i_3 will join G_c .

$i_4 \longrightarrow G_c$: i_4 will join the G_c coalition.

$i_5 \longrightarrow G_d$: i_5 will join the G_d coalition.

Above symbols represent:

\longrightarrow represents the player on the left will join the coalition on the right where the arrow is directing.

$\xrightarrow{\times\!\times\!>}$ represents the player on the left is not able to join the coalition on the right.

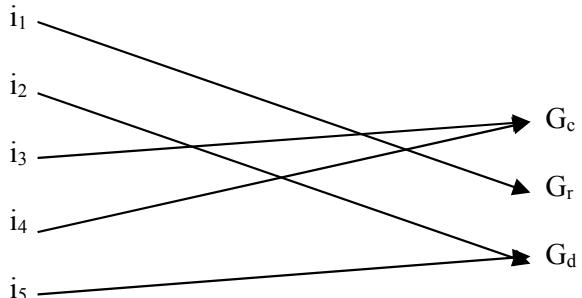
After performing the above iterations, the following result and its pictorial representation (Fig. 1) is observed:

i_1 joined G_r .

i_2 and i_5 joined G_d .

i_3 and i_4 joined G_c .

Fig. 1 Pictorial representation of stable matching between coalition and players



4.2 If the Players Set Have Only Two Coalitions with Lower Transmission Cost in Their Preference Profile and the Coalition Set Have Only Three Players in Their Preference Profile

Assume for this case the preference profiles for coalition and players are as follows (Tables 3 and 4).

A stable matching is achieved through following iterations:

$i_1 \longrightarrow G_d$: Player i_1 will join the G_d coalition.

$i_2 \not\longrightarrow G_r$: Player i_2 wants to join the G_r coalition but the preference profile of G_r did not have i_2 , so i_2 player will now try to join its second preference

$i_2 \longrightarrow G_d$: Player i_2 will join the G_d coalition.

$i_3 \not\longrightarrow G_c$: Player i_3 wants to join the G_c coalition but the preference profile of G_c did not have i_3 , so i_3 player will now try to join its second preference.

$i_3 \longrightarrow G_r$: As G_c player has i_3 in its preference profile, so i_3 will join G_c .

$i_4 \not\longrightarrow G_r$: Player i_4 wants to join the G_r coalition but the preference profile of G_r did not have i_4 , so i_4 player will now try to join its second preference.

$i_4 \longrightarrow G_c$: i_4 will join the G_c coalition.

$i_5 \not\longrightarrow G_d$: Player i_5 wants to join the G_d coalition but the preference profile of G_d did not have i_5 , so i_5 player will now try to join its second preference.

$i_5 \longrightarrow G_r$: i_5 will join the G_r coalition.

Above symbols represent:

\longrightarrow represents the player on the left will join the coalition on the right where the arrow is directing.

$\not\longrightarrow$ represents the player on the left is not able to join the coalition on the right.

After performing the above iterations, the following result and its pictorial representation (Fig. 2) is observed:

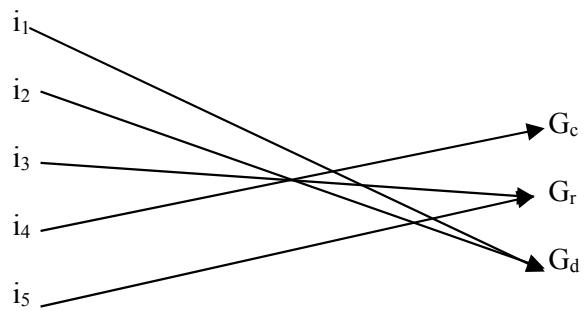
Table 3 Preference profile of coalition for case study 4.2

| Coalition | Players or D2D links | | |
|-----------|----------------------|-------|-------|
| G_c | i_2 | i_4 | i_5 |
| G_r | i_1 | i_3 | i_5 |
| G_d | i_1 | i_2 | i_4 |

Table 4 Preference profile of coalition for case study 4.2

| Players | Coalitions | |
|---------|------------|-------|
| i_1 | G_d | G_c |
| i_2 | G_r | G_d |
| i_3 | G_c | G_r |
| i_4 | G_r | G_c |
| i_5 | G_d | G_r |

Fig. 2 Pictorial representation of stable matching between coalition and players



i_4 joined G_c .

i_2 and i_1 joined G_d .

i_3 and i_5 joined G_r .

After performing two cases 1 and 2, it can be observed that in first case, two D2D links joined cellular transmission mode coalition, one D2D link joins the reuse transmission mode coalition, and other two joined dedicated transmission mode coalition, while in the second case, only one D2D link joined cellular transmission mode coalition, other two links joined reuse transmission mode coalition, and remaining two joined dedicated transmission mode.

This showed theoretically that interference management will be easier in first case but second case is spectral efficient.

5 Challenge

The challenge observed after performing the above cases is.

If a player has any two of the coalition in its preference profile, but the same coalition does not have this player in their respective preference profiles, then that player will remain unmatched and will not be able to join any other coalition. This can be shown through the following case:

Assume the following preference profiles of the coalitions and players (Tables 5 and 6).

Following are the iterations for matching

$i_1 \longrightarrow G_c$: Player i_1 will join G_c coalition.

Table 5 Preference profile of coalition for challenging case

| Coalitions | Players or D2D links | | | | |
|------------|----------------------|-------|-------|--|-------|
| G_c | i_3 | i_1 | | | |
| G_r | i_1 | i_2 | i_5 | | i_4 |
| G_d | i_2 | i_3 | i_5 | | |

Table 6 Preference profile of players for challenging case

| Players | Coalitions | | |
|---------|------------|-------|-------|
| i_1 | G_c | G_r | G_d |
| i_2 | G_d | G_c | |
| i_3 | G_r | G_c | |
| i_4 | G_c | G_d | |
| i_5 | G_d | | |

$i_2 \longrightarrow G_d$: Player i_2 will join G_d coalition.

$i_3 \cancel{\longrightarrow} G_r$: Player i_3 wants to join the G_r coalition but the preference profile of G_r did not have i_3 , so i_3 player will now try to join its second preference.

$i_3 \longrightarrow G_c$: Player i_3 will join G_c coalition.

$i_4 \cancel{\longrightarrow} G_c$: Player i_4 wants to join the G_c coalition but the preference profile of G_c did not have i_4 , so i_4 player will now try to join its second preference.

$i_4 \cancel{\longrightarrow} G_d$: Player i_4 wants to join the G_d coalition but the preference profile of G_d did not have i_4 , so i_4 player will now try to join its third preference.

$i_5 \longrightarrow G_d$: Player i_5 will join the G_d coalition.

Above symbols represent:

\longrightarrow represents the player on the left will join the coalition on the right where the arrow is directing.

$\cancel{\longrightarrow}$ represents the player on the left is not able to join the coalition on the right.

From the above iterations, it is observed that player i_4 is unmatched, which means there is no coalition where i_4 can get the lowest transmission cost.

6 Conclusion

In the above-proposed work, by implementing many-to-one matching, an algorithmic game approach, stability of coalition, and optimum transmission cost are achieved which was shown mathematically by two cases, which represent that without making other worse off, a D2D link cannot change its communication mode and have low transmission cost. A challenging case was also presented in the paper in which for some D2D links, there is no coalition present where it can get lowest transmission cost so it have to settle for the lower transmission cost.

References

1. A. Asadi, Q. Wang, V. Mancuso, *A Survey on Device to Device Communication in Cellular Networks* (IEEE, 2014)

2. X. Shen, Device-to-Device communication in 5G cellular networks. *IEEE Network Magazine* (2015)
3. K. Akkarajitsakul, P. Phunchongharn, E. Hossain, V.K. Bhargava, Mode selection for energy-efficient D2D communications in LTE-advanced networks: Coalitional game approach, in *Proceedings of IEEE ICCS* (2012)
4. Y. Gu, W. Saad, M. Bennis, M. Debbah, Z. Han, Matching theory for future wireless networks: fundamentals and applications. *IEEE Communications Magazine*, 2015
5. K. Doppler, C.H. Yu, C.B. Ribeiro, P. Janis, Mode selection for device-to-device communication underlaying an LTE-Advanced network, in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, April 2010
6. C. Yu, K. Doppler, C.B. Ribeiro, O. Tirkkonen, Power optimization of device-to-device communication underlaying cellular communication, in *Proceedings of IEEE International Conference on Communications*, June 2009
7. M. Zulhasnine, C. Huang, A. Srinivasan, Efficient resource allocation for device-to-device communication underlaying LTE network, in *Proceedings of IEEE 6th International Conference on Wireless and Mobile Computing* (2010)
8. X. Xiao, X. Tao, J. Lu, A QoS-aware power optimization scheme in OFDMA systems with integrated device-to-device (D2D) communications, in *Proceedings of IEEE Vehicular Technology Conference (VTC Fall)*, 2011, 5–8 September 2011
9. D. Gale, L.S. Shapley, College Admissions and Stability of Marriage, in *The American Mathematical Monthly*, vol. 69, No. 1 (Mathematical Association of America, 1962)
10. Narhari, *Game Theory and Mechanism Design* (IISc Press)
11. M.J. Osborne, A. Rubinstein, *Electronic Version of a Course in Game Theory* (MIT Press)
12. Q. Wang, B. Rengarajan, Recouping opportunistic gain in dense base station layouts through energy-aware user cooperation, in *Proceedings of IEEE WoWMoM* (2013)
13. P. Cheng, L. Deng, H. Yu, Y. Xu, H. Wang, Resource allocation for cognitive networks with D2D communication: an evolutionary approach, in *Proceedings of IEEE WCNC* (2012)
14. B. Zhang, X. Mao, J-L. Yu, Z. Han, Resource allocation for 5G heterogeneous cloud radio access networks with D2D communication: a matching and coalition approach. *IEEE Trans. Veh. Technol.* (2018)
15. P. Golle, A Private Stable Matching Algorithm (Palo Alto Research Center)
16. L. Shapley, A. Roth, Stable Matching: Theory, Evidence, and Practical Design, the Prize in Economic Sciences 2012 (Royal Swedish Academy of Science)

A Study of Hybrid Approach for Face Recognition Using Student Database



Sarika Ashok Sovitkar and Seema S. Kawathekar

Abstract The student's attendance plays a very important role during their academic year. Teachers have to spend a lot of time and attention to maintaining the students' attendance records punctually and regularly. Face recognition is the best and smart solution for the maintenance and keeps a regular attendance record of the students for the teachers in their academic year so that they can concentrate on the student's academic developments. Here, we combine two basics feature extraction techniques, i.e., PCA for low dimensionality reduction and LDA for feature extraction whereas an SVM classifier is used for face classification. We perform the experiment using the Student Database (SDB) of different age groups of students, which are collected in different lighting conditions, with different facial poses and expressions. The results obtained from the combination of PCA, LDA are very satisfactory and we successes to reduce the time duration for marking attendance.

Keywords Eigen face · Fisher face · PCA · LDA · SVM · Feature extraction

1 Introduction

An automated attendance using face recognition is the confirmation of student's physical presence in a classroom is compulsory, it also prevents the fraud attendance caused in manual or other automated attendance systems [1] due to various reasons such as in manual system it is very difficult to identify the attendance given by the right person [1]. In RFID automated system, if the card is misplaced or misused by an unknown person [2]. In Bluetooth automated system, mobile phone compulsory instead of the student, which leads to unnecessary use of mobile and in many organizations, there is a restriction on use of mobile [3]. Fingerprint and iris are

S. A. Sovitkar (✉) · S. S. Kawathekar
CSIT Dept, Dr. BAM University, Aurangabad, India
e-mail: sarikakondekar@gmail.com

S. S. Kawathekar
e-mail: seema_babrekar@yahoo.co.in

the biometric technology used for automated attendance but they also have their own drawbacks such as in iris automated attendance, it is complicated to place the transmission lines in the places where the corneal topography is terrible [4]. Also in the fingerprint attendance system, students are waiting in a queue for thumb, it is very time-consuming for their attendance or passing the finger recognition device (a wireless device) during the lecture time may possibly divert the concentration of the student [5].

2 Face Recognition

Face recognition is the study of pattern recognition, computer vision and biometric system. We as human beings recognize faces from our childhood consciously or unconsciously, we were able to identify and classify the person as our family member, neighborhoods, friends, relatives, etc. The ability to recognize a person faces is of vital public consequence for humans and evolutionarily essential for continued existence. As a result, faces may be “*special*” stimuli, for which we have adopted a special, unique method for recognition processes. The most common confirmation for face processing being modular come from cases of *prosopagnosia*, where patients are unable to recognize faces at the same time as retaining the ability to recognize other objects.

In terms of machine, face recognition is a very complicated and tedious job, as it requires a large memory to store a digital image. Due to curiosity and interest in this era, many researches have been done from the last three decades and it still goes on. Face recognition is still a challenge, as it mainly concentrated on the physical appearance of a person; in which the physical features such as eyes, nose, mouth, eyebrows, etc. are extracted and used for identification for a person. Face recognition generally divided into three categories such as Face Detection, Feature Extraction, and Classification.

A. Face Detection. The first step towards FR is face detection, which means locating the region in an input digital image; there the face is to be found. The Viola-Jones algorithm is a broadly used method for object detection. The main possessions of this algorithm are that training is slow, but detection is fast, this means that it takes several time for feature extraction from given input images as we capture images with respect to all the possible variations in pose and facial expressions. This algorithm uses Haar basis feature filters, so it does not use multiplications. Each face recognition filter (from the set of N filters) contains a set of cascade-connected classifiers. Each classifier looks at a square compartment of the detection windowpane and determines if it looks like a face. If it matches, then the next classifier is applying. If all classifiers give a positive answer, the filters reply a positive answer, and the face is recognized [6] (Fig. 1).

B. Feature Extraction Methods. Features are nothing but the core or center part of any kind of things refer for identification or recognition such as we can identify a

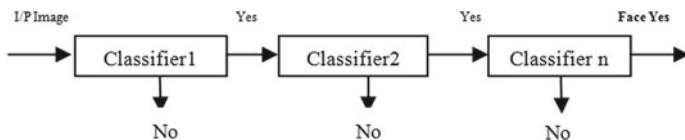


Fig. 1 Viola-Jones face detection method

different kind of trees using their features such as leaf structure, physical properties, chemical properties, etc., we can recognize a person by identifying the face features such as mouth, nose, eyes, eyebrows, etc. [6–8]. Features are nothing but the pattern present over which is used for the identification of an object. We use feature extraction techniques in object recognition also using pattern recognition. As we identify a face using the feature recognition technique, we apply the same idea in machine for face recognition, i.e., we extract features such as mouth, nose, eyes, etc. present on face and use it for identification by comparing with test face image.

There are several methods proposed all over the world for feature extraction they are generally categorized as geometric based, color segmentation based, appearance based and template based techniques. In geometric based technique, the whole face is used for identification, in this method, the position of eyes, mouth, nose, eyebrows, and shape of face are considered as features. This is a very time-consuming and costly method, generally used for criminal identification in security system. In color segmentation based method, the skin color is used as a feature, like the other body part skin color is the same, it is not so effective process, but it is a very simple and easy method. The appearance based techniques are combination of statistical and linear transformation methods in which a face is represented as a vector. The template based technique also called local feature extraction techniques. In this method, we extracted features such as eyes, nose mouth from a face, and used as a template for matching with the test image, if the match found then the face is recognized otherwise unknown image [7]. Figure 2 represents samples for local and global features.

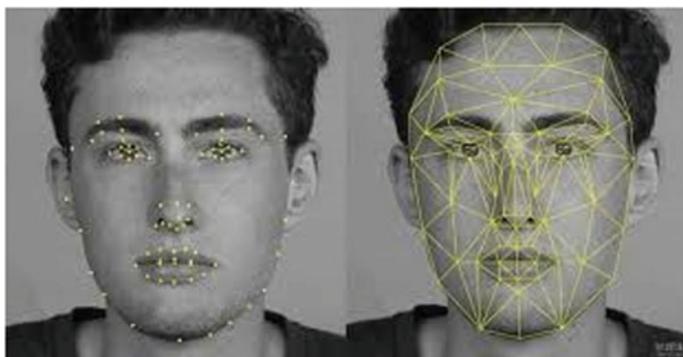


Fig. 2 Facial features

3 Proposed System Using Hybrid Approach

As we focus on template-based feature extraction techniques, the proposed system use the combination of PCA and LDA known as hybrid method used for better feature extraction [9]. The structural design is as shown in Fig. 3. It works into three phases such as Training Phase, pre-processing, testing phase.

A. Training Phase. We capture the face with different face pose, expressions, and at different angles. The photos were capture in real-time using a laptop camera to vary in light illumination condition. We capture at least 8–10 images per person having all the possibilities of variations like face occlusion, wearing sunglasses, having bread, face covered with scarf, etc. The Student Database contains all the possible age groups photos, i.e., from Kinder Garden to Graduate students with 40 minimum samples for each group, for our convinces we broadly categories them into five groups such as PAG (Primary Age Group), SAG (Secondary Age Group), HAG (Higher-Secondary Age Group), GAG(Graduation Age Group), PGAG (Post-graduate and above age group). Each group contains near about 400–500 photos of the students (Fig. 4).

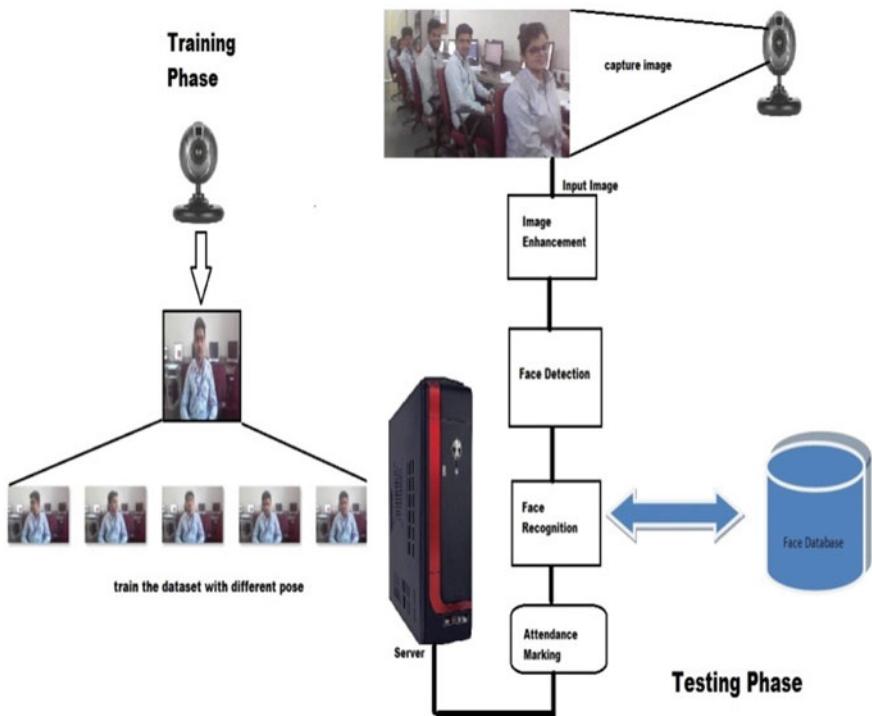


Fig. 3 Overall architecture of proposed system



Fig. 4 Samples from training student database

B. Pre-Processing. This is the most important step in our experiment, as we perform the experiment using real-time database the captured image contains noise, the images vary in size also they are colored images. We need to improve the quality of an image using an image enhancement technique; also we convert the given RGB image into grayscale image for further processing. The image is stored in 92×112 grayscale. The same process is repeated for testing phase, in which the input image contains unknown face which is going to recognize before that the faces are cropped from the image the given input image is converted into 92×112 sized grayscale image after face detection feature are extracted from detected face for classification using SVM as classifier [10, 11].

C. Testing. It is the last stage in an experiment in which from the given input image the faces are identified with the training database and show the results in the form of the number of males and females present in a classroom with their attendance updating in an excel sheet (Table 1).

Table 1 Attendance sheet updated using excel sheet

| Roll No. | Name | Present or absent |
|----------|-----------|-------------------|
| 1. | Amruta | P |
| 2. | Sandeep | A |
| 3. | Sarika | P |
| 4. | Ashok | A |
| 5. | Samidha | P |
| 6. | Sammed | A |
| 7. | Mohini | P |
| 8. | Pandurang | P |
| 9. | Rameshwar | A |
| 10. | Krishna | P |

4 Method

Principal Component Analysis (PCA) is the most common method used for dimensionality reduction. It is the first algorithm that recognizes the faces inexpensively. In PCA, we compute the Eigen faces using mathematical operations, and their consequent projections along each Eigen face are also calculated [9]. Instead of using all the dimensions of an image only meaningful dimensions are measured to represent the image.

Mathematical representation of an image using PCA,

$$\chi = WY + \mu$$

where χ is vector face, Y is vector of eigen faces,
 W is the feature vector, and μ is the average vector face.

These features vector also known as Eigen vectors are then used in classification for face recognition.

Afterward, a **Linear Discriminator Analysis (LDA)** was represented, were the ratio of between-class scatter matrix (S_b) and within-class scatter matrix (S_w) was measured.

The hybrid approach is the concatenation of two best methods to overcome the drawbacks of each method and the derived method results will be encouraging and improved the recognition rate. Face recognition algorithm based on Hybrid approach (PCA + LDA) is the union of PCA and LDA, two different kinds of methods. The combination of two methods in such a way that, PCA reduces the dimension of the data and also data redundancy is minimized as orthogonal components and LDA is used to preserve the class discriminatory information as much as possible. As PCA focus on similarities within the feature of an object whereas LDA extract the different features from an object. For classification, we considered Support Vector Machine (SVM) will be the best as it is used mostly in classification techniques. SVM is an effective pattern classification algorithm recently proposed by the researchers [12]. For pattern recognition SVM finds the best separation of closest points in the training dataset.

The general steps involved in hybrid approach are:

Algorithm 1: The Training Phase

1. Training the input dataset using different poses and variations in facial expressions.
2. Convert the colored images into grayscale images.
3. Get each image into a vector form.
4. Calculate the mean image.
5. Normalizes the each input image by subtracting it from mean.
6. Calculate covariance matrices by keeping the K largest Eigen values.
7. Compute the Eigen vector of covariance matrices.



Fig. 5 Sample of mean face image



Fig. 6 Sample of Fisher image

8. Compute the Eigen faces containing the measure features of face.
9. Calculate the projection face image.

Algorithm 2: The Testing Phase

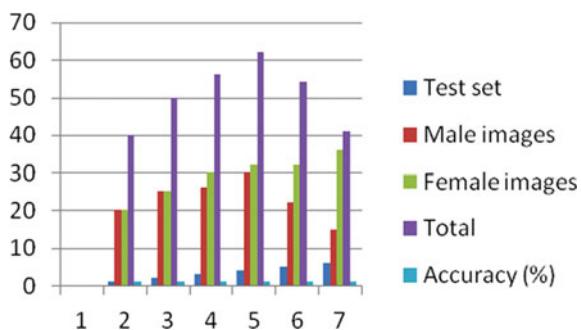
1. Same as training phase, calculate the Eigen face for given input image.
2. Compute the d -dimensional mean vectors.
3. Computing the *Scatter Matrices*, i.e., with-in class scatter matrices S_w and between classes scatter matrices S_b .
4. Solve the generalized Eigen value problem for the matrix $S_w^{-1} S_b$.
5. Select linear discriminates for new feature subspace.
6. Transform the samples onto the new subspace known as Fisher face.
7. Compare the projected image with the sample space for classification (Figs. 5 and 6).

5 Result Analysis

We perform an experiment in different light conditions using the student database. The images are captured using HP Laptop cameras, using natural light at a distance of 4–7 ft away from the camera, the students are seating in a classroom with normal facial expressions. The system also counts the total number of student present in a classroom with boys and girls count. The result is shown in the following Table 2 and Chart 1 (Fig. 7).

Table 2 Face recognitions with varies number of student

| Test set | Male images (*8 pose) | Female images (*10 pose) | Total | Accuracy (%) |
|----------|-----------------------|--------------------------|-------|--------------|
| 1 | 20 | 20 | 40 | 95 |
| 2 | 25 | 25 | 50 | 94 |
| 3 | 26 | 30 | 56 | 93 |
| 4 | 30 | 32 | 62 | 90 |

Chart 1 Graphical presentation of Table 2**Fig. 7** An input data image @ distance 4 ft

6 Conclusion

PCA is weaker in discrimination whereas the LDA requires more time for discrimination. The hybrid approaches to overcome both the problem also increase in accuracy and decrease the time required for face recognition. The average recognition rate obtained here is 90%. We try to build a system which is capable to give the better result with more accuracy here are some limitation we can improve the camera quality by replacing the existing laptop camera by using high-resolution camera. The existing system requires 10–15 min for minimum 50–60 students' attendance, we want to improve it by updating the system hardware.

References

1. S.A. Sovitkar, S.S.Kawathekar, A review on-automated attendance management system using face recognition. *Int. Multi. Res. J.* **5**(12) (2016). ISSN: 2231-5063. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd edn., vol. 2. (Clarendon, Oxford, 1892), pp. 68–73
2. S.K. Jain, U. Joshi, B.K. Sharma, Attendance management system, Masters Project Report, Rajasthan Technical University, Kota
3. V. Bhalla, T. Singla, A. Gahlot, V. Gupta, Bluetooth based attendance management system. *Int. J. Innov. Eng. Technol. (IJIET)* **3**(1) (2013). ISSN: 2319 – 1058
4. M.M.H. Ali, V.H. Mahale, P. Yannawar et al., Overview of finger print recognition system, IEEE Xplore, 24 Nov 2016. <https://doi.org/10.1109/ICEEOT.2016.7754900>
5. M. Turk, A. Pentland, Eigenfaces for recognition. *J. Cognitive Neurosci.* **3**, 71–86 (1991)
6. S.A. Sovitkar, Dr. S.S. Kawathekar, A conceptual model for automated attendance system using principal component analysis (PCA). *Int. Cognitive Knowl. Eng. (ICKE)* (2016)
7. S. Chintalapati, M.V. Raghunadh, Automated attendance management system based, in IEEE International Conference on Computational Intelligence and Computing Research, 2013
8. P. Viola, M.J. Jones, Robust real-time face detection. *Int. J. Comput. Vis. IJCV* **57**(2), 137–154 (2004)
9. S.A. Sovitkar, S.S. Kawathekar, Comparative study of PCA and LDA algorithms for automated attendance system using face recognition, in International Conference on Researches in Science and Technology, 26 Feb 2017, pp. 52–56. ISBN: 9788192958056
10. A.K. Jain, Image Perception. *Digital Image Processing* (Pearson Publications, U.S. Edition, New Delhi, PHI Publication, Chap. 3, Sect. 1, 2), pp. 49–53
11. R.C. Gonzalez, R.E. Woods, S.L. Eddins, Representation and description, in *Digital Image Processing Using MATLAB*, 2nd edn (Tata McGraw Hill Publication, New Delhi, 2012, Chap. 11, Sect. 11.5), pp. 615–626
12. M.O. Faruqe, M. Al Mehedi Hasan, Face recognition using pca and svm, in 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication (IEEE, ASID, 2009), pp. 97–101

Multi-objective Consensus Clustering Framework for Flight Search Recommendation



Sujoy Chatterjee, Nicolas Pasquier, Simon Nanty, and Maria A. Zuluaga

Abstract In order to provide personalized recommendations for travel search queries to online customers, an appropriate segmentation of customers is required using information from the search query. Clustering ensemble approaches have been proposed to address well-known issues of classical clustering methods that each relies on a different theoretical model and can thus identify in the data space only clusters corresponding to this model, and ensemble methods aggregate diverse clustering solutions from dissimilar algorithmic configurations to generate more robust consensus clusters corresponding to agreements between initial clusters. We put forward a new clustering ensemble multi-objective optimization-based framework developed to improve personalized recommendations generated by the flight search engine of the company Amadeus. This framework optimizes diversity in the clustering ensemble search space and finds an appropriate number of clusters automatically without requiring any user input. Experimental results compare the efficacy of this method with other existing approaches on Amadeus customer flight search data in terms of the adjusted Rand index and a business metric defined and used by the company.

Keywords Clustering · Ensemble · Multi-objective optimization

S. Chatterjee · N. Pasquier
Université Côte d'Azur, CNRS, I3S, 7271 Sophia Antipolis, France
e-mail: pasquier@i3s.unice.fr

S. Chatterjee (✉) · S. Nanty · M. A. Zuluaga
Amadeus S.A.S, Sophia Antipolis, France
e-mail: sujoy.2611@gmail.com

S. Nanty
e-mail: simon.nanty@amadeus.com

M. A. Zuluaga
e-mail: zuluaga@eurecom.fr

M. A. Zuluaga
Data Science Department, EURECOM, Biot, France

1 Introduction

In the travel industry, multiple factors can hinder customers from purchasing a ticket in spite of searching for a flight itinerary. The click to conversion rate, which serves as a metric of the translation of the click (i.e., inquiry) rate into conversion, could be increased if a certain type of offers can be recommended to a certain set of customers. Personalized recommendations based on similarity of customers can be very effective to improve the business strategy of a travel company [13]. Customers can be segmented depending on multiple features (e.g., days to flight, distance covered, number of passengers, etc.). Hence, clustering algorithms have a major role in order to segment the customers in a better way.

In this context, clustering algorithms [9] are used to find sets of customers with similar needs and requirements and to identify hidden relationships between their search queries. Hence, clustering the same set of customers with different algorithmic configurations can produce significantly different solutions. Choosing an adequate algorithmic configuration and defining the final number of clusters, as required by most existing clustering approaches, are major practical issues when the required prior knowledge of the data space properties is unavailable. Consensus clustering combines multiple clustering results, obtained from diverse algorithmic configurations, to generate a more robust clustering solution [1, 2, 14–16]. Given its characteristics, it is an interesting solution for travel customer's segmentation with little prior knowledge of the evolving data space. To the best of our knowledge, no study on the integration of consensus clustering for better personalized recommendation in travel has been reported in the literature.

In this research, we study the integration of consensus clustering through a multi-objective optimization process for the clustering of flight search queries performed by different customers using the flight search engine of the company Amadeus. The clustering solution is used to segment the space of customers, so that the search engine is optimized independently for each cluster, and customers with different needs and requirements are provided with different recommendations. In this context, there is no prior knowledge about the data space modeling assumptions, like data distribution or natural number of clusters, and choosing an appropriate algorithmic configuration is an important issue.

The clustering ensemble problem [1, 2, 6, 11, 15, 16] is usually posed as an optimization problem where the average similarity of consensus solution with the base clusterings is maximized in order to obtain a better aggregation. However, since a consensus solution can be very similar to one base clustering, whereas very distant from others, and to remove any kind of bias toward a particular clustering solution, minimizing the standard deviation of these similarity values is also necessary. The proposed framework uses a multi-objective clustering ensemble approach that concurrently optimizes two objective functions as follows [3]: (1) The maximization of the similarity between the consensus solution and the base clusterings and (2) the minimization of the standard deviation of these similarities for a consensus solution. A formal proof is provided demonstrating that the proposed approach automatically

generates a number of clusters at least as appropriate as approaches that consider only co-occurrences of two objects in base clusters for generating a consensus clustering. This framework integrates the ensemble clustering solution and a mapping function to categorize a new customer in an appropriate cluster to improve flight search recommendations provided by the company Amadeus. Experimental results with comparative analysis of existing ensemble methods demonstrate that it works well in most cases. The full version of this work, which includes additional analysis, is available in [4].

2 Problem Formulation

Suppose $X = \{x_1, x_2, \dots, x_p\}$ denote a set of p customers, where $x_i \in \mathbb{R}^d$, d be the number of features used to describe the search query and Y be the set of n clustering algorithms. Here each x_j denotes the customer who searches a query for flight booking. Suppose $C = \{c_1, c_2, \dots, c_n\}$ be the set of base clustering solutions obtained after applying n clustering algorithms. Each c_i partitions p customers into k_i clusters such that $c_i = \{x_1^i, x_2^i, \dots, x_p^i\}$, where c_i belongs to the set of all possible partitions of X and x_j^i denotes the label of j th customer according to the i th clustering: $x_j^i \in \{1, 2, \dots, k_i\} \forall j \in \{1, 2, \dots, p\}$. It should be noted that each c_i might comprise different number of clusters k_i . The goal is to derive the best aggregated ensemble solution from the base clustering ones, while automatically deciding the number of clusters. In the subsequent sections of the paper, we use both the terms customers and objects in an analogous way.

3 Clustering Ensemble Framework

The proposed framework integrating consensus clustering optimization in Amadeus flight search engine is presented in Fig. 1. The upper part of the chart shows the creation of the search space, that is the refined ensemble clustering, from the dataset. The lower part of the chart shows the semi-supervised classification process for learning a customer classification model. The last two steps of this figure are the parts of the mapping function which integrates the consensus solution to the Amadeus flight search engine.

The central four phases of this ensemble framework are detailed in the subsequent subsections. Initially, the base clustering solutions are generated by applying various clustering algorithms with different k , and these are refined by estimating the maximum number of clusters that can appear among these base clustering. For evaluating this maximum number of clusters, a weighted co-association matrix [6] is constructed and an iterative process is applied on this matrix (as described in [3]). Then, depending upon the estimated number of clusters, the base clustering solutions

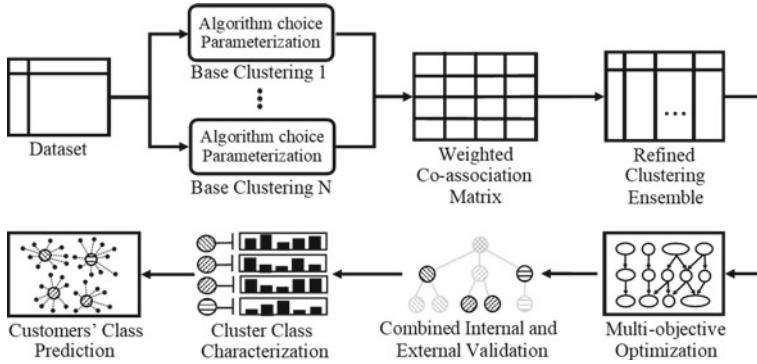


Fig. 1 Semi-supervised classification framework for customer segmentation

are re-labeled according to a reference clustering solution (as mentioned in [3]), and thus some solutions are refined. Ultimately, we combine the original base clustering with the refined set of clustering solutions, and thus, this ensures the diversity of the clustering solutions. In this step, we use a NSGA-II-based multi-objective optimization [5] to overcome the important complexity of the search for a single solution that is optimal in terms of all the objective functions.

This NSGA-II method [3] is applied on the set of refined clustering solutions to yield a final set of non-dominated Pareto optimal solutions. Finally, the consensus solution is integrated in the Amadeus application using a mapping function for classifying new customers and makes the Amadeus flight search engine return more personalized recommendations.

3.1 Weighted Co-association Matrix Based on Confidence

In this process, instead of a classical co-association matrix, a weighted co-association matrix [3] is constructed based on different factors like quality of the clustering solution and pair-wise confidence of two objects. The goodness of a solution is measured by the average similarity of this solution with respect to adjusted Rand index (ARI) [8] while compared to the other solutions.

Therefore, both the quality and the number of clusters of a clustering solution are taken into account to form the weighted co-association matrix. This co-association matrix can be treated as a similarity matrix, where the weights of edge rely on these two aspects, being computed in accordance with the number of clustering solutions that agree to group two particular objects in the same clusters. Furthermore, the quality of the solution, in terms of average similarity with respect to the other solutions, and the confidence of two objects [3] being members of the same clusters are also considered. In this process to measure the confidence, the solutions that contain a higher number of clusters are given greater weight than clustering solutions

having lower number of clusters. Besides, quality being an important factor in this purpose, higher weight is imposed on the quality metric (i.e., similarity) than the number of components. The weight owing to the confidence and quality is added to calculate the final weight as their similarity.

If n is the number of base clusterings and the label of each object j is represented by r_j , then the similarity $\text{Sim}(i, j)$ of two objects i and j is represented by using Eq. 1.

$$\text{Sim}(i, j) = \sum_{p=1}^n (I(r_i, r_j) * \text{cluster}(p)) + 2w \sum_{p=1}^n (I(r_i, r_j) \text{weight}(p)) \quad (1)$$

Here, I represents an indicator function, and it produces 1 when the labels of two objects are same, else it returns 0, $\text{cluster}(p)$ is the number of clusters in the p th solution and $\text{weight}(p)$ is the weight measured by the similarity of p th solution when compared to other base clustering solutions. Due to the difference of two ranges, i.e., variation in number of clusters and similarity value in respect of ARI, w is used to make the two ranges in the same scale [3].

The final number of clusters cannot be selected in straightway from the base clustering with maximum average similarity since a few clustering solutions can have a maximum similarity but have a number of connected components without sufficient stability. Finally, the initial labeling of the base clusterings is changed according to this estimated number of clusters as mentioned in [3].

Theorem 1 *The weighted co-association matrix of a clustering solution is calculated from the numbers of co-occurrences of two objects in the same cluster, the confidence of objects co-occurrences and the quality of the clustering solution. This approach will generate a number of clusters that is at least as close to the number of clusters in the best potential agreement as the number of clusters calculated using a simple co-association matrix where only the numbers of co-occurrences of two objects in clusters are considered.*

Proof Given two clustering solutions S_p and S_q with m and n number of clusters, respectively, suppose the confidence of two objects i and j of remaining in same cluster are $\text{Conf}_{S_p}^m$ and $\text{Conf}_{S_q}^n$. According to the proposed model, the values in the co-association matrix not only depend on the number of co-occurrences of two objects being in a same cluster. Rather, along with this, the confidence of two objects and the goodness of the clustering solutions are also taken into consideration.

Suppose G and G' are the graphs constructed from normal co-association matrix (where only count of co-occurrence of two objects lying in same clusters is considered) and weighted co-association matrix (where co-occurrence, confidence and quality of the clustering solutions are considered). Accordingly to the proposed model, the confidence of two objects remaining in the same clusters is higher for clusterings with a greater number of partitions than for clusterings with lower number of partitions. Therefore, we write $\text{Conf}_{S_p}^m(i, j) \geq \text{Conf}_{S_q}^n(i, j)$ where $m > n$ and if $(i, j) \in C_{S_p}^m$ then $(i, j) \in C_{S_q}^n$. Here, $C_{S_p}^m$ and $C_{S_q}^n$ are two clusters in clustering solution S_p and S_q where the two objects i and j are the members.

Now, the weighted co-association matrix is generated based on confidence, co-occurrence and quality of the clustering solution. The edges are iteratively removed by increasing repeatedly the threshold δ_t by a very small amount and observing each time the number of connected components obtained. This number of connected components basically denotes the number of clusters. Let $N_e(G)$ and $N_e(G')$ be the number of edges deleted from graphs G and G' each time.

Due to the weight in G' as mentioned previously, it can be written that $\forall \delta_t: N_e^{\delta_t}(G) \geq N_e^{\delta_t}(G')$, where $N_e^{\delta_t}(G)$ and $N_e^{\delta_t}(G')$ denote the number of edges e with same weight δ_t in graph G and G' , respectively. Therefore, $\forall \delta_t: N_c^{\delta_t}(G) \geq N_c^{\delta_t}(G')$, where $N_c^{\delta_t}(G)$ and $N_c^{\delta_t}(G')$ are the number of connected components extracted from G and G' for each value of threshold δ_t . We can thus compare the changes in the number of connected components for any two successive iterations in these two graphs, and we have $\frac{d(N_c(G'))}{dt} \leq \frac{d(N_c(G))}{dt}$ for any two successive iterations. Hence, we obtain a number of clusters that are at least as close to the number of clusters in the best agreement than the number of clusters obtained using a classical co-association matrix, where only the co-occurrences of two objects in a same cluster are considered.

After finding the appropriate number of clusters, the label transformation to produce refined base clustering solutions and NSGA-II-based optimization [3] are performed for deriving a set of non-dominated Pareto optimal solutions.

3.2 Mapping Function

Consensus clusters, corresponding each to a customer segment, are characterized to discriminate them, and new customers are classified by assigning them to the cluster they are the most similar to. The characterization of each cluster is represented as the center point of the cluster from the sample features. That is a mean vector of feature values of samples in the cluster. The similarity between a new customer and each cluster is then computed based on these vectors, and a new customer is assigned to the cluster with maximal similarity.

The mapping function for clustering solution C^i is denoted by $f: x \mapsto \text{argmin}_{\{1 \leq l \leq k\}}(d(\gamma_l, x))$, where x represents a new customer, k denotes the number of clusters in the solution, and $\gamma_l = \sum_{j=1}^p I(l - x_j^i)x_j / \sum_{j=1}^p I(l - x_j^i)$ represents the center of cluster l where $I(x) = 1$ for $x = 0$ and $0 \forall x \in \mathbb{R}^*$.

4 Experiments and Results

In the experiments, the crossover and mutation rate were 0.9 and 0.01. Population size was twice the number of the base clusterings.

4.1 Dataset Description and Preprocessing

Data was obtained from Amadeus flight search engine. Search queries for flights departing from the USA during one week on January 2018 were extracted with nine relevant features: distance between two airports, geography (i.e., domestic, continental or intercontinental flight), number of passengers, number of children, advance purchase, stay duration, day of the week of the departure, day of the week of the return and day of the week of search.

As the dataset comprised a huge amount of customers (in millions), and as many of them had similar feature values, the populations were divided into some strata based on similar characteristics. Sampling was performed on these subpopulations to generate a stratified sampling of the whole dataset while preserving the distribution properties of the original one. Finally, three stratified sample datasets were generated with sizes of 500, 1000 and 1500. After generating the sampled data, to prepare the base clustering with different number of clusters, K -means was applied using random subspace strategy by changing the value of K . Finally, the consensus algorithm was applied over the base clustering solutions.

4.2 Experimental Results

In the first experiment, the accuracy of consensus solutions over 500 and 1000 samples produced by different methods was measured using the classical ARI metric [8]. Since no ground truth is available, internal validation of the solution was performed by comparing it to the base clustering solutions. The average similarity of the consensus solution with the base clustering solutions denotes the goodness of the method.

The performance of the proposed model was also compared with well-known classical methods like CSPA, HGPA, MCLA [15] and different variants of WEAC [7]. If a very limited number of ensemble algorithms automatically estimate the number of clusters, the method DiCLENS [12] was also used. Specifically, DiCLENS is the most appropriate for comparison purposes as it produces the number of cluster automatically, similarly to our proposed method. The consensus solution produced by each algorithm was compared with all the base clustering solutions in terms of ARI (Tables 1 and 2). The average similarity obtained is reported in the tables where the two best scores are highlighted. These demonstrate that the proposed method consistently performs good even though the number of clusters is not given as input.

In a second experiment, consensus clustering solutions were used for customer segmentation with the objective to optimize the selection strategy according to the diverse categories of users and their different needs within the flight recommendation selection optimizer of Amadeus. Flight selection depends on a quantity defined as a linear combination of different criteria, including the recommendation price and diverse convenience criteria of the flights. The linear combination of weights is

Table 1 Base clustering performance with 500 samples

| Algorithm | $K = 3$ | $K = 4$ | $K = 5$ | $K = 6$ | $K = 7$ |
|-----------|---------|---------|---------------|---------|---------|
| CSPA | 0.5919 | 0.5633 | 0.5539 | 0.6472 | 0.5388 |
| MCLA | 0.6117 | 0.7293 | 0.8218 | 0.7217 | 0.8066 |
| HGPA | 0.2176 | -0.0052 | 0.3447 | 0.2692 | 0.2388 |
| WEAC-SL | 0.3972 | 0.6654 | 0.8275 | 0.8056 | 0.7924 |
| WEAC-AL | 0.3637 | 0.5964 | 0.8275 | 0.8066 | 0.7917 |
| WEAC-CL | 0.6001 | 0.6654 | 0.8275 | 0.8149 | 0.8002 |
| GP-MGLA | 0.6001 | 0.6939 | 0.8275 | 0.7240 | 0.6995 |
| DiCLENS | - | - | 0.8275 | - | - |
| Proposed | - | - | 0.8275 | - | - |

There are ten base clustering solutions. Six among them contain five clusters, and four solutions comprise three, four, six and seven number of clusters

Table 2 Base clustering performance with 1000 samples

| Algorithm | $K = 5$ | $K = 6$ | $K = 7$ | $K = 8$ | $K = 9$ |
|-----------|---------|---------|---------------|---------------|---------|
| CSPA | 0.5376 | 0.7162 | 0.7044 | 0.6201 | 0.5814 |
| MCLA | 0.6139 | 0.7822 | 0.8173 | 0.8139 | 0.7455 |
| HGPA | 0.2010 | 0.3302 | 0.4642 | -0.0048 | -0.0049 |
| WEAC-SL | 0.6188 | 0.7490 | 0.8177 | 0.8140 | 0.8020 |
| WEAC-AL | 0.5507 | 0.7490 | 0.8177 | 0.8166 | 0.8043 |
| WEAC-CL | 0.7184 | 0.7490 | 0.8177 | 0.8166 | 0.7964 |
| GP-MGLA | 0.7184 | 0.7583 | 0.8177 | 0.7975 | 0.7788 |
| DiCLENS | 0.7183 | - | - | - | - |
| Proposed | - | - | - | 0.8177 | - |

There are ten base clustering solutions. Six among them contain seven clusters, and the four solutions comprise three, four, five and six clusters

optimized to maximize the booking probability of the returned recommendations. This booking probability is estimated using a customer choice model [10], and the mapping function previously described is necessary to assign a new customer to a particular cluster.

Results presented in Table 3 were conducted on the set of base clustering solutions, along with the consensus solutions for 500 customers, to perform the optimization process. The clusters were then evaluated according to a business metric used by Amadeus: the relative difference between the sum of all the booking probabilities of flight recommendations returned by the optimized solution and a reference solution. The reference solution is defined by setting all weights to zero except for the recommendation price, and it corresponds to the default configuration of the flight search engine. The metric indicates to which extent the optimized solution improves the attractiveness of the recommendations returned by the search engine. The reported percentage represents how much the proposed method improves the internal objective

Table 3 Performance measure on booking probability improvement in terms of Amadeus business metric

| Algorithm | $K = 3$ (%) | $K = 5$ (%) |
|------------------|-------------|-----------------------------------|
| Base clusterings | 49 | 24.4, 21.6, 12.2, 4.4, 21.5, 18.6 |
| CSPA | 29.7 | 27.6 |
| MCLA | 22.6 | 12.7 |
| HGPA | 31.3 | 31.9 |
| WEAC-SL | 6.9 | 13.2 |
| WEAC-AL | 22.7 | 19.8 |
| WEAC-CL | 20.8 | 32.2 |
| GP-MGLA | 16.8 | 30.0 |
| DiCLENS | — | 28.6 |
| Proposed | — | 23.6 |

Results are presented for $K = 3$ and 5. As described in Table 1, six base clusterings out of ten have five number of clusters so multiple values appear for $K = 5$

function used to select the flight recommendations in the flight search engine. While there is not a direct link between improvement and conversion rate, this percentage represents a surrogate measure to it: the difference of conversion rate induced by the new configuration. DiCLENS which also finds automatic clusters produces better result in terms of Amadeus metric, however in terms of ARI proposed method performs better in majority situations. Therefore, we need to choose a reliable algorithm showing acceptable results in terms of both ARI and business metric.

5 Conclusions and Discussion

We have presented a consensus clustering method which gives a better average improvement than most base clustering solutions. It is a good compromise among consensus solutions as it constantly gives good ARI values, and, for the problem of flight selection, its associated business metric is above the median of all other consensus methods. This shows that it is more reliable to depend upon multiple diverse clustering solutions and an appropriate consensus generation process. Additionally, this method saves time compared to the current process implemented at Amadeus where N base clustering solutions are compared depending on the outcome of the optimization process. Here, the processing time is divided by N . Being the optimization process of the current bottleneck, this is an important feature. As a future direction, we intend to study how the other objective functions can be deployed for deriving a good quality solution aiming to improve booking probability.

References

1. T. Alqurashi, W. Wang, Clustering ensemble method. *Int. J. Mach. Learn. Cybern.* **10**, 1227–1246 (2018)
2. H.G. Ayad, M.S. Kamel, Cumulative voting consensus method for partitions with variable number of clusters. *IEEE TPAMI* **30**(1), 160–173 (2008)
3. S. Chatterjee, N. Pasquier, A. Mukhopadhyay, Multi-objective clustering ensemble for varying number of clusters, in *Proceedings of KARE Workshop in 14th International Conference on SITIS* (IEEE, 2018), pp. 387–395
4. S. Chatterjee, N. Pasquier, S. Nanty, M. Zuluaga, Multi-objective consensus clustering framework for flight search recommendation. [arXiv:2002.10241](https://arxiv.org/abs/2002.10241) (2020)
5. K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002)
6. A. Fred, A. Jain, Evidence accumulation clustering based on the K-means algorithm, in *Structural, Syntactic, and Statistical Pattern Recognition*, LNCS, vol. 2396 (Springer, Berlin, 2002), pp. 442–451
7. D. Huang, J.H. Lai, C.D. Wang, Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. *Neurocomputing* **170**(C) (2015)
8. L. Hubert, P. Arabie, Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
9. A.K. Jain, R.C. Dubes, P.J. Flynn, Data clustering: a review. *ACM Comput. Surv.* **31**(3), 264–323 (1999)
10. A. Lhéritier, M. Bocamazo, T. Delahaye, R. Acuna-Agost, Airline itinerary choice modeling using machine learning. *J. Choice Model.* (2018)
11. S. Liu, Q. Kang, J. An, M.C. Zhou, A weight-incorporated similarity-based clustering ensemble method, in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control* (2014), pp. 719–724
12. S. Mimaroglu, E. Aksehirli, Diclens: divisive clustering ensemble with automatic cluster number. *IEEE/ACM TCBB* **99**(2), 408–420 (2011)
13. A. Mottini, A. Lhéritier, R. Acuna-Agost, M. Zuluaga, Understanding customer choices to improve recommendations in the air travel industry, in *Proceedings of the Workshop on Recommenders in Tourism (RecTour 2018), held in Conjunction with the 12th ACM Conference on Recommender Systems (RecSys 2018)* (2018), pp. 28–32
14. N. Nguyen, R. Caruana, Consensus clusterings, in *Seventh IEEE International Conference on Data Mining (ICDM 2007)* (2007), pp. 607–612
15. A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining partitionings. *J. Mach. Learn. Res.* **3**, 583–617 (2002)
16. C. Zhong, X. Yue, Z. Zhang, J. Lei, A clustering ensemble: Two-level-refined co-association matrix with path-based transformation. *Pattern Recognition* **48**(8), 2699–2709 (2015)

Profiling JVM for AI Applications Using Deep Learning Libraries



Neha Kumari and Rajeev Kumar

Abstract In contemporary times, artificial intelligence (AI) techniques are widely used in development of software applications. Most of the AI-based applications are designed to perform complex tasks such as medical diagnosis, data analytic, human assistants, etc. The performance of such applications depends on development environment. Virtual machines are commonly being used for such development environments. These act as middlewares that support necessary tools for execution of applications. Java Virtual Machine (JVM) is one of the popular virtual environment which is used for several applications. It provides a secure, flexible, and fast execution environment. Therefore, we choose JVM to explore its suitability for AI applications. In this paper, we analyze JVM performance for different AI applications which include deep learning libraries. We use a profiling tool visualVM which profiles JVM performance for running applications. Our goal is to explore key strengths of JVM for AI applications. This in-depth analysis of JVM may help the developer community to choose an appropriate environment for AI applications development.

Keywords Deep learning · Virtual machine · Image classifier · Image drawer · Tic-tac-toe

1 Introduction

Artificial intelligence (AI) is all about machine performing intelligently [1]. AI is a broader domain that deals with the simulation of human intelligence into a machine. There are various ways to make a machine intelligent. One of the examples is the ability to learn from experience and perform future tasks based on those experiences.

N. Kumari (✉) · R. Kumar
School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi 110067, India
e-mail: nkumari.cse@gmail.com

R. Kumar
e-mail: rajeevkumar.cse@gmail.com

Such learning is known as machine learning [2]. The machine learning (ML) technique enables the system to make their own pattern for decision-making. A more specific form of machine learning is deep learning. The deep learning technique follows the multilayer structure of a neural network [3, 4]. Each layer of a neural network is tangled in such a manner that the output of one layer is used for the input of another layer. The deep learning technique uses a backpropagation algorithm for feature extraction. The backpropagation algorithm changes internal parameters and performs automatic feature extraction from raw data. The software applications which are based on these learning algorithms consume large memory and CPU. Such applications require a development environment that has high computational power, well management of memory, and secure execution. Apart from this, the development environment must be compatible with various kinds of AI libraries and tools.

It is difficult to combine all required software development tools together on a single interface. Virtual machines help to unify a variety of processes together. Virtual machines simulate different physical and non-physical machines on a single platform. Virtualization of physical machines is known as hardware virtualization and non-physical as software virtualization. Hypervisors are examples of hardware virtualization, and Java Virtual Machine (JVM) and Common Language Runtime (CLR) are two examples of software virtualization. In this paper, we focus on software virtualization or software process virtual machines. Such virtual machines provide platform-independence; this implies the execution of an application is free of system configuration. The process virtual machines support services as linking of libraries and optimized execution of code. JVM provides such an execution environment for Java and other Java influenced programming languages. Nowadays, JVM-based programming languages are being used for AI application development. In this paper, we profile JVM for three different AI applications; an image drawer, an image classifier, and a tic-tac-toe game. The analysis of JVM for these AI applications highlights JVM suitability and its efficiency for AI system development.

2 Enabling technology

2.1 Deep learning

Deep learning is a sub-branch of machine learning. In this, learning is based on multiple layers of feature extraction to model a complex structure. Each depth of the layer leads to more abstraction and precise results. This hierarchical approach of machine learning is more powerful and flexible than conventional machine learning techniques. As the amount of data increases, the performance of conventional machine learning saturates, whereas the performance of deep learning enhances. In this paper, we have used deep learning libraries to perform AI applications [5]. Our

purpose is to analyze JVM performance for a variety of AI applications. The deep learning-based applications require high computation power. Therefore, the selection of deep learning based applications is worthy of the analysis of JVM.

2.2 *AI libraries for JVM*

For the last several years, developers have shown their huge participation in the development of AI libraries for JVM and JVM based languages. Some of the libraries are similar to pre-exist AI libraries of other programming languages. For example, Java's ND4J works similarly to Python's NumPy. However, many other libraries are more advanced and powerful than exiting libraries of other programming languages. In this paper, we use three different AI applications that use a variety of machine and deep learning libraries of JVM such as *deeplearning4j*, *slf4j*, *datavec* etc.

2.3 *Java Virtual Machine*

JVM is an abstract form of computer which interprets intermediate languages (byte-codes) into machine understandable instructions [6]. JVM consists of three subsystems: a loader, a data area, and an execution engine. It works as an independent system, it features a reliable, expandable, and fast application development environment. This simplifies the development of large-scale projects like WEKA, Hadoop, Spark and more. Currently, there are many open-source projects that work for the support of AI, ML, and Big Data. For example, Project Panama aims to enrich the connections between JVM and APIs of other programming languages; hence, it improves JVM capacity [7]. Project Valhalla works for the development of a new type system; hence, it manages huge and versatile data [8]. Project Z Garbage collector focuses on the enhancement of overall JVM performance by reducing pause time for multi-terabyte of Heap [9].

2.4 *VisualVM*

VisualVM is an open-source tool. It is used for the performance monitoring of JVM. It can be easily found in JDK's bin directory (JDK version 1.6 onwards). VisualVM is a graphical tool; it represents performance detail in the form of a graph. Using this tool, it is easy to observe the overall performance of JVM for the running application.

3 Parameters description

There are many factors on which the performance of JVM depends [10], for example, CPU performance, cache misses, memory leaks, garbage collector (GC) workloads, etc. Here, we use three performance parameters to profile JVM; CPU usage, memory usage, and GC performance.

3.1 CPU usage

The portion of the CPU that is engaging for JVM execution is one of the important parameters to analyze JVM performance. CPU usage depends on several reasons such as total active threads, GC cycles, and infinite loops in code. A higher number of active threads imply larger load on the CPU. The amount of CPU utilization varies among threads. At times, due to repeated GC cycles and infinite loop threads, CPU consumption reaches to its extreme, and this creates bottleneck like situation. Such problems can be solved by analyzing threads and garbage collection in JVM. If a single application solely uses half and larger portion of CPU, than the application must be analyzed to detect complications.

3.2 Memory usage

It is important to analyze memory usage for running applications. This helps to find a majority of memory issues that affect the overall performance of JVM. For example, memory leaks, class-loader leaks, memory shortage, and performance of garbage collector. In JVM, the memory area is divided into two groups; heap memory area and non-heap memory. Heap area is further divided into young and old area. New objects are allotted in the young area. As the allotted objects of the young area are no longer in use, then it shifts to the old area. Garbage collector cleans unused data from old area and makes space for new objects allocation. Otherwise, it can cause a memory leak.

3.3 Garbage collector performance

Garbage collector releases those objects from heap memory which are no longer in use. This automatic memory release process is based on two steps, namely *Mark* and *Sweep*. It marks the unused objects of the heap and then sweeps it out after some cycle. The main factors affecting GC performances are total available heap memory and GC pause time.

4 Detailed analysis

We choose three different AI applications for our analysis [11]: Image drawer using neural network system [12], image classifier using the nearest neighbor algorithm, and a board game tic-tac-toe using minimax algorithm. The following sub-sections contain detailed performance analysis of the mentioned applications.

4.1 *AI applications*

4.1.1 Image drawer

Image drawer application creates a neural network using external inputs and then it is used for redrawing the image. The trained neural network draws the complete image in countless iterations. It has been observed that the network draws rough outline of the given image quickly but it has taken a huge amount of time (around 10h as per system configuration) for tuning of the image.

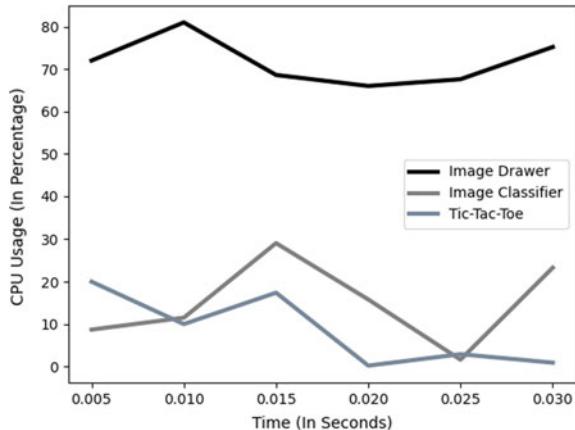
4.1.2 Image classifier

This application classifies images using the “nearest neighbor algorithm.” The nearest neighbor algorithm searches for a predefined class that is close to the unclassified sample data. Based on the minimum distance between sample data and predefined classes, the algorithm predicts class for the unclassified data. In this application, we use Manhattan and Euclidean distance measures. This image classification application has used cifar-10 dataset. The dataset contains 60,000 images; each image is classified into 10 mutual exclusive classes. Among 60,000 image data, 50,000 are being used for training and the rest 10,000 images for testing purposes. The nearest neighbor classifier will take a test image, compare it to every single one of the training images, and predict the label of the closest training image.

4.1.3 Tic-tac-toe

This application has used the minimax algorithm to train a player. The algorithm aims to minimize loss by applying all possible moves of the game against the opponent.

We have used the following system configuration in this work: RAM (4.00 GB), Processor (Intel Core i7-4790 CPU @ 3.60GHz), System type (64-bit OS), JAVA SE 8, JVisualVM (1.8.0 version).

Fig. 1 CPU usage

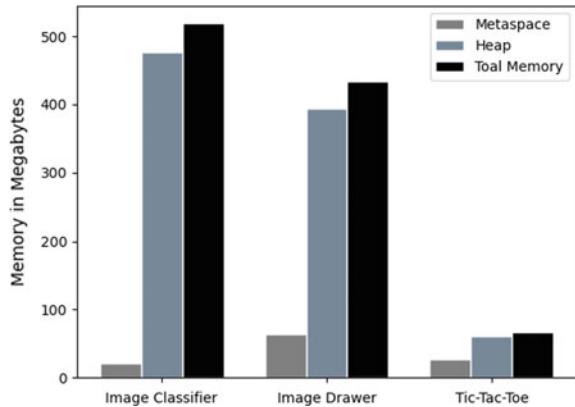
4.2 CPU usage

In a stand-alone system, if a single application uses more than 50% of CPU time, it signals the presence of complications. These complications can be inside the application code or inside execution environment. Here, we focus on the later part. In our analysis, we found one application out of three applications has used higher CPU time. As shown in Fig. 1, the average CPU time used by the image drawer application is about 80% of total CPU time and the application has taken hours to execute completely. The other two applications have used the ideal portion of CPU and completely executed in a few minutes. The reasons which we observed for higher CPU use after monitoring of applications are as follows:

1. Many active threads with maximal use of CPU.
2. Higher garbage collector activity due to higher demand for CPU.
3. Higher loop cycles in code. This implies a lack of loop optimization techniques in a runtime environment.

4.3 Memory usage

In our analysis, we observe both heap and metaspace memory footprint for memory-related issues such as out of memory error. The out of memory error occurs when there is no more space in memory to be allotted for new objects. The new object has to wait until it gets allotted in the memory area. This slows the performance of an application, and sometimes, it results in abrupt failure of the application. Through visualVM profiling tool, we observe heap and metaspace allotments at random intervals. We observe the occupied memory area before processing the garbage collector and also after processing the garbage collector. There are huge differences in mem-

Fig. 2 Memory usage

ory requirement of all three applications but we did not find any memory error. For a larger application, after a major garbage cycle, the heap performs normally. Figure 2 shows the average memory usage of applications. None of the histograms shows heap usage crosses the total allotted memory area. Therefore, we observe the following:

1. No out of memory error.
2. Major pauses of garbage collector for large data applications.

4.4 GC performance

The garbage collector is directly related to memory action. As the young space gets full, the GC performs minor pause, and when old space gets full, it performs major pause. The rate on which objects get allotted in young space is allocation rate and the transfer rate from young space to old is known as the promotional rate. Here, we use the term survivor rate for the objects who survive in young space after a minor pause. Major pause affects the overall performance of JVM. An increase in promotional rate implies an increase in major GC pause and an increase in allocation or survival rate implies an increase in minor pause.

Figure 3a shows the graph of minor and major pauses during a period of application execution and Figure 3b shows the histogram of their promotional and survivor rate. As shown in Fig. 3a, the image classifier graph has several major pauses. The same is reflected in Fig. 3b through a higher promotional rate than the survivor rate. Therefore, we conclude the following:

1. A high promotion rate encourages high major GC pauses.

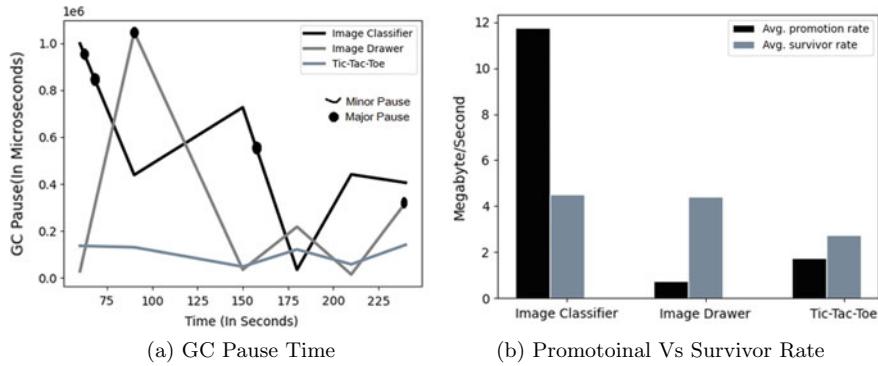


Fig. 3 Garbage collector performance

5 Comparative analysis

JVM-based languages are in use. There are hundreds of thousands of applications and tools which have been developed using these technologies. However, these JVM-based languages are not very popular in AI and ML domains. Some of the possible reasons are: complex programming environments and lesser availability of AI libraries. In the past several years, many AI libraries have been developed exclusively for JVM-based languages. JVM is also enhanced to work flawlessly for complex and huge AI applications. In this section, we compare three popular programming environments such as Java Virtual Machine (JVM), Common Language Runtime (CLR) and Python Virtual Machine (PVM). We choose five important metrics of the virtual machines to compare the efficiency and suitability for AI applications. We consider the following comparative metrics: (i) performance of garbage collectors during the execution of an application, (ii) overall execution speed of the virtual machine, (iii) data security, (iv) compatible environment for external tools, and (v) support of AI libraries. In Table 1, we rank virtual machine property on a three-point scale, namely average, good and best.

On comparing the existing garbage collectors of above mentioned VMs, the newly introduced ZGC of JVM performs best. ZGC is scalable and of low latency. There-

Table 1 Comparison of popular programming environments

| VM metrics | JVM | CLR | PVM |
|------------------------|------|---------|---------|
| GC performance | Good | Average | Average |
| VM performance | Best | Good | Average |
| Execution level safety | Good | Best | Average |
| Compatibility | Best | Average | Good |
| AI library support | Good | Average | Best |

fore, it works properly for huge data applications as compared to CLR and Python VM. The overall performance of a virtual machine depends on factors such as memory usage, throughput, and latency. According to this, JVM is fastest among others due to the good management of memory and garbage collector. Security mechanisms are approx similar in all three virtual machines. However, the cleaner design of CLR and no backward compatibility make it more security capable than JVM [13]. CLR and PVM are more centric to the specified language; therefore, they are not compatible with numerous tools in comparison to JVM. JVM is compatible with robust error resolution and production monitoring tools. In terms of AI support, PVM has a larger collection of libraries for different AI techniques. The enhanced popularity of Java in the field of AI application development has added many new libraries such as Java-ML, MLeap, and RapidMiner.

6 Conclusion

Most of the AI applications are complex and data-centric. They require high computational power and well management of data to develop such applications. The programming development environment plays a vital role in overall performance of applications. Java and other similar programming languages have JVM as a typical development environment. We have analyzed JVM capability in terms of CPU uses, memory management, and performance speed. However, We have limited our analysis only for few deep learning libraries. The performance analysis of JVM has clearly shown its suitability for AI applications. This in-depth analysis of JVM performance is expected to benefit the developer community in selecting an appropriate programming environment for AI application development.

References

1. S.J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach* (Pearson Education Limited, Malaysia, 2016)
2. D. Michie, D.J. Spiegelhalter, C.C. Taylor, *Machine Learning, Neural and Statistical Classification* (Ellis Horwood, USA, 1995)
3. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>
4. A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial. *Computer* **29**(3), 31–44 (1996). <https://doi.org/10.1109/2.485891>
5. B. Quinto, Deep learning, in *Next-Generation Machine Learning with Spark* (Apress, Berkeley, CA, 2020), pp. 289–348
6. B. Venners, *The Java Virtual Machine* (McGraw-Hill, New York, 1998)
7. OpenJDK, Project Panama Homepage. <https://openjdk.java.net/projects/panama/>
8. OpenJDK, Project Valhalla Homepage. <https://openjdk.java.net/projects/valhalla/>
9. OpenJDK, Project ZGC Homepage. <https://openjdk.java.net/projects/zgc/>

10. D. Gu, C. Verbrugge, E.M. Gagnon, Relative factors in performance analysis of Java virtual machines, in *Proceedings of 2nd International Conference on Virtual Execution Environments* (2006), pp. 111–121. <https://doi.org/10.1145/1134760.1134776>
11. Deeplearning4j Examples on Github. <https://github.com/deeplearning4j/dl4j-examples>
12. K. Gregor, I. Danihelka, A. Graves, D.J. Rezende, D. Wierstra, Draw: a recurrent neural network for image generation. arXiv preprint [arXiv:1502.04623](https://arxiv.org/abs/1502.04623) (2015)
13. N. Paul, D. Evans, Comparing Java and .NET security: lessons learned and missed. *Comput. Secur.* **25**(5), 338–350 (2006). <https://doi.org/10.1016/j.cose.2006.02.003>

Offline Signature Recognition Using Deep Features



Kamlesh Kumari and Sanjeev Rana

Abstract Offline Signature Verification plays a significant role in Forensic offices. In our research, we investigate the performance of signature recognition utilizing features based on AlexNet (pretrained Convolution Neural Network model). All the investigations are performed on scanned signatures of fourteen users each from three datasets, namely CEDAR, UTSig, and BHsig260. Two classifiers, i.e., Support Vector Machine (SVM) and Decision Tree are utilized. Utilizing features based on Deep Convolution neural network and SVM as machine learning algorithms show better outcomes. The best output is achieved for Bengali signature recognition utilizing SVM with 100% accuracy. For Persian signature, we obtained an accuracy of more than 80% for each user. Twelve users out of fourteen users for Hindi signature are 100% recognized using SVM.

Keywords Center of excellence for document analysis and recognition (CEDAR) · Decision tree (DT) · Convolution neural network (CNN) · ImageNet large scale visual recognition challenge (ILSVRC) · Support vector machine (SVM) · University of Tehran Persian signature (UTSig) · Signature verification competition (SVC)

1 Introduction

Offline Signature Verification (OSV) plays a significant role in Forensic offices. Fraud Survey report in [1, 2] shows that there are a lot of financial losses occurred due to forged signature in bank checks. This has inspired the research in OSV. The neuro-muscular system of a person is controlled by nerves of the brain. Nerves from the brain

K. Kumari (✉) · S. Rana

Department of Computer Science & Engineering, M. M (Deemed to Be University),
Mullana-Ambala, India

e-mail: savyakamlesh@gmail.com

S. Rana

e-mail: dr.sanjeevrana@mumullana.org

Table 1 Various architectures of CNN

| CNN | Number of layers | Runner up |
|-----------|------------------|---------------|
| AlexNet | 8 | 2012 |
| ZfNet | 7 | 2013 |
| GoogleNet | 22 | 2014 (first) |
| VGGNet | 19 | 2014 (second) |
| ResNet | 152 | 2015 |

guide a person's hand to generate a signature. Signature recognition includes perspectives from disciplines going from human life structures to designing, from neuroscience to software engineering. Extracting features from offline signature using traditional image processing algorithms include methods such as statistical feature, geometric feature, shape-based, wavelet-based, etc. In recent years, Deep Convolution neural network based features show the best result for image classification. So in our research work, we investigate the performance of Signature Identification and Verification utilizing features based on AlexNet.

AlexNet is the name of a Convolution Neural Network model structured by Alex Krizhevsky [3]. Alex Krizhevsky is a computer scientist. In 2012, he won the competition called ImageNet Large Scale Visual Recognition Challenge (ILSVRC). ImageNet is a database of over fifteen million images belonging to roughly 22,000 categories. AlexNet consists of eight layers. There are various architecture of CNN like AlexNet, i.e., GoogleNet [4], VGGNet [5], ResNet [6], DenseNet [7], ZfNet [8]. Table 1 shows the Number of layers and year of competition (won) for various CNN.

The goal of our research is mainly to investigate the performance of handwritten signature identification and verification utilizing features based on AlexNet on four diverse language signatures namely Hindi, Persian, English, and Bangla.

The paper is organized into seven segments as follows: Sect. 1 describes the Introduction, Sect. 2 describes the CNN, Sect. 3 depicts Literature Review, Sect. 4 presents the Methodology, Sect. 5 portrays Experimental results, Sect. 6 presents Comparison with Bag of Features and Sect. 7 presents the Conclusion.

2 Convolution Neural Network

In 1995, the idea of Convolution Neural Network was introduced by Yann LeCun and Yoshua Bengio [9]. Convolution Neural Network belongs to a special group of neural networks. CNN learns image features automatically. Neurons are fundamentally the pixel strength of an input image in the context of image recognition. It is impractical to attach each neuron to all neurons, instead, we connect each neuron to only an area of input image. The receptive field of a neuron in a CNN refers to the part of the image that is visible to one filter at a time. Downsampling operations aggregate information in each part of the image and Parameter sharing concept in CNN means a feature detector that is beneficial in one part of image is perhaps also useful in another part

of the image. Local receptive fields, downsampling operations, and weight sharing properties are the important properties of CNN for pattern recognition. CNN is invariant to rotation, scaling, translation, distortion, squeezing. CNN uses ReLU activation function. CNN has the following three layers:

- (A) The Convolution Layer: Convolution layer is the primary layer on CNN. This layer should have input and kernel for convolute with input. First three elements of kernel are applied to first three elements of input image by calculating dot product, for example, $-4 * 1 + 0 * 1 * -1 + 1 * 1 + 6 * 0 + 5 * -1 + 2$ for the following data as shown in Fig. 1. In this way, we apply kernel to whole input image. After the convolution, we will get output of $4 * 4$ sizes.
- (B) The Pooling Layer: To diminish the size of information in the image, pooling layer is utilized. By reducing the image size, pooling layer speed up the computation. Consider a $6 * 6$ matrix as shown below in Table 2. For every consecutive $3 * 3$ block; we take max value in max pooling as shown in Table 3.
- (C) Fully Connected Layer (FC): In this layer, every neuron accepts input from every element of the previous layer. There can be more than one fully connected layer in the network To produce yield from CNN we need FC layer.

| | | | | | |
|---|---|---|---|---|---|
| 4 | 0 | 1 | 2 | 7 | 4 |
| 1 | 6 | 5 | 3 | 9 | 1 |
| 2 | 7 | 2 | 1 | 1 | 3 |
| 0 | 1 | 3 | 1 | 7 | 8 |
| 4 | 2 | 1 | 6 | 2 | 8 |
| 2 | 4 | 4 | 2 | 3 | 9 |

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

Fig. 1 Convolution between input image and kernel

Table 2 Input

| | | | | | |
|----|----|----|----|----|----|
| 12 | 1 | 39 | 5 | 9 | 14 |
| 5 | 19 | 3 | 1 | 2 | 24 |
| 13 | 2 | 23 | 26 | 14 | 4 |
| 4 | 19 | 2 | 4 | 2 | 13 |
| 14 | 1 | 26 | 29 | 15 | 5 |
| 20 | 18 | 3 | 3 | 12 | 11 |

Table 3 Max pooling

| | |
|----|----|
| 39 | 26 |
| 26 | 29 |

3 Literature Review

Accuracy of 87.5% is obtained in work reported in [10] for Chinese signatures. Feature extraction is based on pretrained CNN model. The model used in the work is AlexNet. Signature image is divided into equal sub-images for feature extraction in the work reported in [11]. Two local binary patterns are extracted for each sub-image and centroid of two LBP is calculated. The final feature vector of a signature image contains feature-length equal to twice of number of sub-images of signature image. Database used in signature verification competition in 2009 is used for research purposes. Accuracy of 94.8% is achieved with this method.

The work proposed in [12] uses signature of Uyghur type and used the Bag of Visual Word model for feature extraction. Performance of this method is compared with CEDAR database. Accuracy of 93.81% is obtained in Uyghur signature verification with six hundred visual words. With four hundred visual words, accuracy of 95.8% is obtained in CEDAR dataset. Delaunay Triangulation concept is used for feature extraction in the [13] and it obtained a false rejection rate of ten. Endpoint and the intersection point of a signature image are used in the work. These points are extracted in a way that the points represent the Delaunay triangle.

One hundred features are used in research work proposed in [14]. These are grid and global features. User-specific features are selected for each user and fuzzy-based clustering approach is used. Equal error rate of 7.66 with fourteen training samples is obtained for CEDAR. Equal error rate of 13.67 with five training is obtained for MCYT-75. Two types of features are extracted in [15] that are reference features and parametric features. Reference features depend on the vertical and horizontal projection of image. Results show that reference features give better performance than parametric features. Fuzzy clustering approach and moment-based features are used in the work [16]. Neural network approach is used in [17] and the numbers of features used are equal to the size of the image. Number of input nodes in the system is 8192.

In the work proposed in [18], the database of disguised signature is used. Local features used in their work are SURF, FAST and FREAK. SURF is Speed up Robust Features. FAST is Features from Accelerated Segment. FREAK is Fast Retina Key points. Besides it, Bunke features are also used. GMM classification method is used. Equal error rate of thirty is achieved using combination of FREAK and FAST. EER obtained using Bunke features is twenty.

Deep neural networks are used in the work proposed by [19]. The structure of the network used is based on VGG-16. Four types of signature are used in work for experimentation. Dutch and Japanese signature from ICDAR dataset, signature from MCYT, and signature from GPDSsyntheticSignature. Euclidean distance is utilized to look at the distance between two signatures. They named their approach as signature Embedding. Result shows that the accuracy of 93.39% is achieved for Japanese signature verification which is higher compared to other signatures. Log-likelihood ratio as a metric is also used in the work.

Neural network as a classifier is utilized in the work proposed in [20]. In research work, they used features based on the shape of signature and database of 3528 signatures are used for experimentation. Correct classification rate of 90% is obtained. The resolution of signature used is 96 dots per inch [21]. Global and Grid Features are combined and the backpropagation neural network used as a classifier obtained better results than KNN. GCLM. The region properties based features are used in [22]. Experiments are performed with different kernel functions of SVM. Two types of optimization are used, i.e., SMO and Least Square with a different kernel function. Best results are obtained using rbf with SMO.

SVC named as ICDAR 2015 held in 2015 was jointly organized by Forensic Handwriting Expert and Researchers [23]. The result of SVC is described in the paper. In SVC, two kinds of offline signature are utilized, i.e., Italian and Bengali. Also, online signatures of the German language are used for SVC. All the verification results of signature are described in terms of log-likelihood ratio. Signature image of Bengali is in 300 dpi and in Tagged Image file format. The Signature of Italian scripts are collected from employees of the University of Salerno by filling various applications and forms. In SVC, participants are from three universities namely Tebessa, Sabanci, and Qatar. Also, one commercial organization participated in Competition. Features used in SVC are LBP, HOG, edge-hinge, multi-scale run length, geometric features, and morphical features of image.

In the work [24], six features are used for signature verification. These features are based on center of gravity, spanning tree, Delaunay Triangulation, and hand pressure. The result shows that features based on Delaunay Triangulation result in better performance.

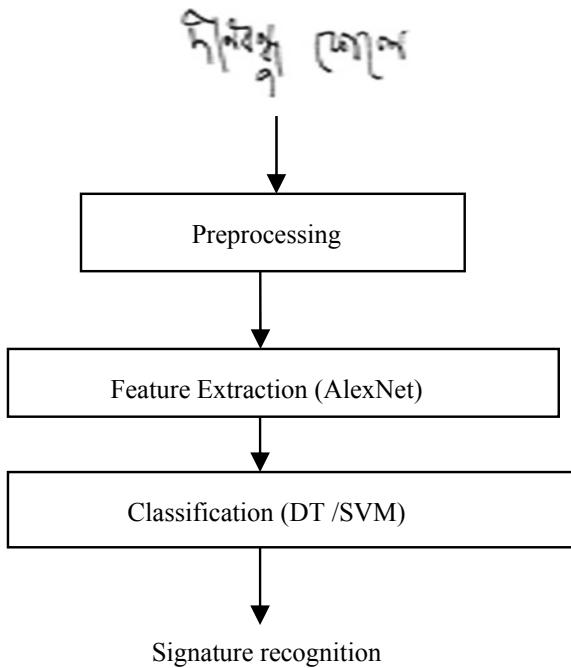
In the work [25] six classifiers are used for classification, a majority vote rule is used for the result. Features used are based on the polar coordinates of signature image and discrete random transform. The result shows improvement using an ensemble method instead of single classifier.

Four features based on the difference of signature pair are used in the work proposed by Kumari and Rana [26, 27]. Signatures of 55 users are used for research purposes. Best results are obtained by using SVM. For writer-indendepent model, an accuracy of 72% and 80% is achieved for Bengali and Hindi signature respectively. For the English signature, accuracy is better than these for writer-independent model. Empirical Evaluation of Persian signature verification is also investigated in the research work [28]. The results obtained by using four features in work are average.

4 Methodology

Procedure of Signature Recognition utilized in our exploration is shown in Fig. 2.

Fig. 2 Process of signature identification and verification



4.1 Preprocessing

AlexNet recognizes input size of $227 * 227 * 3$, so all the scanned signatures of dataset are resized to $227 * 227$ and converted from gray scale image to RGB image.

5 AlexNet as Feature Extractor

Each Pretrained CNN has its architecture, i.e., Number of layers used and size of Kernel. The architecture of AlexNet is shown in Table 4. The total number of parameters in CNN depends on the weight and biases.

$$\text{Number of Weights (W)} = \text{Number of Kernel} \times \text{Width of Kernel} \times \text{Number of Channels}$$

$$\text{Number of Biases (B)} = \text{Number of Kernel}$$

$$\text{Total Parameter} = W + B$$

In our research work, we extracted the features from FC layer of AlexNet. The length of feature vector is 4096.

Table 4 Architecture of AlexNet [3]

| Layer | Number of filter | Image size | Kernel size | Stride (St)/padding (Pa) | Parameter |
|--------------------|------------------|---------------|-------------|--------------------------|------------|
| Input | | 227 * 227 * 3 | | | |
| Conv layer | 96 | 55 * 55 | 11 * 11 | St = 4/Pa = 0 | 34,944 |
| Pool layer (3 * 3) | | 27 * 27 | | St = 2 | |
| Conv layer | 256 | 27 * 27 | 5 * 5 | St = 1/Pa = 2 | 614,656 |
| Pool layer (3 * 3) | | 13 * 13 | | St=2 | |
| Conv layer | 384 | 13 * 13 | 3 * 3 | St = 1/Pa = 1 | 885,120 |
| Conv layer | 384 | 13 * 13 | 3 * 3 | St = 1/Pa = 1 | 1,327,488 |
| Conv layer | 256 | 13 * 13 | 3 * 3 | St = 1/Pa = 1 | 884,992 |
| Pool layer | | 6 * 6 | | St = 2 | |
| FC | | 4096 | | | 37,752,832 |
| FC | | 4096 | | | 16,781,312 |
| FC | | 1000 | | | 4,097,000 |

5.1 Classifier

5.1.1 SVM

Decision boundaries concept is used in SVM. As shown in Fig. 3. There are two types of observations in training data labeled as red and blue. The observations that define the boundary are called support vectors. There are three support vectors (encircled) in

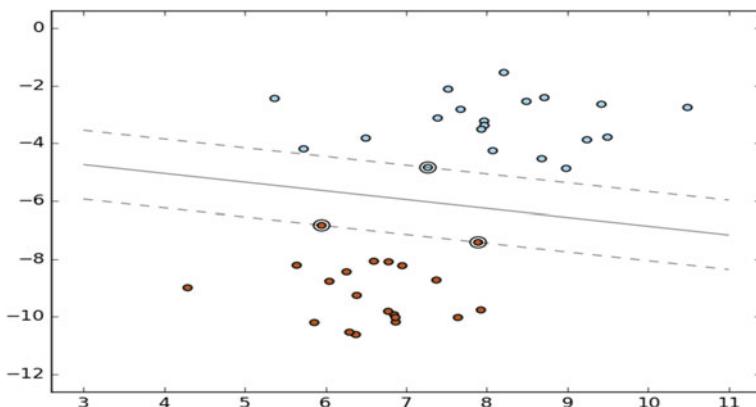
**Fig. 3** SVM

Fig. 3. If we remove the support vector, then it will change the decision boundaries. Finding the decision boundary is an optimization problem. The boundary can be linear or nonlinear. There are a lot of applications of SVM in regression and pattern recognition [29].

5.1.2 Decision Tree (DT)

DT is used for regression and classification purposes. Each node in the DT represents a feature and the outcome of node is represented by a branch. Final decision is made on the leaves of DT. Figures 4 and 5 show the DT of AND logic [30].

For classification, select the node (best feature) which best separates the training data: To select the best feature (node), information gain is used. Feature with high information gain is chosen as the best. The Information gain (IG) of each feature is calculated as:

Fig. 4 DT of AND logic

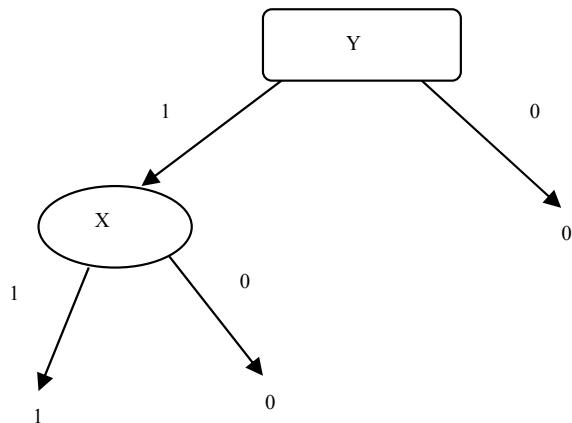
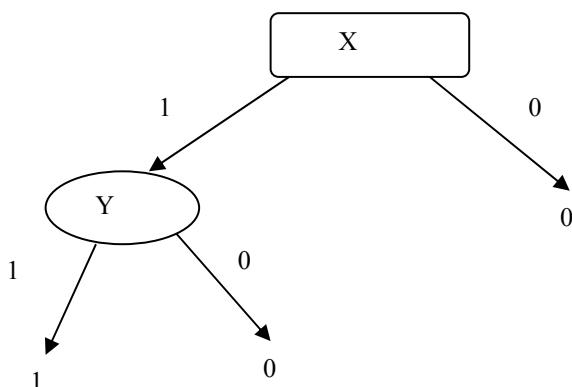


Fig. 5 DT of AND logic



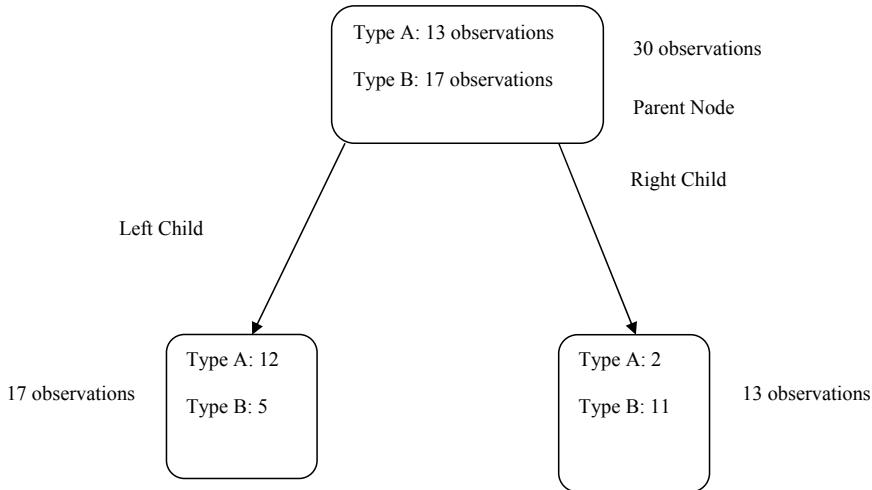


Fig. 6 DT with thirty observations

$$IG = E(\text{parent node}) - AE(\text{children}),$$

where E is entropy and AE is average entropy.

Entropy of training data is defined in terms of proportion of each type of observations in the training data and the probability of each type. Suppose there are 30 observations in training data as shown in Fig. 6 and there are two types of classes, i.e., Type A and Type B in the data, the entropy of training data or node is 0 if all observation belongs to the same class. Entropy is 1 if observations of both class types are equal in data. There are thirteen observations of type A and seventeen observations of type B. Then entropy will be between 0 and 1 because of unequal type of observations.

$$\text{Entropy}(T) = - \sum_i p \log_2 p_i$$

where p_i is the Probability of class.

With reference to the example,

$$E(T) = \sum_i \text{Proportion of Type A}_A \text{ Proportion of Type B}_B$$

Proportion of Type A = 13/30, Proportion of Type B = 17/30

For AE (children), we have to determine entropy of left child, entropy of right child and average of child node.

Entropy of left child = $-12/17 * \log_2 p - 5/17 * \log_2 p_B$

Entropy of right child = $-2/13 * \log_2 p - 11/13 * \log_2 p_B$.

| | | Confusion Matrix | | | | | | | | | | | | | | | | |
|--------------|------|------------------|--------------|--------------|----------------|--------------|----------------|--------------|----------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------|
| Output Class | P10F | P10F | P10G | P11F | P11G | P12F | P12G | P13F | P13G | P14F | P14G | P15F | P15G | P16F | P16G | P17F | P17G | |
| | | 11 8.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 8 6.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 0 0.0% | 11 8.3% | 2 1.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 6 4.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 1 0.8% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 91.7% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 5.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8.3% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 5.3 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 1 0.8% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 91.7% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 5.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8.3% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 10 7.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 6.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11.1% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8 6.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 0 0.0% | 0 0.0% | 0 0.0% | |
| | | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8 6.0% | 0 0.0% | |
| | | 100% 100% | 100% 0.0% | 100% 0.0% | 75.0% 25.0% | 100% 0.0% | 87.5% 12.5% | 100% 0.0% | 87.5% 12.5% | 90.9% 9.1% | 100% 0.0% | 96.2% |
| | | 100% 0.0% | 100% 0.0% | 100% 0.0% | 75.0% 0.0% | 100% 0.0% | 87.5% 0.0% | 100% 0.0% | 87.5% 0.0% | 90.9% 0.0% | 100% 0.0% | 3.8% |

Fig. 7 Confusion matrix of Persian signature

6 Experimental Results

All the execution is acted in Matlab [31]. The proportion of training data and testing data is 7:3.

Three datasets are used in our work CEDAR [32], UTSig [33], and BHsig260 [34]. Image format of scanned signature utilized in CEDAR is Portable Network Graphics. TIF image format is utilized in UTSig and BHsig260. We have used the signature of 14 users in our research. Experiments are performed for two different sets of users. First set consist of seven users from each dataset and second set consist of another seven users, i.e., user 8 to user 14 from each dataset. Confusion matrix is calculated for all the experiments in which SVM used as a classifier.

6.1 Persian Signature Recognition

Figure 8 shows the outcome of Persian signature recognition utilizing SVM for first set in the form of the confusion matrix. Figure 7 shows the Confusion matrix of Persian signature recognition utilizing SVM for the subsequent set. Accuracy of 94.7% for the first set and 96.2% for the subsequent set is obtained. Utilizing Decision Tree, the accuracy of Persian signature recognition is underneath 70%.

| Confusion Matrix | | | | | | | | | | | | | | |
|------------------|-----|------------|-----------|------------|-----------|------------|-----------|-----------|------------|-----------|------------|-----------|------------|-----------|
| Output Class | P1F | 10 7.5% | 2 1.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P1G | 1 0.8% | 6 4.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P2F | 0 0.0% | 0 0.0% | 10 7.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P2G | 0 0.0% | 0 0.0% | 1 0.8% | 8 6.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P3F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P3G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8 6.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P4F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 8.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P4G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8 6.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P5F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 10 7.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P5G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 8.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P6F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 0 0.0% | 0 0.0% | 0 0.0% |
| | P6G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 8 6.0% | 0 0.0% | 0 0.0% |
| | P7F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 11 8.3% | 2 1.5% |
| | P7G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 6 4.5% |
| 90.9% | | 75.0% | 90.9% | 100% | 100% | 100% | 100% | 100% | 90.9% | 100% | 100% | 100% | 100% | 75.0% |
| 9.1% | | 25.0% | 9.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 9.1% | 0.0% | 0.0% | 0.0% | 0.0% | 5.3% |

Fig. 8 Confusion matrix of Persian signature

6.2 English Signature Recognition

Utilizing DT, we acquired an accuracy of 65% for samples of seven users in the first set and 76% for the subsequent set. Figure 9 shows the Confusion matrix of English signature recognition utilizing SVM for first set with an accuracy of 92.9%. Also,

| Confusion Matrix | | | | | | | | | | | | | | |
|------------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Output Class | C1F | 7 7.1% | 1 1.0% | 0 0.0% |
| | C1G | 0 0.0% | 6 6.1% | 0 0.0% |
| | C2F | 0 0.0% | 0 0.0% | 6 6.1% | 1 1.0% | 0 0.0% |
| | C2G | 0 0.0% | 0 0.0% | 1 1.0% | 6 6.1% | 0 0.0% |
| | C3F | 0 0.0% | 0 0.0% | 0 0.0% | 7 7.1% | 0 0.0% |
| | C3G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 7.1% | 0 0.0% |
| | C4F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 7.1% | 0 0.0% |
| | C4G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 7.1% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% |
| | C5F | 0 0.0% | 6 6.1% | 0 0.0% | 1 1.0% | 1 1.0% | 0 0.0% | 0 0.0% |
| | C5G | 0 0.0% | 1 1.0% | 7 7.1% | 0 0.0% | 0 0.0% | 0 0.0% |
| | C6F | 0 0.0% | 6 6.1% | 1 1.0% | 0 0.0% | 0 0.0% |
| | C6G | 0 0.0% | 5 5.1% | 0 0.0% | 0 0.0% |
| | C7F | 0 0.0% | 7 7.1% | 0 0.0% | 0 0.0% |
| | C7G | 0 0.0% | 7 7.1% | 0 0.0% |
| 100% | | 85.7% | 85.7% | 85.7% | 100% | 100% | 100% | 100% | 85.7% | 100% | 100% | 85.7% | 71.4% | 100% |
| 0.0% | | 14.3% | 14.3% | 14.3% | 0.0% | 0.0% | 0.0% | 0.0% | 14.3% | 0.0% | 0.0% | 14.3% | 28.6% | 0.0% |

Fig. 9 Confusion matrix of English signature

| | | Confusion Matrix | | | | | | | | | | | | | | | | |
|--------------|------|------------------|------|------|------|-------|-------|------|------|------|------|-------|-------|------|------|------|-------|--|
| | | Output Class | | | | | | | | | | | | | | | | |
| | | C10F | C10G | C1FF | C1FG | C1GF | C12F | C12G | C1FF | C1FG | C1GF | C14F | C14G | C8F | C8G | C9F | C9G | |
| Output Class | C10F | 7.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| | C10G | 0.0% | 7.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| | C11F | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 1.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 12.5% | |
| | C11G | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| | C12F | 0.0% | 0.0% | 0.0% | 0.0% | 6.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| | C12G | 0.0% | 0.0% | 0.0% | 0.0% | 1.0% | 6.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 14.3% | |
| | C13F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 100% | |
| | C13G | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| | C14F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 1.0% | 0.0% | 0.0% | 0.0% | 0.0% | 12.5% | |
| | C14G | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| | C8F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 5.1% | 1.0% | 0.0% | 0.0% | 0.0% | 0.0% | 83.3% | |
| | C8G | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 1.0% | 6.1% | 0.0% | 0.0% | 0.0% | 0.0% | 14.3% | |
| | C9F | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 0.0% | 100% | |
| | C9G | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 7.1% | 0.0% | 0.0% | 100% | |
| | | 100% | 100% | 100% | 100% | 85.7% | 85.7% | 100% | 100% | 100% | 100% | 71.4% | 85.7% | 100% | 100% | 100% | 94.9% | |
| | | 0.0% | 0.0% | 0.0% | 0.0% | 14.3% | 14.3% | 0.0% | 0.0% | 0.0% | 0.0% | 29.6% | 14.3% | 0.0% | 0.0% | 0.0% | 5.1% | |

Fig. 10 Confusion matrix of English signature

Fig. 10 shows the Confusion matrix of English signature recognition utilizing SVM for second set with an accuracy of 94.9%.

6.3 Bengali Signature Recognition

Utilizing DT, we acquired accuracy of 86% for first set and 74% for the subsequent set. Figures 11 and 12 show the Confusion matrix of Bengali signature recognition utilizing SVM for first set and subsequent set, respectively. Accuracy of 100% is obtained in both cases.

6.4 Hindi Signature Recognition

Figures 13 and 14 show the outcome of Hindi signature recognition utilizing SVM for first set and another set respectively. Accuracy of 99.1% is obtained in both cases. Utilizing Decision Tree, accuracy of Hindi signature Recognition is 82% for first set and 77% for another set.

| Confusion Matrix | | | | | | | | | | | | | | | | |
|------------------|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Output Class | B1F | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B1G | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | B2F | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B2G | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | B3F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B3G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | B4F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B4G | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B5F | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B5G | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B6F | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B6G | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B7F | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B7G | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| 100% 0.0% | | 100% 0.0% | |

Fig. 11 Confusion matrix of Bengali signature

| Confusion Matrix | | | | | | | | | | | | | | | | |
|------------------|------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Output Class | B10F | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B10G | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | B11F | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B11G | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | B12F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B12G | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | B13F | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 5 4.3% | 0 0.0% | 100% 0.0% |
| | B13G | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B14F | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B14G | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B8F | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B8G | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | B9F | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | B9G | 0 0.0% | 7 6.3% | 0 0.0% |
| 100% 0.0% | | 100% 0.0% | |

Fig. 12 Confusion matrix of Bengali signature

7 Comparison with Bag of Features

Experiments are also performed for offline signature recognition using a bag of Features and SVM. Bag of features in context of handwritten signature image means to represent signature image as set of features in the same way as words in a document. In our research work, we used SURF (Speed up Robust Features) features, and 70%

| Confusion Matrix | | | | | | | | | | | | | | | |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|--------------|--------------|--------------|---------------|
| Output Class | H1F | H1G | H2F | H2G | H3F | H3G | H4F | H4G | H5F | H5G | H6F | H6G | H7F | H7G | |
| | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 8 7.1% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.9% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 87.5% 0.0% |
| | 0 0.0% | 0 0.9% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 87.5% 0.0% |
| | 0 0.0% | 0 0.9% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.9% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.9% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 88.9% 11.1% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% |
| 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 100% 0.0% | 99.1% 0.9% |

Fig. 13 Confusion matrix of Hindi signature

| Confusion Matrix | | | | | | | | | | | | | | | |
|------------------|--------------|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|
| Output Class | H10F | H10G | H11F | H11G | H12F | H12G | H13F | H13G | H14F | H14G | H15F | H15G | H16F | H16G | |
| | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 8 7.1% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 90.0% 0.0% |
| | 0 0.0% | 0 0.0% | 1 0.9% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 9 8.0% | 0 0.0% | 0 0.0% | 100% 0.0% |
| | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 7 6.3% | 0 0.0% | 100% 0.0% |
| 100% 0.0% | 100% 0.0% | 88.9% 11.1% | 100% 0.0% | 99.1% 0.9% |

Fig. 14 Confusion matrix of Hindi signature

strong features from each class are used. For Persian signature recognition, results are not good. For English, Hindi, and Bengali signature recognition, we obtained an average accuracy of 90% and above. Table 5 shows the comparison of Accuracy with pretrained CNN features. Results using features extracted from AlexNet are good for all signatures used in our work.

Table 5 Comparison with features extracted from AlexNet

| Dataset (users) | Bag of features | CNN based features (AlexNet) |
|-----------------|-----------------|------------------------------|
| CEDAR (1–7) | 94 | 92.9 |
| CEDAR (8–14) | 90 | 94.9 |
| Persian (1–7) | 68 | 94.7 |
| Persian (8–14) | 66 | 96.2 |
| Hindi (1–7) | 95 | 99.1 |
| Hindi 8 (8–14) | 98 | 99.1 |
| Bengali (1–7) | 94 | 100 |
| Bengali (8–14) | 92 | 100 |

8 Conclusion

Two classifiers, i.e., Decision Tree and SVM are used for offline signature recognition. Utilizing features based on AlexNet, i.e., deep features and SVM as classifier shows better outcomes. Best results are achieved for Bengali handwritten signature with 100% accuracy. For Persian signature, we got accuracy of more than 80% for each user. Twelve users out of fourteen users for Hindi signatures are 100% recognized. For English signature, we acquired average accuracy of 94% for samples of seven users (first set) and 90% for second set. Result shows that we obtained average accuracy of more than 92% using features extracted from pretrained DCNN, i.e., AlexNet and SVM as classifier for all signatures as shown in Table 5. Experiments are also performed for offline signature recognition using bag of Features and SVM. We obtained average accuracy of more than 90% for all signatures except Persian signature using bag of Features as shown in Table 5. In our work, all the investigations were performed on scanned signatures of fourteen users each from three dataset namely CEDAR, UTSig, and BHsig260. In the future, we can apply this strategy with entire dataset and other diverse language signatures.

References

1. <http://www.aba.com/Products/Surveys/Pages/2013DepositAccount.aspx>
2. <https://nacm.org/pdfs/surveyResults/afp-payments-fraud-results.pdf>
3. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* (2012)
4. C. Szegedy et al., Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015
5. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
6. K. He et al., Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016
7. H. Gao et al., Densely connected convolutional networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017

8. M.D. Zeiler, R.Fergus, *Visualizing and Understanding Convolutional Networks* (European Conference on Computer Vision, Springer, Cham, 2014)
9. S. Minaee, A. Abdolrashidiy, Y. Wang, An experimental study of deep convolutional features for iris recognition, in *2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)* (IEEE, 2016)
10. W. Cui et al., Offline Chinese Signature Verification Based on AlexNet, in *International Conference on Advanced Hybrid Information Processing* (Springer, Cham, 2017)
11. N.N. Kamal, L.E. George, Offline signature recognition using centroids of local binary vectors, in *International Conference on New Trends in Information and Communications Technology Applications* (Springer, Cham, 2018)
12. S.-J. Zhang, Y. Aysa, K. Ubul, BoVW Based Feature Selection for Uyghur Offline Signature Verification, in *Chinese Conference on Biometric Recognition* (Springer, Cham, 2018)
13. Z. Jan et al., An automated System for offline signature verification and identification using delaunay triangulation, in *New Contributions in Information Systems and Technologies* (Springer, Cham, 2015), pp. 653–663
14. K.S. Manjunatha, D.S. Guru, H. Annapurna, Interval-valued writer-dependent global features for off-line signature verification, in *International Conference on Mining Intelligence and Knowledge Exploration* (Springer, Cham, 2017)
15. M. Ammar, Performance of parametric and reference pattern based features in static signature verification: a comparative study, in *[1990] Proceedings of 10th International Conference on Pattern Recognition*, vol. 1 (IEEE, 1990)
16. D. Suryani, E. Irwansyah, R. Chindra, Offline signature recognition and verification system using efficient fuzzy kohonen clustering network (EFKCN) algorithm. *Procedia Comput. Sci.* **116**, 621–628 (2017)
17. N.G. See, O.H. Seng, A neural network approach for off-line signature verification, in *Proceedings of TENCON'93. IEEE Region 10 International Conference on Computers, Communications and Automation*, vol. 2 (IEEE, 1993)
18. M.I. Malik, M. Liwicki, A. Dengel, Local features for forensic signature verification, in *International Conference on Image Analysis and Processing* (Springer, Berlin, Heidelberg, 2013)
19. H. Rantzs, H. Yang, C. Meinel, Signature embedding: writer independent offline signature verification with deep metric learning, in *International Symposium on Visual Computing* (Springer, Cham, 2016)
20. K. Huang, H. Yan, Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recognit.* **30**(1), 9–17 (1997)
21. D.R.S. Kumar et al., Combined off-line signature verification using neural networks, in *International Conference on Advances in Information and Communication Technologies* (Springer, Berlin, Heidelberg, 2010)
22. K. Gyimah et al., An improved geo-textural based feature extraction vector for offline signature verification. *J. Adv. Math. Comput. Sci.* 1–14 (2019)
23. M.I. Malik et al. ICDAR2015 competition on signature verification and writer identification for on-and off-line skilled forgeries (SigWIcomp2015), in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* (IEEE, 2015)
24. A.K. Shukla et al., Offline signature verification system using grid and tree based feature extraction, in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)* (IEEE, 2014)
25. C. Fick, J. Coetzter, J. Swanepoel, Efficient curve-sensitive features for offline signature verification, in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)* (IEEE, 2016)
26. K. Kumari, S. Rana, Offline signature verification using intelligent algorithm. *Int. J. Eng. Technol.* 69–72 (2018)
27. S. Rana, A. Sharma, K. Kumari, Performance analysis of off-line signature verification, in *International Conference on Innovative Computing and Communications* (Springer, Berlin, 2020), pp. 161–171

28. K. Kumari, S. Rana, Off-line persian signature verification: an empirical evalution. *Int. J. Innovative Technol. Exploring Eng. (IJITEE)* (2019)
29. M. Soroush, S.H. Sardouie, N. Samiee. A Novel algorithm based on decision trees in multiclass classification, in *2018 25th National and 3rd International Iranian Conference on Biomedical Engineering (ICBME)* (IEEE, 2018)
30. M.S. Fahmy, A.F. Atyia, R.S. Elfouly, Biometric fusion using enhanced svm classification, in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (IEEE, 2008)
31. <http://www.mathworks.in/>
32. <http://www.cedar.buffalo.edu/NIJ>
33. A. Soleimani, K. Fouladi, B.N. Araabi, UTSig: a Persian offline signature dataset. *IET Biometrics* 6(1) (2016)
34. <https://goo.gl/9QfByd>

An Analysis and Comparative Study of Data Deduplication Scheme in Cloud Storage



Pronika and S. S. Tyagi

Abstract Data deduplication is another method for compressing the information and it is useful in productive utilization of storing the space and ends up being a superior strategy to deal with duplicate information. Deduplication permits unique or single data copies are stored in the database initially and sequential duplicates are delivered with interfacing indicator to the authorized stored replica of information. This paper analyses the deduplication scheme and comparison between them and the idea about the methodology also. The study showed is structured in smooth way to carry out the crisp of improvement in the deduplication schemes.

Keywords Cloud computing · Data deduplication · Cloud storage · Convergent encryption

1 Introduction

Data deduplication is a fundamental system to develop storing proficiency in cloud computing. By directing excess documents toward a single replica, cloud specialist co-ops decrease their extra space just as information transfer costs [1]. The deduplication that is utilized to decrease data by eliminating the copy duplicates of similar information and its extensive utilization in cloud storage to spare bunches of transfer speed and decrease the space. Distinct client utilizes their own private key for encoding or decoding to accomplish the confidentiality of delicate data. In Fig. 1, if a duplicate of the record occurs and client rights coordinated with the rights exist in the cloud and just the client can get the indicator for a proportional document [2].

Pronika (✉) · S. S. Tyagi

Manav Rachna International Institute of Research & Studies, Faridabad, Haryana, India
e-mail: pronika.fet@mriu.edu.in

S. S. Tyagi
e-mail: shyam.fet@mriu.edu.in

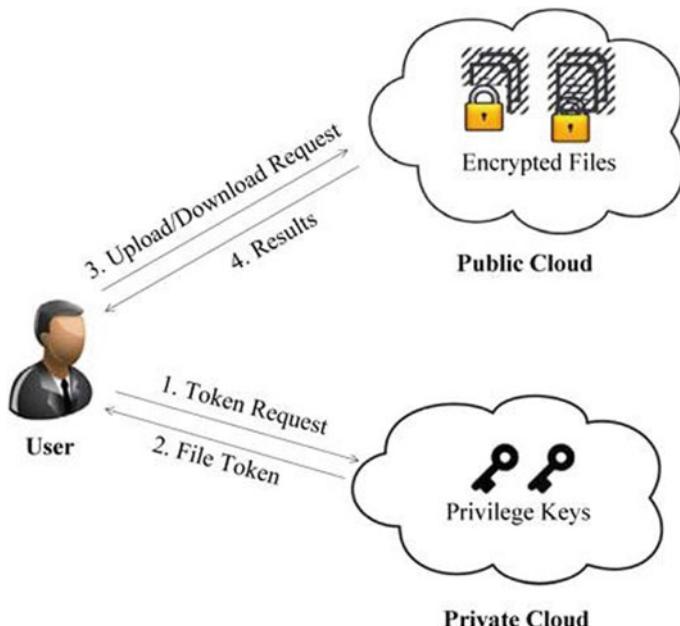


Fig. 1 Authorized deduplication [2]

In deduplication, the role of convergent encryption technique is very important. With the help of this technique, the data which is to be uploaded to the cloud are encrypted [3, 4]. The main drawback of this technique is the huge quantity of storage necessary in key management [5].

The e-learning based on cloud computing is shown an essential and commanding part in the field of learning and education. In this first, the message is encrypted by DES encryption and to encode or decode it with the help of erasure code. The overview of cloud architecture for e-learning shows in Fig. 2 [6].

In the data deduplication frameworks, the impacts of the damage of a block can be estimated by the quantity of records that are out of reach because of this loss. In like manner, we estimate the significance of a block by counting the quantity of records based upon the piece [7]. Using Blowfish algorithm calculation it encourages the client to produce an exceptional id for encoding the message and a similar key is utilized to recover the information [8]. Deduplication can either be block-level or file level. Block-level methodology breaks the record into fixed size or variable size blocks [9].

Deduplication method basically removes duplicate files and keeps on a unique copy of file as shown in Figs. 3 [9]. To process each block of data with a hash algorithm, this is used to generate a unique hash value. If the hash value of data block is already in an index, the data block need not be stored and it is considered as duplicate [10].

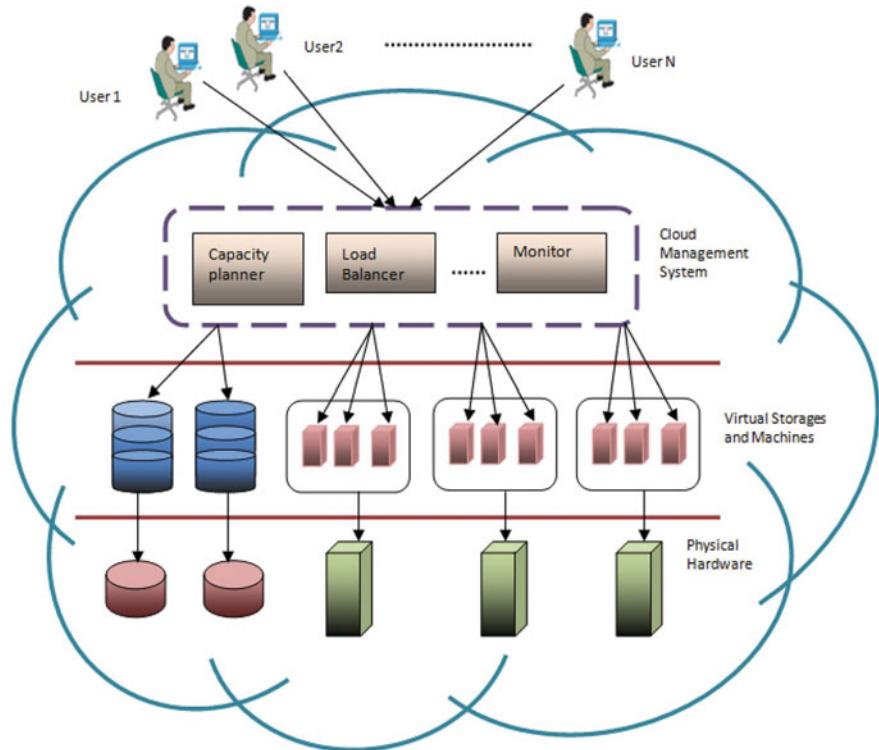


Fig. 2 Overview of cloud architecture for e-learning system

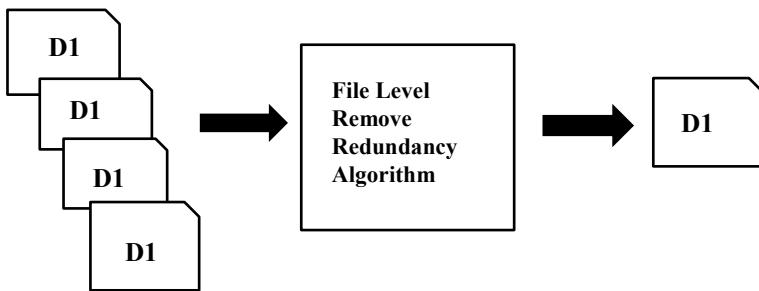


Fig. 3 Deduplication method remove duplicate files

2 Literature Survey

Fan et al. ([1], p. 127) discussed the techniques related to improve storage efficiency in cloud computing with the help of data deduplication. They proposed a trusted

execution environment scheme in which authorized cloud users have a correct privilege set. In this scheme, convergent encryption with correct privilege set and it believes on trusted execution environment to offer secure key management, enhance the capability of that cryptosystem to battle chosen ciphertext attacks and plain text attacks. This scheme provides the confidentiality of sensitive data as well as performs the security analysis related to performance evaluation.

Uma and Jayasimman [3] purposed a scheme for key management and key generation as a service device for maintaining and generating the key for the deduplication process. With the help of convergent encryption, the key or hash value provided, and using this hash value the original data will be encrypted. Using this technique the burden on the user for generating and maintaining the key will be reduced.

Waghmare and Kapse ([4], p. 815) suggests some terms related to an authorization which give help in purposed schemes like symmetric encryption, deduplication, convergent encryption, and token generation. In this scheme, first doing the registration process then file uploading, encrypt the data and check that data already stored or not. With these authors improve the storage efficiency and they reduce the duplicate copies of the data in cloud storage.

Yan et al. ([11], p. 547) proposed a technique to de-duplicate the encoding information saved in cloud which depends upon PRE. Authors measure the efficiency and performance of the scheme, according to them the data sharing and data update with the assistance of de-duplication also possible when the data owner is offline. In this scheme, basically three entities are data holders, authorized party, and cloud service, providers. Efficiency test of every operation of this technique with varying sizes of AES symmetric keys like 256 bits, 196 bits, and 128 bits done, and the purposed scheme is efficient.

Singh et al. ([5], p. 156) proposed a technique that talks about the issue of efficient key management, data confidentiality, integrity, and fault tolerance. In this scheme, the key overhead is lowest with the help of the Chinese remainder theorem, and data is spread into arbitrary looking segments at multiple servers using the concept of proof of ownership. Authors take different data sets like audio clips, android application data, and other application files found on mobile devices. The purposed scheme provides some features like client-side deduplication, tag consistency preservation, deduplication of encryption data, an update of outsourced data and it provides one more feature of fault tolerance. So, the authors purposed a scheme which provides efficient key management, minimized key overhead, and deduplication rate is 24% more than the previous techniques.

Jose and Christopher ([6], p. 12,857) purposed a technique in e-learning system in which first encodes the information using different encryption schemes. In the purposed scheme authors discuss the Reed Solomon Code, in which user encrypts the file but when the size of the file is big then divide this file into blocks. After that, the blocks of the file are encoded individually and save the data. In this paper, delay time measurement can be found with the help of request time and response time of the file when it is uploaded on the cloud storage system. The purposed scheme requires less block size for maximum performance.

Rawal et al. ([12], p. 1161) discussed a secure disintegration protocol (SDP) for the protection of privacy of the data in the cloud and on-site also. The purposed protocol is used in combination with encoding techniques and data compression techniques. It used firewall routers for the security of the data like SSL, TLS, etc. SDP basically used its own redundancy system and fault tolerance system that fails when the whole server situated on multi-cloud fails. The purposed work efficiently used the CPU but it does not utilize the cryptography method effectively.

Shafi'i et al. [8] analyzed cloud user wants to secure his data but the provider does not promise the security like as integrity, confidentiality, and security of data. The authors purposed Blowfish algorithm in which the user can encode and decode the data. The purposed technique basically a license-free, unpatented, and highest speed algorithm and it works on below 4 GB file size. With this scheme, users get the solo identification for encoding the information or data and a similar key is used for retrieving the message from the cloud. With the help of unique id user get the authentication and he can decrypt the data from the cloud. The main drawback of blowfish is when the file size more than 4 GB then it can't encrypt the data because of small bit block size.

Pronika and Tyagi ([9], p. 364) deliberate deduplication techniques like block level and file-level deduplication, with the deduplication the redundancy of the data can be removed and storage space in cloud can be increased. But there are some drawbacks of deduplication like harm of private data, issues of backup machine, the effect on capacity execution, and downfall of information integrity.

3 Observations and Discussions

In this paper, different schemes related to data deduplication are studied. Some schemes depend upon block level or file level and they are used different types of encryption techniques for removing the duplicity of the data. In traditional deduplication, for storing the hash value of file or block single global index table is used, so performance and extra overhead are decreased. Basically, de-duplication removes the duplicity of the information but it also faces some issues related to the security of the data. We analyzed the comparison analysis of deduplication techniques with methodology in tabular form (Table 1).

Table 1 Comparison analysis of deduplication techniques with methodology

| S. No. | Author name | Year | Techniques | Methodology used |
|--------|----------------------|------|--|---|
| 1 | Waghmare and Kapse | 2016 | Convergent encryption and token generation | Deduplication using Hashing function, symmetric encryption, convergent encryption |
| 2 | Li et al. | 2015 | Technique related to Map Reduce | Integrity auditing checked by SecCloud and SecCloud + performs auditing and integrity on encrypted files |
| 3 | Zheng et al. | 2016 | Block-level deduplication | Encoded information can be securely retrieved by symmetric keys. Use PRE method to accomplish encoded information with de-duplication |
| 4 | Yan et al. | 2016 | Encryption related to attribute | Encryption related to attributes to remove duplicate data which is saved in the cloud and encouraging secure data control at same time |
| 5 | Jian et al. | 2015 | File and block-level deduplication | Based on password authentication key exchange protocol |
| 6 | Gode and Rupali Devi | 2017 | Convergent encryption | DARE scheme support authorization and it used duplicate adjacency information for same copy of information means likeness detection |
| 7 | Uma and Jayasimman | 2018 | Convergent encryption | Convergent encryption is exposed to different attacks like brute force and dictionary attacks. KGMAaaS deliver keys to new user with similar information by mentioning the authorized users |
| 8 | Puzio et al. | 2013 | Convergent encryption | Author purposed a technique ClouDedup which depends upon convergent encryption and it also includes block-level deduplication |

(continued)

Table 1 (continued)

| S. No. | Author name | Year | Techniques | Methodology used |
|--------|-------------------|------|-----------------------------|--|
| 9 | Diao et al. | 2017 | Security policy | Achieve data security of cloud storage and analyzing the security risks of user data in cloud storage |
| 10 | Zhou et al. | 2013 | Hash indexing | Performance characterization with analysis and impact of energy by deduplication under big data environments |
| 11 | Xu et al. | 2014 | Clustering and sampling | Numerical technique is used for computing size of sample space |
| 12 | Kaur et al. | 2017 | Data compression techniques | Deduplication removes duplicity of data, improve consumption of storage and reduce the cost of storage and techniques based on text, video and multimedia data |
| 13 | Yinjin et al. | 2016 | Map reduce technique | Map reduce technique is used for parallel deduplication |
| 14 | Choon et al. | 2018 | Cryptography based | Cloud integrity schemes follows POR and PDP have mostly same functionality. PDP does not offer retrieval to faulty or degraded information but POR recover that data |
| 15 | Alphonse et al. | 2017 | Convergent encryption | The authorized deduplication system provides to protect the data from attackers by generating key from Data Colouring and Water Marking and secure deduplication |
| 16 | El-Shimi et al. | 2012 | Data compression techniques | Offline Dedup new primary data deduplication system implemented in windows server operating system |
| 17 | Srinivasan et al. | 2012 | Capacity oriented | iDedup technique used for primary workload |

4 Conclusion

Data deduplication is another method for compressing the information and it is useful in productive utilization of storing the space and ends up being a superior strategy to deal with duplicate information. Deduplication permits unique or single data copies are stored in the database initially and sequential duplicates are delivered with interfacing indicator to the authorized stored replica of information. There are some techniques related to deduplication which are used by users to store their data in cloud storage. In this paper, we compare these schemes with their methodology in tabular form. Different types of encryption techniques also study that was related to deduplication. Watermarking and Data coloring generate the key and protect the data from the attackers in the authorized deduplication system. Unauthorized users cannot enter the system and they don't access the data. When we study different techniques related to deduplication, we found block-level deduplication better performs as compare to file-level deduplication and security is delivered with the help of convergent encryption to encrypt the information or data. Researchers who want to study in the field of data deduplication in cloud storage, this paper provides them some initial help. In the future, more work related to the field of data deduplication in cloud storage will be explored.

References

1. Y. Fan, X. Lin, W. Liang, G. Tan, P. Nanda, A secure privacy preserving deduplication scheme for cloud computing. Future Gener. Comput. Syst. **101**, 127–135 (2019)
2. R.V. Gode, R. Dalvim, An effective storage management in a twin cloud architecture using an authorized deduplication technique, in *The Proceedings of International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (IEEE, 2017), pp. 796–801
3. G. Uma, L. Jayasimman, Enhanced convergent encryption key generation for secured data deduplication in cloud storage. J. Phys. Conf. Ser. **1142**(1), 012006 (2018)
4. V. Waghmare, S. Kapse, Authorized deduplication: an approach for secure cloud environment. Procedia Comput. Sci. **78**, 815–823 (2016)
5. P. Singh, N. Agarwal, B. Raman, Secure data deduplication using secret sharing schemes over cloud. Future Gener. Comput. Syst. **88**, 156–167 (2018)
6. G.S.S. Jose, C.S. Christopher, Secure cloud data storage approach in e-learning systems. Cluster Comput. **22**(5), 12857–12862 (2019)
7. S. Wu, K.C. Li, B. Mao, M. Liao, DAC: improving storage availability with deduplication-assisted cloud-of-clouds. Future Gene. Comput. Syst. **74**, 190–198 (2017)
8. N.A. Shafi'i Muhammad Abdulhamid, M.A. Sadiq, M. H. Nadim Rana, Development of Blowfish encryption scheme for secure data storage in public and commercial cloud computing environment (2018)
9. Pronika, S. Tyagi, Deduplication in cloud storage. Int. J. Innovative Technol. Exploring Eng. **9**(2S), 364–368 (2019)
10. S. Hema, A. Kangaiammal, An analysis and comparison of data deduplication approaches to efficient storage. Int. J. Adv. Res. Comput. Sci. **9**(2) (2018)

11. Z. Yan, W. Ding, H. Zhu, A scheme to manage encrypted data storage with deduplication in cloud, in *The Proceedings of International Conference on Algorithms and Architectures for Parallel Processing* (Springer, Cham, 2015), pp. 547–561
12. B.S. Rawal, V. Vijayakumar, G. Manogaran, R. Varatharajan., N. Chilamkurti, Secure disintegration protocol for privacy preserving cloud storage. *Wirel. Pers. Commun.* **103**(2), 1161–1177 (2018)

Prediction of the Most Productive Crop in a Geographical Area Using Machine Learning



Atharva Karwande, Medha Wyawahare, Tejas Kolhe, Soham Kamble, Rushikesh Magar, and Laksh Maheshwari

Abstract Agriculture has a direct correlation with the development of the country. Although it is the backbone of the country, people are baffled in deciding which crop to plant and about the right time to plant it for maximizing the profits. One of the main risks that are considered is the changing climate and other environmental factors. So, we can give estimation using proper arrangement and calculations from previously stored data. This problem can be tackled using Machine Learning algorithms and we can find an efficient solution. Crop Yield Prediction involves predicting yield of the crop from available historical available data like district, produce, area, season of cultivation, year of cultivation. This paper focus on predicting the yield of the crop and also the best suitable crop based on authenticate data from Maharashtra State in India by using Random Forest algorithm.

Keywords Agriculture · Machine learning · Maharashtra · Random forest algorithm · Suitable crop · Yield prediction

A. Karwande (✉) · M. Wyawahare · T. Kolhe · S. Kamble · R. Magar · L. Maheshwari
Vishwakarma Institute of Technology, Pune, Maharashtra, India
e-mail: atharva.karwande18@vit.edu

M. Wyawahare
e-mail: medha.wyawahare@vit.edu

T. Kolhe
e-mail: tejas.kolhe18@vit.edu

S. Kamble
e-mail: soham.kamble18@vit.edu

R. Magar
e-mail: rushikesh.magar18@vit.edu

L. Maheshwari
e-mail: laksh.maheshwari18@vit.edu

1 Introduction

Agriculture acts as a pivot in the Indian economy. It contributes about 30% of the country's GDP. Major Part of the agriculture in India is located near the Indo-Gangetic Plains. As per the statistics of 2016 around 272.82 million farmers are working tirelessly in the state of Maharashtra. With this huge number of farmers and increasing suicide rates in all the states, we want to help farmers to understand the importance of crop yield prediction and the prediction of the best suitable crop in a given district. The cultivation and yield of a specific crop is largely influenced by the fertility, rainfall, moisture contents of the soil, average and the diurnal range of the temperatures, etc. But these conditions are almost uniform over an area. This area can be considered to a district. But these parameters that influence the crop yield and selection of the crop fluctuate tremendously from district to district. The crop to plant also depends on the season in which the cultivation is done e.g., Kharif, Rabi, etc. Most of the existing systems are hardware based and uses sensors and other input devices which make them expensive and difficult to maintain. The farmers also lack the technical know-how for using the device. Also, they lack to give accurate results. Our system enables the farmer to input the district and the area under cultivation and the system uses the random forest algorithm to give the prediction of the crop which has maximum yield and the predicted yield for the given input of area.

2 Related Work

In order to do the analysis, different algorithms have been studied for predicting the produce and the best suitable crop. For this, we must know the basic knowledge of preprocessing techniques, different algorithm techniques, implementation and the analysis of the results.

Priya et al. used the random forest classifier algorithm. The random forest classifier uses decision trees. The author states some advantages in the algorithms as over fitting is not an issue in it unlike the decision tree. Pruning is not necessary in this. Three parameters were used by author in the algorithm. Parameter ntree—no. of trees, mtry—how many variables are needed to split a node, nodesize—it indicates the no of observations we want in the terminal nodes. The parameters of the dataset used in the dataset are: rainfall in mm, kharif and Rabi season of each year in each district, maximum temperature in the district, crop production in tons and perception. The author has used RStudios in the process. The split ratio of the train and test dataset was 67% and 33% respectively. The results were really good but no accuracy data was provided in the paper by the authors. We can site that random forest classifier can work as an effective algorithm in prediction of crop [1]. Veenadhari et al. used the random forest classifier algorithm to predict the yield crop. The variables are all climatic factors which include cloud cover, temperature, and precipitation. The crops that are predicted by the authors were Soybean, Paddy, Maize, and Wheat.

The output accuracy that the authors give is 87%, 85%, 76%, 80% respectively. This accuracy is good [2]. Sml Venkata Narasimhamurthy et al. used random forest algorithm to predict the yield of a specific crop that is Rice. The study is restricted to the state of Andhra Pradesh of India where the staple crop is Rice. The author considers various numerical parameters while implementing the algorithm which are in the dataset. The parameters include temperature, rainfall, production, precipitation, type of crop, etc. The conclusion of the paper states that the overall accuracy of the model that is built is 85.89%. The author states that the accuracy was 8% better than MLR. Also, the implementation is done the RStudios IDE [3]. Jeong et al. wrote a paper on the efficiency outputs of different regression algorithm and the random forest algorithm which gives yield output. The author states that RF algorithm has many advantages but it lacks the usage in the field of agriculture. They deduced that random forest provides better-quality performance in predicting yield of crops in every region tested. The strong potential of this algorithm is highlighted in this paper [4]. Kulkarni et al. helped us understand better about the algorithm Random Forest. The author compared the no. of trees and the output and the accuracy of each output. This paper dives into the mathematics involved in the computation of the output and the formation of the trees [5]. Everingham et al. used random forest algorithm to accurately predict the yield of specific crop sugarcane. The author takes into account the data from Tully region of Australia from 1992 to 2013. The parameters used in the predictions include radiation, temperature, and yield per unit area. This paper was just to demonstrate the use of big data to predict the sugarcane yield in Tully, Australia [6]. Renuka et al. compared different algorithms and found that decision trees which are the backbone of RF has way better accuracy as compared to other algorithms. This proves that random forest algorithm which uses decision tree is the best approach to predict the yield. But the problem in hand after implementation of the algorithm is over fitting. This was done in the Spyder IDE [7]. Pandhe et al. used 5 climatic parameters for prediction, i.e. Precipitation, Temperature, Cloud Cover, Vapour pressure, Wet Day Frequency. Number of trees that were used was 200 to implement the RF regression. The output gave 87% accuracy [8].

3 Proposed Methodology

Due to recent advent in machine learning algorithms, both supervised and unsupervised, there are many algorithms that can handle our dataset. The best found algorithm was random forest as our data mostly contains categorical variable. The SVM algorithm did not give any output as the number of levels in the training and testing dataset were not equal. When SVM was implemented on complete India dataset the accuracy was 27% which is extremely poor RStudios IDE was used for computational purposes as it is the top data processing and machine learning implementation tool. Dataset used is an authenticate dataset published by Government of Maharashtra, India. The dataset is in the following format (Table 1).

Table 1 Dataset published by Government of Maharashtra, India

| State name | District name | Crop year | Season | Crop | Area | Production |
|-------------|---------------|-----------|--------|-----------|---------|------------|
| Maharashtra | Ahmednagar | 1997 | Autumn | Maize | 1 | 1113 |
| Maharashtra | Ahmednagar | 1997 | Kharif | Arhar/Tur | 17,600 | 6300 |
| Maharashtra | Ahmednagar | 1997 | Kharif | Bajra | 274,100 | 152,800 |
| Maharashtra | Ahmednagar | 1997 | Kharif | Gram | 40,800 | 18,600 |
| Maharashtra | Ahmednagar | 1997 | Kharif | Jowar | 900 | 1100 |

3.1 Data Cleaning and Preprocessing

The data acquired contains approximately 246,100 data points. The dataset includes the data from all the states of India. The parameters i.e. the column head include State, District Name, Year of Cultivation, Season, Crop, Area, and Production. The dataset years are from 1997 to 2014. The metadata of the dataset is not available therefore there are no units that are to be considered in this project. The data had too many points. So due to computational limitations we sorted out the data of Maharashtra State only. This reduced the data points to approximately 13 thousand. The preprocessing of the data is done manually. We removed the rows which contained NaN and also merged some crops which were in more than one season. The data is arranged alphabetically according to the district. All the character variables were converted into factor variables. The library used for this job was “dplyr”. Another column was created named “Production_per_unit_Area” which was obtained by dividing the Production by the Area column.

3.2 Implementation of the Algorithm

Random forest is a supervised learning algorithm which forms multiple decision trees by random selection of part of dataset. Each tree is formed after sampling different part of the dataset. The library “randomForest” in R is used to implement the Algorithm. Input from the user was taken. The input includes the district and the area of cultivation. This input is passed to the function as argument. This is done to decrease the runtime of the program. If we don't do this the time taken is much more. Within the function, a new dataset is created of the given district. Splitting of the new data into training and validating dataset was done. The ratio of splitting is 0.8 and 0.2 respectively. We use the function set.seed(123) to fix a time for the random splitting, which produces the same training and validating dataset when the code is re-run. The next step is to use the algorithm on the training dataset. We use the function randomForest() and pass the parameter “Production per unit area” to be predicted and the variables are all the rest variables. The model is obtained. Then we use the function predict() to predict the Production per unit are for the testing dataset. After the predictions are stored in a variable, we add that column to the Testing dataset.

3.3 Error and Accuracy

The error was calculated on traditional bases. Then the error is found out by subtracting the predicted value and the true data value. Then the Root Mean Square Error for that is found.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^N (\text{Predicted}_i - \text{Actual}_i)^2}{N}}$$

4 Result

With the help of the aggregate() function, we find out the average value of the predicted value with respect to each crop. Then we arrange the data in descending order the predicted production per unit area. Then we multiply all the predicted value by the Area which is given as input by the user and store the corresponding value in a column in the data. Then the data of the three columns is formed which include the predicted production per unit area, crop name and the estimated production for the given area. Then we display the top six result of that data (Fig 1).

The root mean square error of each and every district is different as we only run the algorithm for the district that the user gives as input so as to decrease the runtime of the algorithm. The root mean squares of each district are as follows:

The log scale is being used to scale the graph.

Figure 2 shows the scatter plot of log of predicted vs. original values for Jalgaon (District with maximum RMSE = 3.391512). The correlation is = 0.3822853

Figure 3 shows the scatter plot of log of predicted vs. original values for RAIGAD (District with minimum RMSE = 0.02757). The correlation = 0.9526128.

```
> source('G:/SY EDI SEM 1/EDI_FINAL/file.R')
Enter the District:JALGAON
Enter the Area of cultivation:50
[1] "The Root Mean Square Error of the prediction is: "
[1] 3.391515
[1] "The Production per unit area and the best suitable crops are of the district is given below: "
      Crop predicted Estimated_Production
9     Maize 152.01991    7600.9954
16   Safflower 102.06821    5103.4107
3    Castor seed 99.26944    4963.4720
19   Sugarcane 54.46142    2723.0708
4 Cotton(Lint) 26.90263    1345.1313
6     Grapes 15.35250     767.6249
> |
```

Fig. 1 Output of the program for Jalgaon district of Maharashtra with the Area input: 50 units

Fig. 2 The scatter plot of log of predicted versus original values for Jalgaon

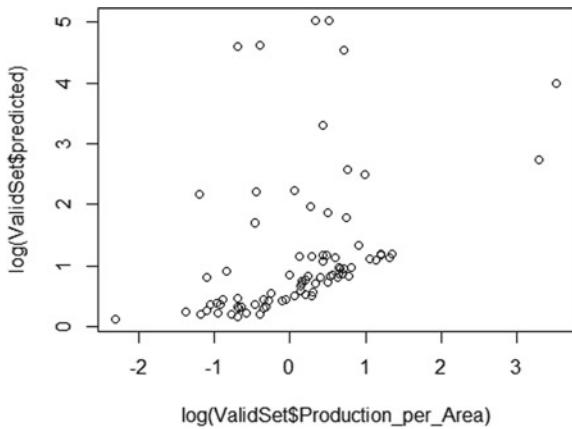
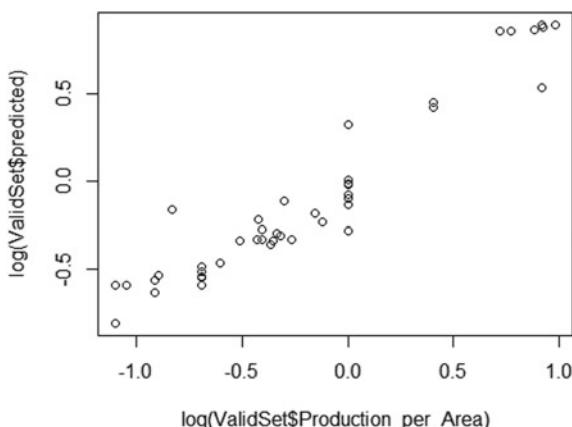


Fig. 3 The scatter plot of log of predicted versus original values for Raigad



5 Discussion

Analysis of the RMSE is as follows.

The Box plot of the results Table 2 is as follows (Table 3; Fig. 4).

The Histogram of the results Table 2 is as follows (Fig. 5).

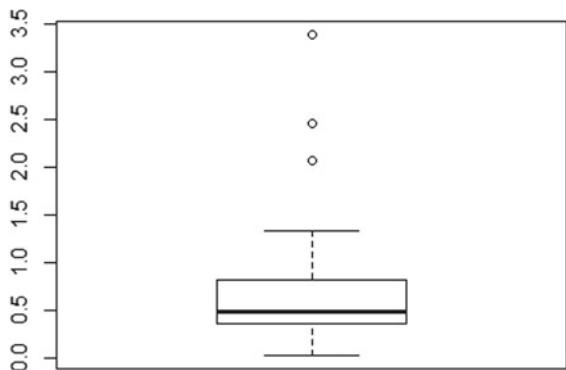
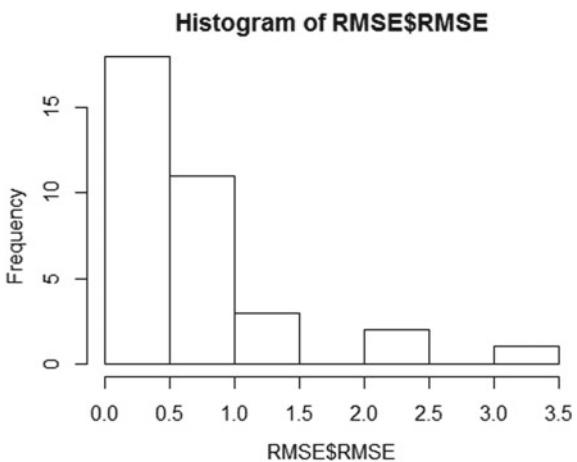
Our dataset contained mostly categorical variable. All the other algorithms, such as linear regression, support vector regression etc. failed to give any output. The random forest algorithm works best in our case. The alternative of this could be decision tree but to avoid the problem of over-fitting, the method was discarded. The random forest gives the average of many decision trees and the system is therefore is bias-variance balanced.

Table 2 Root mean square error of districts of Maharashtra, India

| District | RMSE |
|------------|----------|
| Ahmednagar | 0.551799 |
| Akola | 0.767941 |
| Amravati | 0.872806 |
| Aurangabad | 0.651883 |
| Beed | 0.297759 |
| Bhandara | 0.454282 |
| Buldhana | 0.557073 |
| Chandrapur | 1.292958 |
| Dhule | 1.14727 |
| Gadchiroli | 0.094282 |
| Gondia | 0.44003 |
| Hingoli | 0.390279 |
| Jalgaon | 3.391515 |
| Jalna | 0.244364 |
| Kolhapur | 0.392774 |
| Latur | 0.50645 |
| Nagpur | 0.711356 |
| Nanded | 0.367457 |
| Nandurbar | 0.104483 |
| Nashik | 1.33505 |
| Osmanabad | 2.063202 |
| Parbhani | 0.489779 |
| Pune | 0.898194 |
| Raigad | 0.027569 |
| Ratnagiri | 0.048851 |
| Sangli | 2.45419 |
| Satara | 0.477299 |
| Sindhudurg | 0.352843 |
| Solapur | 0.350085 |
| Thane | 0.059841 |
| Wardha | 0.933874 |
| Washim | 0.450866 |
| Yavatmal | 0.377975 |

Table 3 Mean and five number summary of Table 2

| Minimum | 1st quartile | Median | Mean | 3rd quartile | Maximum |
|---------|--------------|---------|---------|--------------|---------|
| 0.02757 | 0.36015 | 0.48978 | 0.71388 | 0.82037 | 3.39152 |

Fig. 4 Box plot of Table 2**Fig. 5** Histogram of Table 2

6 Conclusion

Crop prediction using Machine Learning based on previous available authentic data is the requirement of time. Existing research employs complex neural networks which can be computationally very expensive and difficult to develop. Hence it is very crucial to develop a system which is relatively less expensive computationally.

Our system implements Random Forest Algorithm to predict the crop which has maximum yield per unit area in the district. This system is optimized and is very fast keeping in mind the enormous data being processed. Our system has succeeded to achieve the root mean square error of 0.48978 (median value). This algorithm works best in our case where the variables are mostly categorical whereas other algorithms fail to give any plausible output.

References

1. P. Priya1, U. Muthaiah, M. Balamurugan, Predicting yield of the crop using machine learning algorithm. Int. J. Eng. Sci. Res. Technol. <https://doi.org/10.5281/zenodo.1212821>
2. S. Veenadhari, B. Misra, C.D. Singh, Machine learning approach for forecasting crop yield based on climatic parameters, in *2014 International Conference On Computer Communication And Informatics (ICCCI-2014)*, 03–05 January 2014, Coimbatore, India
3. SML V. Narasimhamurthy, AVS P. Kumar, Rice crop yield forecasting using random forest algorithm. Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET) 5(X) 2017. ISSN: 2321-9653; (Ic Value: 45.98; Sj Impact Factor:6.887)
4. J.H. Jeong, J.P. Resop, N.D. Mueller, D.H. Fleisher, K. Yun, E.E. Butler, D.J. Timlin, K.-M. Shim, J.S. Gerber, V.R. Reddy, S.-H. Kim, Random forests for global and regional crop yield predictions. Plos One (2016). <https://doi.org/10.1371/journal.pone.0156571>
5. V.Y. Kulkarni, P.K Sinha, Effective learning and classification using random forest algorithm. Int. J. Eng. Innovative Technol. (IJEIT) 3(11) (2014)
6. Y. Everingham, J. Sexton, D. Skocaj, G. Inman-Bamber.: Accurate prediction of sugarcane yield using a random forest algorithm. Agron. Sustain. Dev. 36, 27 (2016). <https://doi.org/10.1007/s13593-016-0364-z>. Accepted 22 Mar 2016. Published Online: 19 April 2016 # Inra And Springer-Verlag France 2016
7. Renuka, S. Terdal, Evaluation of machine learning algorithms for crop yield prediction. Int. J. Eng. Adv. Technol. (IJEAT) 8(6). ISSN: 2249 – 8958 (2019)
8. A. Pandhe, P. Nikam, V. Pagare, P. Palle, D. Dalgade, Crop yield prediction based on climatic parameters. Int. Res. J. Eng. Technol. (IRJET) 6(3) (2019). E-ISSN: 2395-0056

The Smart Set: A Study on the Factors that Affect the Adoption of Smart Home Technology



S. Shanthana Lakshmi and Deepak Gupta

Abstract The integration of Artificial Intelligence with Smart Technologies, which entails automated solutions to common and important global challenges, holds a promising future in India. Smart Home technology permits for well-organized monitoring and control of Smart Home devices. AI integration with home automation has accelerated the pace for Smart Home adoption among house and apartment owners. However, the scholarly research aimed at understanding the user's acceptance of Smart Home technology is surprisingly limited. With the booming market for Smart Home products in India in mind, our study investigates the major factors that affect the adoption of Smart Home technology. The study also aims to understand the influence of environmental centric attitude of consumers on their intent to embrace Smart Home technology. By extending the UTAUT2 (The unified theory of acceptance and use of technology) model, this research integrates other factors such as trustworthiness, psychological risk, tech-savvy and energy conservation attitude of users to understand the purchase intention of residents. Ordered logistic regression is applied to data collected from a pan-India sample of 148 respondents. The result shows that performance expectancy and trialability along with trustworthiness and tech-savvy attitude has a significant influence in affecting the user's adoption of Smart Home technology. Additionally psychological risk and environmental concern attitude of user's negatively influences the purchase intention of residents.

Keywords Smart homes · Smart home adoption · IOT · UTAUT2 · Energy conservation

S. Shanthana Lakshmi (✉) · Deepak Gupta
Amrita School of Business, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: sandyselvan666@gmail.com

Deepak Gupta
e-mail: dgshobs@gmail.com

1 Introduction

Rapid advances have been made in the production and adoption of Internet of Things (IOT) devices in the past few years to improve the quality of our lives and our standard of living [1]. Potential IOT devices enhance the functioning of day-to-day life in many aspects like healthcare, supply chain management, etc. One of them is Smart Homes, which enriches one's quality of life by providing remote control and monitoring of Smart Homes/environment [2]. Though the digital trends show that there is a high potential for Smart Home technology in India, the adoption rate of Smart Homes in India is considerably low. Understanding these factors can help the marketers promote and project Smart Home technology in a better way. In our study, we aim to analyse specific factors from UTAUT2 model; alongside other factors like trustworthiness, psychological risk, tech-savvy attitude and energy conservation through an empirical survey consist a base of 148 respondents.

The structure of this paper is as follows: Sect. 2 consists of literature review followed by the conceptual model and hypotheses development in Sects. 3 and 4, respectively. Section 5 consists of methodology used followed by data analysis and results in Sect. 6. Section 7 consists of limitations of the research followed by conclusion in Sect. 8.

2 Literature Review

This section reviews the current knowledge base relevant to this discussion.

2.1 *Previous Research on Barriers to Adoption of Smart Home Technology*

The exploration started by analysing the association between perceived risk factor and adoption of innovation. Extant research suggests that consumers have genuine concerns about loss of control over their homes and their lives, as well as of a decline in their physical activeness [3]. In sum, the findings from the existing researches suggest that perceived risk factor has a significant influence in the adoption of Smart Home technology. Accordingly we have also considered perceived risk as a major variable of study and have divided perceived risk as—psychological risk, financial risk (price value), privacy risk and performance risk.

An important factor that helps mitigate the perceived risks of adoption is the trust that consumers play on the technologies and their vendors. Existing research suggests that trust is indeed playing an important role in determining the consumer attitudes towards and adoption of Smart technologies too [1]. Consumers trust towards an innovative technology plays a crucial role in helping them overcome the perceived

risk factors and adopt the Smart Home technology. Thus we have also included trustworthiness of the technology as an important factor in our study.

In contrast to prior innovations, IOT gadgets are autonomous devices that work on an always on mode usually using sensors. While they do not require continuous active inputs from the user there is a certain amount of skill needed to make optimum use of the technologies [4]. In other words smart technologies require learning new skills that makes it easier for a person to understand and interpret the interface. Tech-savvy consumers are considered to be the first movers towards the adoption of innovative technologies, as they have the basic knowledge and motivation towards purchasing the technology, as well as the willingness to make the necessary learning investments. Therefore we hypothesize a positive role for the tech-savvy attitude of consumers in our study.

The ongoing debate on climate changes and environmental degradation have also spurred a strong interest in the limiting the use of fossil fuels and energy conservation. The potential of Smart technologies in aiding a more efficient use of energy in user homes has also been explored and current research indicates that it could be a potential factor in the adoption of smart homes, though consumer knowledge about utilizing Smart technology for conserving energy is still limited [5]. There is a clear research gap in understanding the energy conserving attitude of consumers and its influence on the adoption of Smart Home technology. Accordingly, we have considered Energy Conservation as one of the major variables in our study.

In addition to these, we have taken Price value, Performance expectancy and Effort expectancy variables from UTAUT2 Model.

3 Conceptual Model

See Fig. 1.

4 Hypotheses

4.1 UTAUT2 Model

The UTAUT 2 model (The unified theory of acceptance and use of technology) is among the most popular models today with respect to technology adoption and acceptance behaviour of user's. The UTAUT2 model—consists of 7 constructs like performance expectancy, effort expectancy, social influence, facilitating conditions, hedonic motivation, price value and habit.

In the research context we use performance expectancy to measure the perceived level of benefits that a user derives from using a Smart Home technology. Effort expectancy is used to measure the perceptions of consumer regarding the effort

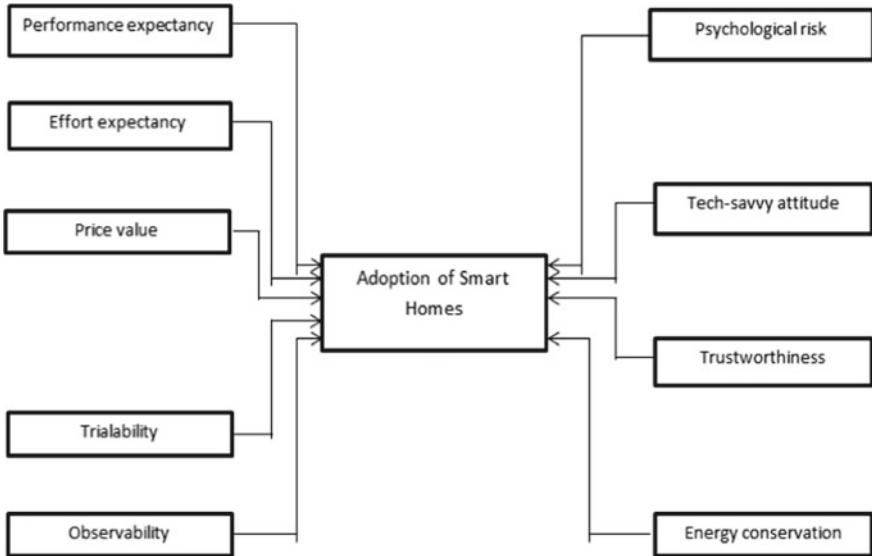


Fig. 1 Conceptual model regarding the factors affecting the adoption of Smart Home technology

required to utilize a Smart Home technology. In general terms, effort expectancy is used to measure the level of effort a consumer needs to put in and the easiness of using the Smart Homes. Price value is the perception of consumers that the price they pay for the Smart Home technology is equivalent to the benefit they derive from using it [6].

H1 The performance expectancy is positively related to the potential user's intent to adopt Smart Home technology.

H2 The effort expectancy is negatively related to the potential user's intent to adopt Smart Home technology.

H3 The price value perceived by potential users is positively related to the potential user's intent to adopt Smart Home technology.

4.2 Trustworthiness

Trust is an important factor which strongly influences the adoption of an innovative technology. When a user has a high level of trust towards a technology, he/she will have a positive attitude towards adoption of it. In our research context we have used trust to define the user's belief about the reliability aspect of Smart Home technology and posit its significant influence on their purchase intention.

H4 The trustworthiness of Smart Homes as perceived by the users is positively related to the potential user's intent to adopt Smart Home technology.

4.3 Trialability and Observability (DOI)

Trialability and observability plays a crucial role in innovative technology products, as it is important for consumer's to be experienced to understand the full potential/value of the technology.

H5 The trialability factor is positively related to the potential user's intent to adopt Smart Home technology.

H6 The observability factor is positively related to the potential user's intent to adopt Smart Home technology.

4.4 Psychological Risk

Generally Risk is an inevitable factor, which is attached with Smart Home usage. One of them is psychological risk, which defines the significant impact of Smart Home technology on the consumer's self-image or lifestyle.

H7 The psychological risk attached with Smart Homes as perceived by the users is negatively related to the potential user's intent to adopt Smart Home technology.

4.5 Tech-Savvy Attitude

Highly tech-savvy simply means the proficiency or expertise of a person in the field of technology. Tech-savvy people are considered to be the first movers in the adoption of innovative technology and this should be true for Smart Home technology too.

H8 The tech-savvy attitude of users is positively related to the potential user's intent to adopt Smart Home technology.

4.6 Energy Conservation

Energy conservation refers to the optimum use of resources with an intention of reducing energy usage/wastage. Smart Home technology allows users to use energy

resources efficiently by monitoring and controlling appliances with advanced technologies allowing for more efficient energy conservation [7]. Thus we posit that the energy conservative attitude of consumers may influence their intent to adopt Smart Home Technology.

H9 The Energy Conservation attitude of users is positively related to the potential user's intent to adopt Smart Home technology.

5 Methodology

The cross-sectional survey method was used in this study in order to gather quantifiable data and to analyse the research findings using statistical/mathematical techniques. A questionnaire instrument was developed and employed in this study. The questionnaire consists of different scales from academic literature for measuring the independent variables (as depicted in figure). A pilot testing was conducted to improve the questionnaire, remove perceived ambiguities and enhance the reliability and validity of the instrument.

The sample involved 148 respondents from across India who were demographically segmented on the basis of age, gender, residential location and annual household income. A non-probabilistic quota sampling technique and snowball sampling was used in the survey to secure an effective response rate. Finally in the end an aggregate of 140 responses were viewed as valid and was used in the analysis.

6 Data Analysis and Results

In this research we have used Ordered logistic regression model to test and run the data analysis. Our results showed that the perceived level of performance expectancy of Smart Home technology significantly influenced the intent to adopt Smart Homes—the odds increased by 17.2%. The trialability factor also significantly impacted the user's acceptance of Smart Home Technology, increasing the odds by 16.7% (Table 1).

It was also observed that as expected psychological risk has a negative influence towards the adoption of Smart Homes, with the odds of adoption decreasing by 19.21%. On the other hand, trustworthiness had a positive impact on the intent to use with the odds increasing by 7.5%. The users with high level of trust towards the Smart Home technology are more likely to purchase it in future. The regression also showed that the tech-savvy individuals are 9.2% more likely to adopt the Smart Home technology than others in the first place.

Our analysis indicated that Energy Conservation attitude of people matters—the odds for intent to purchase of Smart Home technology went up by 6%.

Table 1 Results of ordered logistic regression model

| Variables | Odds ratio | Z | P > z | 95% conf. interval |
|------------------------|------------|-------|--------|--------------------|
| Performance expectancy | 1.172*** | 2.68 | 0.007 | 1.043 1.318 |
| Effort expectancy | 1.006 | 0.10 | 0.920 | 0.8842 1.146 |
| Price value | 1.023 | 0.70 | 0.487 | 0.9583 1.093 |
| Trialability | 1.167** | 2.18 | 0.029 | 1.0155 1.342 |
| Observability | 0.9435 | -1.11 | 0.266 | 0.8515 1.045 |
| Psychological risk | 0.8079*** | -2.95 | 0.003 | 0.7011 0.9310 |
| Trustworthiness | 1.075** | 1.86 | 0.043 | 0.9960 1.160 |
| Tech-savvy attitude | 1.092*** | 3.25 | 0.001 | 1.035 1.152 |
| Energy conservation | 1.060* | 1.82 | 0.068 | 0.9955 1.130 |

Note ***P < 0.01, **P < 0.05, *P < 0.10

The other factors like effort expectancy and price value from UTAUT2, along with observability did not have any salient impact on the intent to adopt Smart Home technology by users.

7 Limitations and Future Research

The sample size of the study was limited and also the target group consisted of only Indian population. By expanding the sample size and by conducting the study globally can make a significant difference in the results. Further our study was only focused on specific set of limited factors that affects the user characteristics. However, there are other factors which can be analysed and tested (like the ecological motivation and hedonic motivation as the driving factor).

8 Conclusion

The results affirm that the factors like performance expectancy, trialability, trustworthiness, psychological risk and tech-savvy attitude has a salient influence on the purchase decisions of Smart Home technology by users.

By creating trust and by explicitly showing the performance value attached with Smart Homes, one can increase the adoption rate or purchase of Smart Home technology. Consumers with higher level of performance expectancy in relationship to the benefits they perceive are more likely to purchase Smart Homes. The researchers can use this study as a base for understanding some key factors that affects the Smart Homes adoption.

References

1. Ahmed Shuhaimi, Ibrahim Mashal, Understanding users' acceptance of smart homes. *Technol. Soc.* **58** (2019)
2. S. Nikou, Factors driving the adoption of smart home technology: an empirical assessment. *Telematics Inf.* **45** (2019)
3. H. Areum, C. Nam, S. Kim, What will be the possible barriers to consumers adoption of smart home services. *Telecommun. Policy* **44**(2) (2020)
4. P.S. de Boer, A.J.A.M. van Deursen, T.J.L. van Rompay, Accepting the Internet-of-Things in our homes: the role of user skills. *Telematics Inform.* **36**, 147–156 (2019)
5. Abhishek Bhati, M. Hansen, C.M. Chan, Energy conservation through smart homes in a smart city: a lesson for Singapore households. *Energy Policy* **104**, 230–239 (2017)
6. Y. Kim, Y. Park, J. Choi, A study on the adoption of IoT smart home service: using value-based adoption. *Total Q. Manag. Bus. Excellence* **28**(9–10), 1149–1165 (2017)
7. D. Marikyan, S. Papagiannidis, E. Alamanos, A systematic review of the smart home literature: a user perspective. *Technol. Forecast. Soc. Change* **138**, 139–154 (2019)

New Approach for Multimodal Biometric Recognition



S. Preetha and S. V. Sheela

Abstract It is known fact that passwords are no more suitable for security needs of twenty-first century world. It is possible to steal passwords, loose, and forget. Anyone can misuse the password. As an alternative, nowadays, biometric devices are being used for identifying the people. Biometric device uses the physical and behavioral characteristics of the people for identification. There are two types of biometric identifiers. One is physiological characteristics and the other is behavioral characteristics. Biometric technology recognizes an individual based on fingerprints, signature, face, DNA, typing rhythms, or iris. This provides secure and convenient authentication. These systems are in use across many areas like defense, government, finance, banking, consumer electronics, home safety, healthcare, etc. Internet-based applications like online banking and other e-commerce applications are making biometric as the first and convenient choice. Future scope of biometric in each and every aspect of life is limitless and it is aimed toward improved security and protection of data that is personal. The research in biometric has gained momentum as security needs are increasing for protection of personalised information. Biometrics research is boosted further because of the interest shown in this technology by government, consumers, and other enterprises. Multimodal biometric systems provide better security and convenience to the users compared to traditional methods of authentication. Multimodal biometric systems enhance the accuracy of a recognition system as they consolidate evidences from different biometrics. Multimodal system is a combination of face recognition, finger print verification, voice verification, or any other combination of biometrics. This takes the advantage of proficiency of individual thus overcoming the limitations of a single biometric. Biometric systems are also vulner-

S. Preetha (✉) · S. V. Sheela
B.M.S. College of Engineering, Affiliated to VTU, Bengaluru, India
e-mail: preetha.ise@bmsce.ac.in

S. V. Sheela
e-mail: ssv.ise@bmsce.ac.in

able to attacks. There exists interesting and more specific attacks on several biometric systems. There are some attacks on methods used to index biometric data. There is a need to implement powerful biometric system. In this work, a novel framework of secured multimodal biometric system is proposed.

Keywords Iris · Biometric · Finger print · Multimodal · Wireless sensor network · Security

1 Introduction

In the field of unique identification, an extensive research and development have been taken place in the recent years. The example for biometrics was finger print that was used in China for differentiating children from one another. This is one of the earliest biometrics use and still being used. Biometric systems should be able to provide reliable personal identification methods. There are two ways in which biometric can be used: Identification and Verification. Biometrics can be used for determination of a person's identity without his/her knowledge or consent. An example for this is the usage of face recognition technique using which a person in a crowd can be identified. Biometrics can be used for verification of person's identity. Using biometric verification, one can gain access to an area in a building or get an access to bank account at ATM. Applications of biometric systems include secure banking, security of computer systems, credit cards, social or health services. The purpose of verification of identity is to ensure that only legitimate user has the access to services. Biometric applications like iris, fingerprint, voice, vein-scan recognition considerably increase the security over password use. These types of biometric applications are in use in devices like laptops, mobile phones, and company security systems. The biometric setup usually is depended on one or more biometric processes. The key driver of each of the system is the architecture presented to the user. To measure the biometric system effectiveness, it is significant to know the modalities and better way to use modalities. The continued improvement is possible through research into several interrelated areas. Some of them are briefed here.

| | |
|------------------|---|
| Sensors | Improvement in signal-to-noise ratio, hardware cost reduction, and extending the life expectancy |
| Segmentation | Improvement in the reliability of identifying a region of interest when the biometric characteristics of the user are presented to the system |
| Robust matching | Improving the matching algorithm's performance in the presence of noisy features, imperfect segmentation, and inherent signal variance |
| Reference update | Development of methods to update references so that they can account for variations and the aging of references data in long-lived systems |
| Indexing | Development of partitioning and binning methods for speeding up the searches in large database |
| Robustness | Improvement in security against attacks |

2 Related Works

Kurban et al. [1] states that the increased usage of biometric data expects the systems to work robustly. Biometric systems are expected to give effective results in complex scenarios and falsification. Biometric systems such as face recognition, variables like facial expression, light and reflection make it challenging for identification. It is possible to achieve safety and high performance results with biometric fusion. Researchers have used dataset of Body Login Gesture Silhouettes and Eurocom Kinect face for creation of virtual dataset and later fused with score level. For the database of faces, deep learning framework of VGG face has been used for extracting features. To extract gesture features, energy imaging method was used. The interpretation of results indicated that the face detection with deep learning features achieved better results. Multimodal biometric results reached good matching degree performance and reduced false rate. The conclusion is that the gesture energy imaging could be used for recognition of person. It can also be used for biometric data. Bellaaj et al. [2] opine that biometric systems that are unimodal are random which can be efficient for some contexts but not for all contexts. The recognition performance of biometric systems can be improved by fusing multiple modalities. This reduces the limitations of single modality biometric systems like intentional fraud and universality problem. The universality problem is the inability of the system to capture certain person's data. Several parameters like reduced quality of image, contrast, or noise cause data imperfections in biometric systems. It is possible to handle these redundancies or imperfections by using probabilistic modeling. In this work, the authors have proposed a novel multimodal biometric recognition system that is integration of palm print and fingerprint based on probabilistic modelling approach. The work proposed depends on concepts of possibility theory for biometric modeling features. From image samples, biometric features sets are extracted and analyzed statistically and represented by a possibility distribution. By applying score level data fusion process, the biometric templates from the fingerprint and palm print have been used for decision making. By using CASIA, the public palm print image database and FVC, the public fingerprint database, validation of the proposed method is done.

A multimodal biometric recognition system method was proposed by Wang et al. [3] which is based on complex KFDA. This method uses two phases for generalization of KFDA and classification of fusion feature set, complex LDA and complex KPCA plus. Iris and face biometric models are fused in parallel for algorithm testing. The results of the experimentation showed that the recommended method provides improved result when compared with other conventional multimodal biometric algorithms. Park [4] state that, to address the limitations of unimodal systems in biometric, multimodal systems have been used widely. Multimodal systems also achieve high accuracy of recognition. But, the users feel uncomfortable because several steps are required for data capturing in multimodal systems. They also require user's specific behaviors. In this work, the researchers have proposed an innovative method which consists of finger-vein and fingerprint. The researchers have achieved novelty in four ways. First, finger-vein and fingerprint are captured simultaneously by the device.

Second, the size of the capturing device is very small so that it can be adapted to mobile device. Third, the recognition of the finger print is based on ridge area minutia points and recognition of finger-vein is done on the basis of local binary pattern (LBP). Fourth, the results of fingerprint and finger-vein are combined based on decision level fusion. Results of the experimentation confirmed the usefulness and efficiency of the proposed method.

Rane et al. [5] suggested a multimodal system of biometric that uses palm print and face. The main goal of the work is to enhance the strength of the recognition systems. By using various biometric fusion techniques, both modalities are combined. Implementations of biometric system which are multimodal make use of two algorithms: combination and classification. Classification methods make use of algorithms such as random forest and k-nearest neighbors (kNN). Methods like HOG, DCT, wavelet transformation, etc., are used in confidence-based approaches. Galloway et al. [6] state that although single biometric can be used for authentication, multimodal methods are more reliable. In the work presented, biometric system that is multimodal is developed. The setup consists of WSN that consists of infrared cameras and accelerometers. This enables vein scanning and gait analysis authentication. Proposed method achieved error rates of 13% and 11% for gait and vein, respectively. The error rate is 8% for combined metrics.

Abdulminiuim [7] proposed WSN-based face recognition algorithm that depends on the principles of the unique algorithm for holding the network capacity to the sink node and compress the data pertaining to communication to 89.5%. The proposed method is hybrid and is based on the advantage of Zak transform to offprint the farthest different features of the face and Eigen face method for assorting the acceding to the minimum value. The recognition rate achieved is 100% with least computation time. Liu et al. [8] developed optimal solution for mobile ad hoc networks that depend on multimodal system of biometric. A centralized method is implemented in which traits of biometric are stored on a system which is much secured. At discrete intervals of time, authentication is checked. As a result, there is no need to acquire biometric traits simultaneously. Apart from the previous works explained here, there exists some other works [9, 10] for multimodal biometric systems. It can be observed from the related works that there is a need to improve the efficiency of multimodal biometric systems. Several attempts have been made to design effective multimodal biometric systems [11–18].

3 Proposed Model

In the authentication of wireless multimedia, the widely used method is single biometric. But, it is not secure as it is not free from spoofing and has limited accuracy. In order to resolve this problem, in this work, a multimodal fusion technique is proposed which makes use of fingerprint and iris that uses dynamic Bayesian method. This method exploits feature specificity extraction by a unimodal biometrics

method. It also authenticates the users at the level of decision making. Flexible accuracy and authentication can be achieved when this method is extended to more modal biometric authentication. The results of the experimentation indicate that stability and recognition rate have been greatly improved (5.94% and 4.46%, respectively) when compared to unimodal. It is also observed that there is an increase of 1.94% than general methods which are multimodal for the biometric fusion recognition.

Proposed methodology for hybrid multi-model biometric system consists of iris and fingerprint.

3.1 Initial Research

Voice prints, human faces, irises, finger prints, and finger veins are widely used biological features in use for biometrics of humans. It is more convenient to collect samples of voiceprint, fingerprint, and face. The application rate is also higher. But, in order to perform face recognition, large numbers of face sample banks are needed. The cost of operation and training is also high. In the proposed work, the characteristics of low-cost fingerprint and voiceprint are used as investigation objects.

3.2 Fingerprint Authentication Method

Fingerprint authentication is a four-step process as depicted in Fig. 1.

1. The authentication of images of fingerprint by making use of optical instruments or other such equipments.

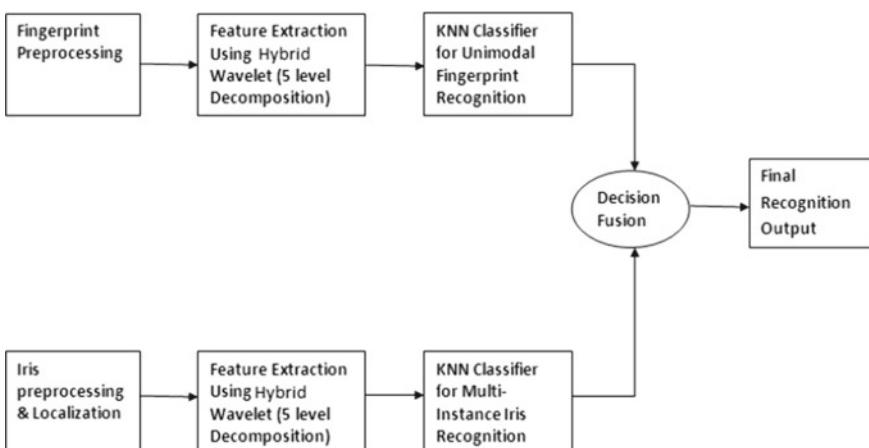


Fig. 1 Proposed multimodal system based on fingerprint and iris

2. Preprocessing of captured image which results in obtaining fingerprint thinning map.
3. Feature extraction of fingerprint which is stored as a template.
4. Matching of fingerprint process which matches the feature template in the database with the feature vector of fingerprint.

3.3 Recognizing Iris

Various subsystems constitute the iris system which corresponds to phases of recognition of iris, respectively. The various phases are as follows.

| | |
|------------------|--|
| Segmentation | This stage locates the iris region in an eye image |
| Normalization | For consistent representation of iris region |
| Enhancement | By equalizing the histogram of normalized iris region |
| Feature encoding | For creation of iris code that consists of most discriminating features only |
| Matching | Matching by hamming distance for making a rejection or acceptance matching |

The first phase in iris preprocessing is isolation of the selected region of iris from the whole eye. This is done by splitting the image between the outer boundary and inner boundary. For detection of edges, Canny method is used which searches for maxima of the gradient of iris image. The computation of the gradient is done using the derivative of a Gaussian filter. This calculates two values as thresholds to reveal weak and strong edges. Canny technique is robust to noise and identifies true weak edges effectively. The strength edge and the orientation image are the outputs of the canny edge detector. The intensity of the image can be enhanced by regulating the gamma correction factor. Local maxima can be suppressed with the orientation image and the regulated gamma image as the input. In order to detect boundaries of iris and pupil, the circular Hough transform can be used. It reveals both center co-ordinates and radius. The complete workflow of the iris recognition system is demonstrated in Fig. 2.

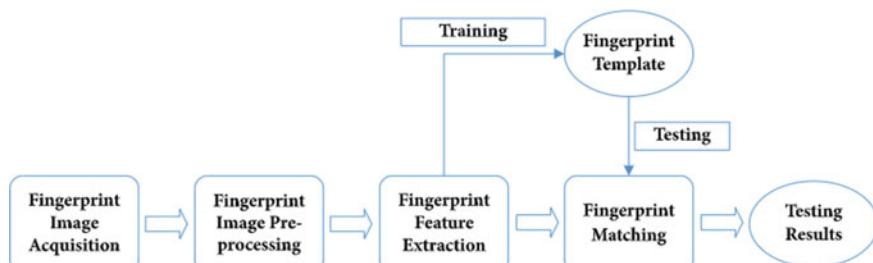


Fig. 2 Flowchart indicating the process of fingerprint authentication

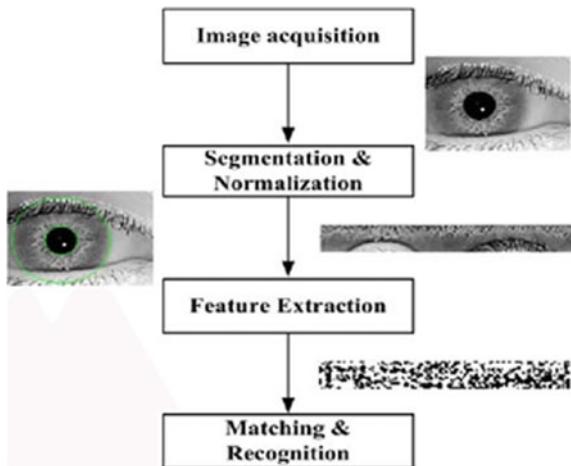


Fig. 3 Iris recognition workflow

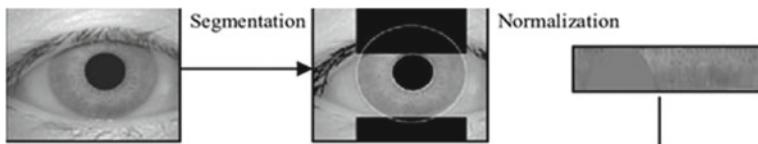


Fig. 4 Iris segmentation with feature extraction and normalization

Iris matching is a procedure for determining whether two iris templates are taken from same person or not. Iris segmentation and feature extraction with normalization are demonstrated in Fig. 3.

Hamming distance is applied for comparison of images bitwise. Masking is done for noise. From iris region, the significant bits are generated for use in Hamming distance calculation between two iris templates (Fig. 4).

3.4 Fusion of Features of Fingerprint and Iris Features

Matching score is provided by each system that indicates feature vector nearness with template vector. Scores obtained can be combined for assertion of the veracity of the claimed identity. The matching scores information is more rich compared to ranks and decisions. It is easier to study and implement compared to image-level and feature-level fusion. The same process can be used in all types of biometric fusion scenarios. Each process of biometric provides its own binary result. They are fused together to output single binary decision of accept or reject.

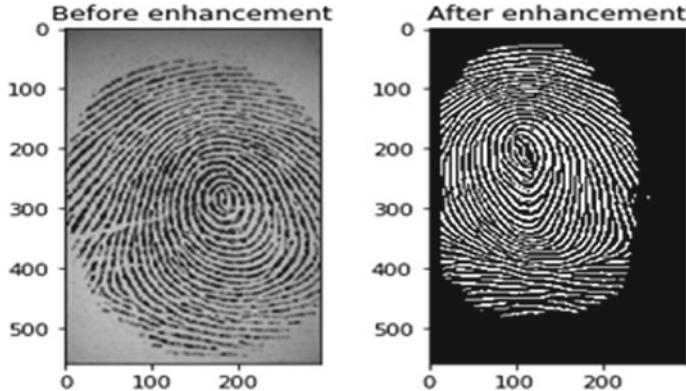


Fig. 5 Images before and after enhancement

3.5 Fingerprint Matching Score Calculation

Before extracting relevant features from the fingerprint images, those images have to be enhanced to get better features as shown in Fig. 1. In this paper, finger print image enhancement based on oriented Gabor filter has been used, and following equations are to extract finger print features from the fingerprint images (Fig. 5).

$$A = \{m_{A_1}, m_{A_2}, \dots, m_{A_p}\}, \quad \text{where } m_{A_i} = \{x_{A_i}, y_{A_i}, cn, \theta_{A_i}\}, \quad 1 \leq i \leq p \quad (1)$$

$$B = \{m_{B_1}, m_{B_2}, \dots, m_{B_q}\}, \quad \text{where } m_{B_j} = \{x_{B_j}, y_{B_j}, cn, \theta_{B_j}\}, \quad 1 \leq j \leq q \quad (2)$$

$$A = \{x_1, y_1, cn, \theta_1\} \quad \text{and} \quad B = \{x_2, y_2, cn, \theta_2\} \quad (3)$$

Co-ordinates of the detail points are symbolized by x and y . Fingerprints are collected when being pressed and hence, it is easy for the collected ones to be offset. Hence, during the authentication process, the geometric limitations on the details of the matching point are proposed. It includes geometric distance and angle of detail deviation from the limit which can be represented as follows.

$$\text{dist}_r(m_{A_i}, m_{B_j}) = \sqrt{(x_{A_i} - x_{B_j})^2 + (y_{A_i} - y_{B_j})^2} < r_\delta \quad (4)$$

$$\text{dist}_\theta(m_A, m_B) = \min(|\theta_A - \theta_B|, 360 - |\theta_A - \theta_B|) < r_\theta \quad (5)$$

Global registration process follows a local search. Matching feature points are the two distinctive points satisfying Formulas (4) and (5). Two fingerprints with sufficient matching feature points are considered to be matched fingerprints (Figs. 6

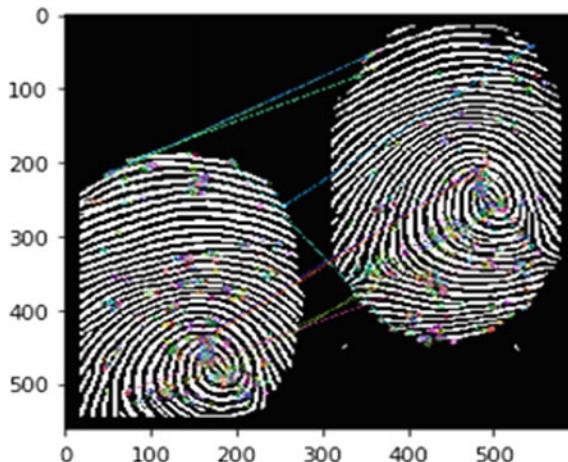


Fig. 6 Fingerprint feature and matching with another fingerprint

and 7). Table 1 represents the comparison of various ML algorithms for fusion-based fingerprint and iris recognition for WSN.

$$\text{sim}(A, B) = \frac{n_{\text{match}}^2}{n_A n_B} \quad (6)$$

Performance Comparison of ML algorithms for Fingerprint and Iris recognition

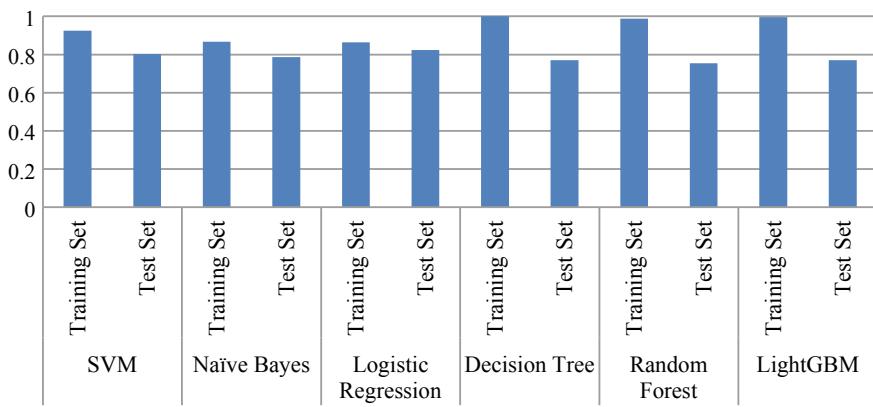


Fig. 7 Comparison of various ML algorithms for fusion-based fingerprint and iris recognition for WSN

Table 1 Performance comparison of ML algorithms for fusion-based fingerprint and iris recognition

| SVM | | Naive Bayes | | Logistic regression | | Decision tree | | Random forest | | Light GBM | |
|------|--------------|-------------|--------------|---------------------|--------------|---------------|--------------|---------------|--------------|-----------|----------|
| | Training set | Test set | Training set | Test set | Training set | Test set | Training set | Test set | Training set | Test set | Test set |
| 0.92 | 0.80 | 0.86 | 0.78 | 0.86 | 0.82 | 1 | 0.7 | 0.98 | 0.75 | 0.99 | 0.77 |

4 Conclusion

This paper contributes to the fusion and application of machine learning algorithms for biometric recognition using both fingerprint and iris images for wireless sensor networks (WSN). This paper briefly discussed regarding the extraction of fingerprint and iris features and fusion at feature level and then application of machine learning algorithms for better recognition rate and same is reported in the graph Fig. 3. As this methodology is basically for WSN, fusion of fingerprint and iris at feature level is very much required to reduce the computational requirement by the system. The proposed method achieved very good accuracy of above 95% with all machine learning algorithms.

References

1. O.C. Kurban, T. Yildirim, A. Bilgiç, A multi-biometric recognition system based on deep features of face and gesture energy image, in *2017 IEEE International Conference on Innovations in Intelligent Systems and Applications (INISTA)*. IEEE (2017)
2. M. Bellaaj et al., Possibilistic modeling palmprint and fingerprint based multimodal biometric recognition system, in *2016 International Image Processing, Applications and Systems (IPAS)*. IEEE (2016)
3. Z. Wang et al., Multimodal biometric recognition based on complex KFDA, in *2009 Fifth International Conference on Information Assurance and Security*, vol. 2. IEEE (2009)
4. Y.H. Park et al., A multimodal biometric recognition of touched fingerprint and finger-vein, in *2011 International Conference on Multimedia and Signal Processing*, vol. 1. IEEE (2011)
5. M.E. Rane, A.J. Pande, Multi-Modal biometric recognition of face and palm-print using matching score level fusion, in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE (2018)
6. B.M. Galloway et al., Multimodal biometric authentication in wireless sensor networks, in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. IEEE (2016)
7. M.E. Abdulminiuim, Propose an efficient face recognition model in WSN based on zak transform. Iraqi J. Sci. **58**(2A), 759–766 (2017)
8. J Liu et al., Optimal biometric-based continuous authentication in mobile ad hoc networks, in *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)*. IEEE (2007)
9. H. Aronowitz et al., Multi-modal biometrics for mobile authentication, in *IEEE International Joint Conference on Biometrics*. IEEE (2014)
10. K. Gunasekaran P. Mahalakshmi, Implementation of multimodal biometric authentication using soft computing techniques, in *International Conference on Information Communication and Embedded Systems (ICICES2014)*. IEEE (2014)
11. A. Soria-Frisch, A. Riera, S. Dunne, Fusion operators for multi-modal biometric authentication based on physiological signals, in *International Conference on Fuzzy Systems*. IEEE (2010)
12. A. Drosou et al., Unobtrusive multi-modal biometric recognition using activity-related signatures. IET Comput. Vision **5**(6), 367–379 (2011)
13. J. Dong et al., Template protection based on DNA coding for multimodal biometric recognition, in *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE (2017)
14. C. Jamdar, A. Boke, Multimodal biometric identification system using fusion level of matching score level in single modal to multi-modal biometric system, in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE (2017)

15. B. Rajalakshmi, S. Sumathi, Survey of multimodal biometric using ear and finger knuckle image, in *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE (2018)
16. W. Kabir, M.O. Ahmad, M.N.S. Swamy, A two-stage scheme for fusion of hash-encoded features in a multimodal biometric system, in *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*. IEEE (2018)
17. S.B. Verma, C. Saravanan, Performance analysis of various fusion methods in multimodal biometric, in *2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES)*. IEEE (2018)
18. M. Kaur, S. Sofat, Fuzzy vault template protection for multimodal biometric system, in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE (2017)

Survey on Object Detection, Distance Estimation and Navigation Systems for Blind People



Bela Shah , Smeet Shah , Purvesh Shah , and Aneri Shah

Abstract Loss of vision is a huge problem that is faced by many of the people around the world either from birth or due to some accident or else disease. Due to this, they face many difficulties while interacting with surrounding. In this paper, we have given a brief case study on the existing systems for object detection, distance estimation and navigation for blind and visually impaired people. Many systems have been developed using the electronic sensors or using the concepts of machine learning and deep learning to assist them. These new techniques are far more efficient and reliable than the prior methods like walking cane, guiding dogs, etc. We have also proposed a system based on machine learning integrated with a voice assistant-mobile-based application and external camera using existing methodologies. It aims to help blind people to identify nearby objects along with their respective distances. To achieve this operation, we will be using existing technologies YOLOv3 and DisNet based on neural networks. The system also makes traveling task from one place to another easier by suggesting the fastest transportation system available at that specific time.

Keywords Object detection · Distance estimation · Navigation · Ultrasonic sensors · Neural networks · Blind people · Visually impaired people

B. Shah · S. Shah (✉) · P. Shah · A. Shah

The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India
e-mail: shahsmeet2804@gmail.com

B. Shah
e-mail: bela.shah-cse@msubaroda.ac.in

P. Shah
e-mail: shahpurvesh511@gmail.com

A. Shah
e-mail: anerishah2000@gmail.com

1 Introduction

Vision is among the most essential human senses which plays the most important role in human life while dealing with adjoining environment. People who have lost their eyes or are visually impaired face have to be reliant on many people for their day to day routine works, going from one destination to another, identifying surrounding objects and in these they often encounter many hassles and, in many cases, eventually result in hurting themselves by colliding with the nearby objects.

There are many applications built for object detection for blind people but they are not able to estimate the accurate distance of the object or in the other hand they are able to estimate the distance of the object but not able to recognize the type of the object. Moreover, there are also different applications of navigation for blind people but they do not support object detection and distance estimation. Hence, through our idea, we are going to attain all the functionalities in a single system. We have proposed a system having an android application and voice assistant as a platform for interacting with the user. The application will provide three major functionalities to the user: (1) indoor navigation, (2) outdoor navigation and (3) travel around city. Indoor navigation can be used when in some close place like room for acknowledging user nearby objects and their distance. Outdoor navigation is GPS-based navigation that can be used while walking from one place to another. And travel around city can be used for using transportation mediums available in that particular city for going to a desired destination.

2 Literature Survey

In the fields of object detection and object recognition, a lot of research is being performed. There are also various methods for finding the distance using sensors but without sensors or using concept of deep learning very few methodologies are developed. Various algorithms, techniques and methods are evolved with different characteristics and can be used according to our requirements.

In paper [1], the authors describe system made for indoor navigation for partially and totally blind people along with object detection done simultaneously. Hardware used is made up of fisheye, depth camera, ultrasonic rangefinder, CPU board along with OST glasses and earphone. For partially sighted, wearable OST glasses are used, and for completely blind people, earphones are used to display guiding and surrounding information. For locating the user, they have used visual SLAM algorithm. To find the route from one place to another, globally shortest path A* algorithm is implemented to the PoI graph generated of the virtual blind road. Virtual blind road is built by a sighted person wearing the proposed navigation device. To detect the distance of the object, an ultrasonic rangefinder is used. A* is the best path-finding algorithm but it has some limitations as it is unable to handle scenarios such as cooperative movements, moving obstacles, changes in map and making judgments

of object sizes, turn radius and dangerous areas [2]. Moreover, it does not produce the shortest path always as it relies too much on heuristics or approximations to find out the heuristic function.

The system proposed in [3] is based on ultrasonic navigation to guide the visually impaired and blind to walk easily with the help of walking cane consisting sensors. The system is made using Arduino Uno board which consists of microcontroller Atmega328P-PU. The system provides two modes to the user—hurdle and fixed. In hurdle mode, ultrasonic sensor is used to detect solid obstacles and water sensor is used to detect liquid obstacles. In fixed mode, the system provides information and direction to move from one place to another by setting a fixed route in blind stick from source to destination location. The measurements of fixed route are embedded in the Arduino Uno board, and the user is notified about the turns through the audio. All the cautions to be given to the user are sent with the help of the Bluetooth sensor to the android application. The android application changes over the text-based notifications into speech and the user are informed about the directions as sound. It also does not detect the type of object.

In [4], the author has designed smart wearable wristwatch to help visually impaired people. It aims to assist blind people while walking in public places and streets. It can keep track of path up to 4 m of the user. Three sensors are used in making the system—(1) ultrasonic sensor—to detect objects in front of the user, (2) accelerometer—to detect the user if he is in the stage of falling and (3) voice recognition—to detect the user voice in case if he needs assistance. When the object is detected in the path of the user alarm sound or vibration is generated. Its intensity increases as user approaches the object. Moreover, the system also consists of GPS which is used to send system location as a phone message to user's family members in an emergency situation.

Systems using ultrasonic sensor to detect the distance of the object have some limitations [5]. Due to wide beam angle of ultrasonic, ultrasonic systems cannot detect the exact location of obstacles. If an object which is to be detected is placed in such a way that the ultrasonic signal is deflected in some other direction rather than being reflected back to the ultrasonic sensor, and in that case, there might be a huge error in distance calculation. If the target object is small, then also it might be difficult to detect its distance by ultrasonic signals.

The system proposed in [6] is also for object detection made using Kinect Depth Camera. The approach is to receive frame from Kinect camera, apply image filtration and histogram equalization techniques on frames converted to 8-bit from 16-bit resolution. The image is then divided into nine sub-images and windowing technique is used to detect the object. The distance of the object is calculated on the basis of brightness of the object, and the volume of the tone determines the distance from the visually impaired people. If the obstacle is closer volume becomes louder and vice versa. The computational cost of sliding window algorithm is very high [7]. Many regions are cropped and convert is run on each of them individually. To make it faster, we can increase window and stride size but accuracy is decreased. It also does not detect type of object. Moreover, the user has to carry laptop with him for all the processing.

The paper proposed in [8] uses deep learning for object recognition. The system uses camera of smartphone having Internet connectivity to capture video. It will send frame after every 3 s to the server. At the server image, segmentation is done in two phases. In the first phase, fuzzy Canny edge detection technique is implemented, and in the second phase, morphological operations are done. After that objects are detected from the image and separate object images are formed. Then, deep neural network is implemented which uses CNN to extract distinct features of various categories of images. The type of object is detected from the features and output of name of object is sent to smartphone. The application on the smartphone converts text to speech and gives information about the object into the hearing aid of a user.

In [9], authors have illustrated system made using techniques like deep learning and CNN. It detects object along with their distance from the camera. System is implemented using Nvidia Jetson TX1 with a Zed stereo camera. Live video stream is captured from stereo vision camera. From video stream, rectified image frames are taken and 3D cloud is formed via disparity calculation. Pertained CNN is used to detect and identify objects. Distance estimation is done by constructing 3D point cloud using triangulation method. Though the proposed system identifies type of object and its distance it is not designed for blind people. But the system can be further improved by some modifications to make it appropriate for blind people.

The system illustrated in [10] is capable of detecting the type of object and its distance from the user. They have used Mini PC, two cameras and battery. All of these components are attached to glasses. In this system, input image is taken from a camera and it is processed with CNN. Feature extraction layer of CNN consists of two layers, convolution layer and max pooling layer. Convolution layer using Gaussian kernel algorithm shrinks the pixel size of image. Max pooling layer is used to maximize the sharpness of image and it is performed after every image convolution process. Now, classification layer of CNN classifies the result of feature extraction layer with the specified dataset value. Distance estimation of object is done with the help of stereovision equation. This equation is used to measure distance of object from camera. The obtained result of object and its distance is converted to sound form and is send to earphones.

3 Proposed System

The system will consist of mobile application having a voice assistant along with an external camera along having net connectivity. The camera can be fixed on spectacles. The application will provide three modes: (1) indoor navigation, (2) outdoor navigation and (3) travel around city. The user will be connected through the server through its unique user ID. At the starting of the application, voice assistant will ask user to select from one of the three modes through voice input. Afterward, the application will respond as per the functionality selected. The user needs to turn on the Bluetooth, Internet and GPS of its mobile. Bluetooth will be used to connect

voice output to headphones, Internet connection will be used to connect application with the server and GPS will be used for outdoor navigation.

3.1 Indoor Navigation

When the user will select this mode, he can wander around home or some fixed place like a room. This functionality will help user to recognize object along with its distance. The camera will start capturing videos and send it to the server. Processing will be done on the video at the server side and a grayscale image will be generated. At the server, with the help of YOLOv3 algorithm, we will be able to detect the type of object and by using DisNet neural network and we will measure the distance of the object. The type of object and its distance will be sent as a message to the application from server and the application will convert this text to audio. For example, if there is a chair in the path of a user at a distance of 2 meters suppose then from server message will be sent to application like “Chair at a distance of 2 meters” and application will convert this text message to audio which will be audible to the user with the help of earphones. If suppose there is cluster of objects in the way of the user, the server will send message accordingly to alert the user, otherwise, the voice assistant will speak all the objects along with their distance making the process cumbersome (Fig. 1).

3.2 Outdoor Navigation

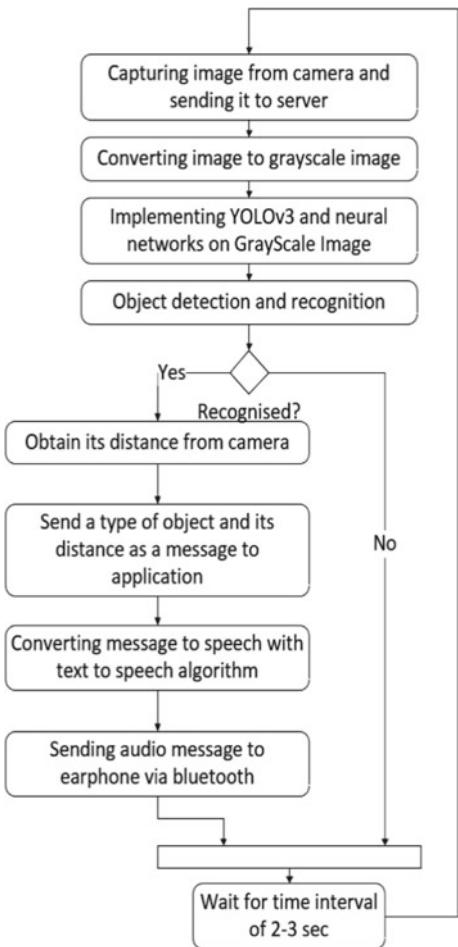
In this mode, the user can walk from its current location to its desired place. The current location of the user will be found using the GPS of the smartphone. The place where the user wants to travel will be taken as a voice input. Afterward, the voice assistant will navigate the user from his current location to the place specified. The voice assistant will also guide the user of nearby objects (Fig. 2).

3.3 Travel Around City

When the user will select this functionality, the server will sort the fastest available transportation on the basis of difference between journey start time and end time. The server will send this information as message to application and it will further be converted into audio form in the application and sent to earphone. The voice assistant will assist the user for the navigation of reaching the specific transportation stop. Moreover, if the user wants, he can keep activated the object detection feature by interacting with the voice assistant (Fig. 3).

Below, we have described an outline of technologies that will be used for making different components of the system.

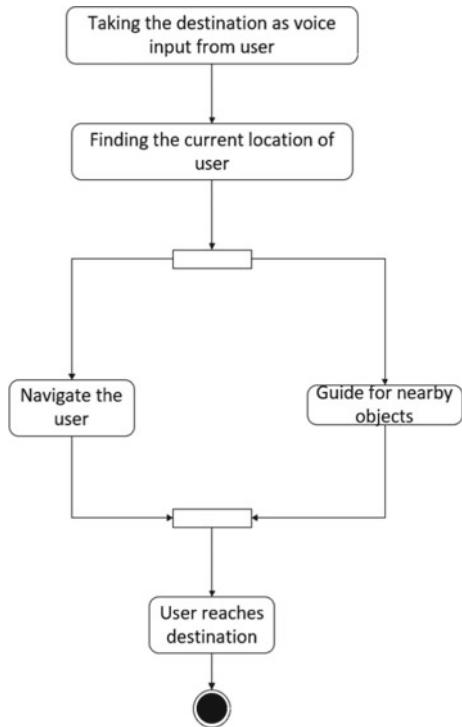
Fig. 1 Workflow of indoor navigation



3.4 Object Detection

For detecting the type of objects, we are going to use YOLOv3 algorithm [11]. The purpose of choosing YOLOv3 algorithm is its accuracy and speed. In this model, a single neural network is applied to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. Logistic regression is used for predicting objectness score of each bounding box. It then performs multilabel classification for objects detected in images. YOLOv3 makes detection at three different scales. It predicts boxes at 3 scale and 3 boxes at each scale, in total 9 boxes. In YOLOv3, 53 convolutional layers are used—Darknet-53.

Fig. 2 Workflow of outdoor navigation

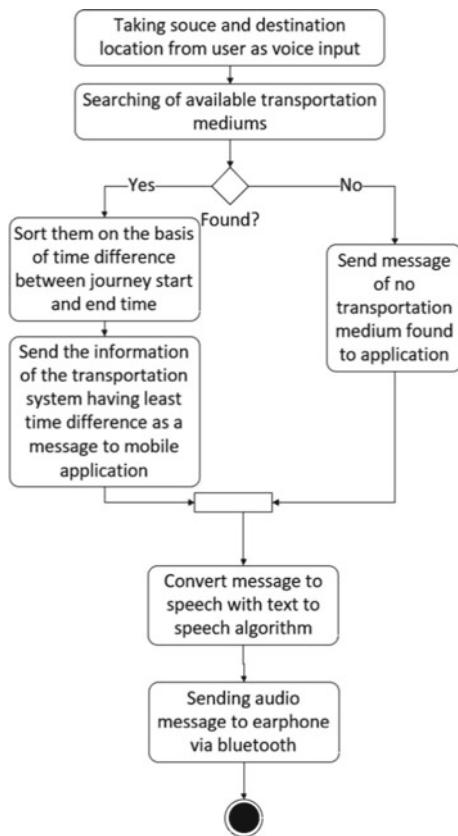


3.5 Distance Estimation

Though there are some techniques existing for finding the distance of object from the camera, we are going to use DisNet technology proposed in [12]. DisNet is based on multihidden-layer neural network. It estimates the distance of the object from monocular camera. From the object, bounding boxes obtained from YOLO are processed to calculate features. With features as input, the trained DisNet predicts distance of the object form the camera. The training of DisNet was done by supervised learning technique where the input features were taken as calculated parameters of the object bounding boxes which were obtained from the YOLO object classifier, and outputs were taken as 3D laser scanner measurements of the distances to object.

The distance obtained will be in the form of meters or centimeters but it can be easily converted into the number of footsteps by simple calculations so that it would be easier for the user to locate the object.

Fig. 3 Workflow of travel around city



3.6 Voice Assistant

For converting the input speech to text, we are going to use Google's speech recognition. Taking the obtained text as an input to the programming, the processing will take place and the system will function accordingly. Text to speech is the opposite process that translates text to a human speech. The voice assistant will use this technique to give speech output of object detected, their distance, navigation details and other information. The noises from the surroundings will make users noise unclear; therefore, noise control will be an important aspect while designing the system.

4 Conclusion

To find out different advanced technologies existing for overcoming the various difficulties faced by blind and visually impaired people while interacting with

surrounding in their routine life was the main aim of this paper. Moreover, how the existing technologies can be better utilized to solve problems further was also of great concern to us. By reviewing many papers and systems, we founded that ultrasonic sensor is widely used in detecting nearby obstacles or for finding their approximate distance. In addition to that different algorithms of machine learning and deep learning are also used for object detection and distance estimation like YOLO, CNN, R-CNN, etc. But it was seen that most researchers mainly focused on obstacle detection rather than finding out its distance from the blind person. Moreover, systems built using different algorithms, like A*, Dijkstra, SLAM, etc., are proposed for indoor navigation. At the end, we have also discussed system integrating all the useful functionalities into one which can be helpful to blind and visually impaired people. With the implementation of proposed system, the blind and visually impaired people will have an aid as the system can predict the distance and type of object they come to contact and moreover it will also suggest the route and the medium of transport to reach from one place to another.

References

1. J. Bai, S. Lian, Z. Liu, K. Wang, D. Liu, Virtual-Blind-Road Following-Based Wearable Navigation Device for Blind People. *IEEE Trans. Consum. Electron.* **64**(1), 136–143 (2018). <https://doi.org/10.1109/TCE.2018.2812498>
2. Limitations of A* algorithm. www.redblobgames.com/pathfinding/a-star/introduction.html
3. R. V. Jawale, M. V. Kadam, R. S. Gaikawad and L. S. Kondaka, “Ultrasonic navigation based blind aid for the visually impaired,” 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, 2017, pp. 923–928. <https://doi.org/10.1109/icpcsi.2017.8391846>
4. Ali Jasim Ramadhan, Wearable smart system for visually impaired people. *Sensors* **18**(3), 843 (2018). <https://doi.org/10.3390/s18030843>
5. Limitations of Ultrasonic Sensors. www.microcontrollertips.com/principle-applications-limitations-ultrasonic-sensors-faq
6. A. Ali and M. A. Ali, “Blind navigation system for visually impaired using windowing-based mean on Microsoft Kinect camera,” 2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME), Beirut, 2017, pp. 1–4. <https://doi.org/10.1109/icabme.2017.8167560>
7. Limitations of Sliding Window algorithm. www.datalya.com/blog/machine-learning/object-detection-with-sliding-window-algorithm
8. N. Parikh, I. Shah and S. Vahora, “Android Smartphone Based Visual Object Recognition for Visually Impaired Using Deep Learning,” 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 0420–0425. <https://doi.org/10.1109/iccsp.2018.8524493>
9. Rahul and B. B. Nair, “Camera-Based Object Detection, Identification and Distance Estimation,” 2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, India, 2018, pp. 203–205. <https://doi.org/10.1109/icmete.2018.800052>
10. R. Bastomi et al., “Object Detection and Distance Estimation Tool for Blind People Using Convolutional Methods with Stereovision,” 2019 International Symposium on Electronics and Smart Devices (ISESD), Badung-Bali, Indonesia, 2019, pp. 1–5. <https://doi.org/10.1109/isesh.2019.8909515>

11. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018)
12. Haseeb, Muhammad Abdul, Jianyu Guan, Danijela Ristić-Durrant, and Axel Gräser. "DisNet: a novel method for distance estimation from monocular camera." 10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS (2018)

Classroom to Industry: A Pathway of a Student to Be an Engineer



K. Rajeshwari, S. Preetha, and H. M. Anitha

Abstract Evolutions in engineering education attempts to witness and inculcate Outcomes-Based Education (OBE) in multiple paradigms. A higher level of knowledge gain is to meet standards across the world. Two such approaches namely Activity Based Learning and Project Based Learning had been experimented with undergraduate students of Information Science and Engineering for course Software Engineering and Object Oriented Modeling. Activity Based Learning is attributed to an optimum learning environment which facilitates effective learning including improvement in soft skills. Project Based Learning is found to enable students to connect engineering concepts with possible solutions and prepares students to be better team players. This paper analyses the challenges of industrial visit and an activity, codeathon on software engineering and object oriented modeling towards the direction of being a stepping stone to trace pathways in the field of engineering education.

Keywords OBE · Activity based learning · Project based learning · Software engineering and object oriented modeling · Industrial visit · Codeathon · Course outcomes · Industry expectations

1 Introduction

Washington Accord is the international accreditation body to accredit professional courses. The accord primarily concentrates on Outcomes-Based Education (OBE).

K. Rajeshwari (✉) · S. Preetha · H. M. Anitha
B.M.S. College of Engineering, Bengaluru, India
e-mail: rajeshwarik.ise@bmsce.ac.in

S. Preetha
e-mail: preetha.ise@bmsce.ac.in

H. M. Anitha
e-mail: anithahm.ise@bmsce.ac.in

India has been a signatory country for this accord since 2014, in the representation form as Nation Board of Accreditation (NBA). The board emphasizes on teaching, learning and development of students' skill sets in a wider angle using learning outcomes. Criterion 1 monitors students' progress, while Criterion 2 ensures the program educational objectives. Criterion 3 discusses the student outcomes; Criterion 4 assesses and evaluates the progress of student outcome. Criterion 5 concentrates on curriculum. Criterion 6 emphasizes on the competency of the faculty, Criterion 7 highlights the facilities and Criterion 8 focuses on the quality of institution infrastructure through financial support, administrative and technical staff. Software engineering focuses on the development of software products and their maintenance. Software engineers build software solutions for real world problems using software engineering principles, methodologies, and best practices. The domain has requirement analysis, design phase, implementation, testing and maintenance. Object-oriented modeling and design is about problems with usage of models for better understanding and representation of real world concepts.

2 Literature Review

The learning outcomes are the graduate attributes which are programme specific, defined by various disciplines. Rayford et al. in [1] discuss how industrial experience helps students to achieve learning outcomes through planning and customer management, team dynamics, and performance, evaluation. These aspects enhanced the technical proficiency, continuous learning, and project management skills in multidisciplinary fields. The industry was keener to hire a team member who can work in odd circumstances, work during stress, ability to upgrade, adapt to new technology. Students when hired to work with industry for a project understand the documentation process, cost factor, time factor and also man hours required to complete a module. Ozdemir et al. in [2] categorizes the forms of industry involvement in academia. The students are motivated through awards/prizes and scholarships which fetch them better employment. Stephens [3] emphasizes projects, internships to promote students' learning outcomes. As there is continuous increase in the graduation rates, it is always challenging for a student to be outstanding. For this, best technical education, soft skills and smart work helps to stand successful. Smith et al. [4] motivates classroom based pedagogy of engagements like cooperative and problem based learning for enhancing students' involvement in their learning. As per their findings, psychological adjustments occur for engineering students. Students feel insecure when they work competitively and individualistically rather than working cooperatively. Perkmann et al. [5] proposed a framework to distinguish industry-academia relationships to explore the characteristics of the collaborative relationships in the perspective of developing research agendas. De Fuentes and Dutrenit [6, 7] focuses on the impact of channels of interaction, which drives the collaboration and benefits between the researchers and the firms. Future of engineering

education programs would represent liberal education, communication, interdisciplinary projects and leadership skills. National Academy of Engineering [8] recommends effective engagements of engineering students in education for next generations. Learning experience should be enriched to improve public understanding and enhance the status of engineering profession. In [9] Analysis of Alumni data and surveys seek out perceptions of graduates. Questionnaire regarding technologies, teamwork, and multidisciplinary projects helped in identifying the skills required for a successful engineer.

3 Activity Based Learning

The Activity Based Learning as described by Yates et al. [10], provides students an exposure to real world processes, the latest technology trends, and the challenges in the existing domain. There is knowledge transfer from the business owners to the students and there is a strong network built between the academia and the industry.

Software Engineering and Object Oriented Design, course was introduced to the curriculum of sixth semester, Information Science and Engineering at BMS College of Engineering in the Academic Year [AY] of 2016–17. The course pedagogy includes theory, laboratory, and self-study component. The course assessment is split into tests, quiz and laboratory along with the self-study reviews. Self-study reviews were spread across the semester. There were three reviews. The first review was Analysis of Business Solutions in current industry avenues and the next two reviews were on Design and Implementation of modeling techniques for system solutions using Object Oriented Programming (OOPs) concepts. The laboratory for the course had an introduction of StarUML 2, trial version software to design the Unified Modeling Language (UML) diagrams. The laboratory had various exercises to solve on class diagrams, object diagrams, use-case diagrams, sequence diagrams, state chart diagrams and activity diagrams. Skill based objective is proposed by Karunasekera et al. [11]. The framework focuses on building the students personal skills, managerial, and engineering knowledge. Skill based framework depicts the students' overall development. The first review was based on the Industrial visit; the students were permitted to visit software, entrepreneurial firms or startup companies for technical exposure. For self-study, teams were framed with size of 2–3 members in each. Felder, et al. in [12], discusses the use of cooperative learning. Students work better in teams than individually. Cooperative learning promotes positive independence; it makes every student of the team accountable for their dos and don'ts. Feedback could be collected for presentation skills and technical writing skills from their peers. Students were asked to form the teams themselves for better compatibility. But as per authors, the teams could be framed by the instructor, in view of their ability and performance of the students. Also they suggest rotating the team members after completion of each assignment, but not too often.

3.1 Challenges Noticed Before the Industrial Visit

Since it was kind of its first experience for the students who were in their pre-final year, there were a lot of contradictory opinions about the implementation of the scheme and as teachers too had a lot of apprehension on the idea. Challenges noticed before the first reviews were

- a. Initially the students were concerned about the possibility of obtaining permission from industrial houses for visits. As per the Non-Disclosure Agreement, many companies would not give the opportunity to enter their campus or even share any of the confidential information. The first impression of the idea went well only with 20% of the students getting a visit opportunity for one person—one company was a tedious task from student perspective. 20% of the students were confident to get the permit as they had some references to seek permission.
- b. Since the response was bleak, the decision was changed to form groups of 2–3 students.
- c. Student groups pursued references to obtain permissions from software firms. Relatives, friends or in a few instances alumni of the college had helped them. Few teams got extended help from faculty members for getting necessary approval.
- d. Before the visit, student groups had gathered information of the company they intend to visit.
- e. To further facilitate the learning process, questionnaires were framed by the course instructors.
- f. The questionnaires helped the students to gather information from the companies' authorized person.

3.2 Methodology

Extended class hours were conducted to explain the software engineering principles and various key terms. Students were briefed on the following topics to further the effective learning by students.

- a. What are the hierarchy levels of job profiles? What are their role plays?
- b. The lifecycle of the software process? How does the requirement phase happen? What is a functional requirement? What are the non-functional requirements to be considered?
- c. What are the intermediate teams? How is the cost and time estimated? How is the buffer time estimated? How are the teams assigned to the task? How are the sprints decided? Is it agile?
- d. How does the solution architect finalize on the design? Whom does he/she consult before doing so?
- e. What are the risk management steps taken? What arrangements are made to continue running application software if the server fails?

- f. What are the development tools? What level of testing do developers implement?
What are the best practices followed by the developers?
- g. What are the stages of testing? What kind of testing is followed? What are the tools used for testing?

3.3 Observations and Feedback

Some of the observations after the review are noteworthy. At the end of the visit the students had clarity of the Software Development Life Cycle (SDLC); many of the companies illustrated the models followed in their organizations. Students had an understanding of how to use a subversion control system (ex: TortoiseSVN, software to manage different versions of their program code or GIT is an open source distributed version control). The company mentors shared knowledge about development tools and testing tools they chose in their teams, most of them being proprietary. Students understood the different levels of testing and the various best practices carried out for development and coding.

At the end of the visit, students were familiarized with the operations of software firms through look and feel rather than reading from a prescribed textbook, or listening to the course instructor in the classroom. As part of feedback we asked the students the following questions and the analysis is as shown in Table 1. Many students were successful in finding the companies well in time, some of them settled with startups; therefore the score is 85%. The companies' domain area mismatch was a problem for some few cases. They got permission to visit non-technical firms like manufacturing biscuits.

Nevertheless they still had answers for the questions given by the instructor, leading to a 90% score. Some students had an opportunity to visit almost all the teams, starting from design, development, testing and maintenance. This is the traditional approach. But 15% of the students felt, they had a mentor who helped them in understanding the working culture of his company through formal talk which disappointed them. 65% found they were still in contact with the company mentors, and can approach them for their internship or job. The Course Outcomes (CO) for the review is analyzing software architecture, design models, testing methods and business solutions. Table 2 discusses the expected outcomes for the evaluation. The outcomes certainly map to a higher level of Programme Outcomes (PO) defined

Table 1 Student feedback on Industrial visit

| Queries | % |
|---|----|
| Was the search for a firm easy? | 85 |
| Did you understand the domain after the visit? | 90 |
| Was the information collected traditionally (walk though across the departments)? | 85 |
| Was a network built? | 65 |

Table 2 Rubrics for evaluating industrial visit

| Rubrics | Outcomes expected |
|--|---|
| Management architecture | Able to understand the organizational structure of the company |
| Software development life cycle | Able to distinguish between various software development frameworks |
| Tools identification in the current industry | Able to identify the existing tools that are present in the current industry and explore their importance |
| Understanding and usage of the tool | Should have deeper knowledge about the usage and comparison with other existing tool |
| Team work | Team achieved the objectives, worked well together |

by the college in line with graduate attributes of National Board of Accreditation (NBA). They are PO9, PO10, and PO12. PO9 is defined as students' performance as an individual or in a group to perform well. PO10 defines communication skills, able to make effective presentations, design documents and effective reports. PO12 is how life-long learning is achieved.

4 Project Based Learning

Project Based Learning gives students a holistic understanding while they carry out a project. The process starts from collecting the information, identifying the requirements, understanding the customer expectations. Many times, the customer requirements are reiterated multiple times. In this entire process students understand the challenges on design, development and deployment issues. There were continuous feedbacks collected from the initial phase of the project towards its completion. This turns out to be time consuming, therefore runs through a semester. Thomas [13] also suggests that knowledge of students gets garnered when they work on multi-disciplinary projects. Thereby the model building details are shared, new challenges open up, which are not their engineering discipline. One such method was experimented for the course of Software Engineering and Object Oriented Design for sixth semester students of Information Science and Engineering. Total number of students was 120 for the AY 2018–2019 and 2019–2020.

4.1 Case Studies on Project Based Learning

Codeathon befittingly marks the beginning of both strategic and creative terms defining, delivering, documenting and refining the learning program which effectively builds both academic and life skills in students. Project based learning is an effort to recognize students and allow modes of delivery as learning becomes more

active, touching real-life skills. Same group of three students continues to experiment with the codeathon activity. Students were asked to perform roles like Consultant, Programmer and Tester as shown in Fig. 1. Mentors or evaluators assign the topic for the codeathon process. Students were allowed to use the Internet, to work for a period of six hours to get the near working model. Consultant reports to the mentor at 9:00 a.m. to select the topic and identify the problem. He understands the problem statement and plans the phases of project along with execution process. Project outline framework is done and briefly explains it to programmers and testers. Programmers identify the requirements (functional and non-functional) and provide solutions with milestones and deliverables. Figure 1 shows the process of Codeathon Software Development Life Cycle. Table 3 shows the evaluation parameters for the Codeathon.

- Around 10:00 a.m., requirement specification, software requirements and hardware specifications along with functional and non-functional requirements were identified and a report was prepared. The consultant presented these details to the mentors seeking further inputs or doubts clarification is done.
- At 2:00 p.m., the initial phase of the project is demonstrated which means the implementation process should be completed.

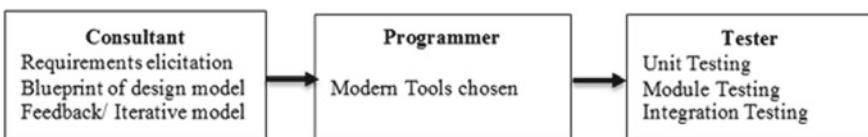


Fig. 1 Role-plays for team members

Table 3 Rubrics for evaluating codeathon

| Rubrics | Outcomes expected |
|---|--|
| Problem analysis | The students takes a business model or a system and analyze all requirements of the system to be developed, elicit the requirements |
| Analysis and synthesis of functional requirement | The students should analyze and synthesize functional requirements of the system |
| Analysis and synthesis of non-functional requirements | Non-functional requirements of the system are analyzed |
| Designing class diagram | The students should evaluate the requirements document artifact and come up with class diagram of the business/system |
| Designing interaction diagram | The students should evaluate the requirements document artifact and come up with identifying use cases and sequence diagram of the business/system |
| Testing | The Student shall produce various test cases |

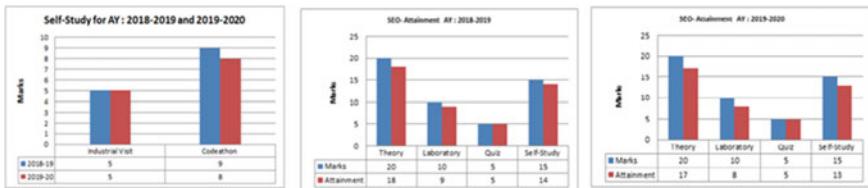


Fig. 2 Attainment for the course

- At around 5:00 p.m., the analyst comes back for a final project demonstration with his teammates. Here the tester would have adhered to testing phases and checked the code for Unit testing, Integration testing.

Outcome of the course can be quantified by student's performance. Also students were able to demonstrate, design, use, increase and produce, innovate, work in a sprint.

The variation in course outcome attainments can be improved by understanding the design principles, discussion of additional assignments and enhancement of Lab exercises. Figure 2 represents the attainment of students for the course during the academic year 2018–19 and 2019–20.

Improvement of the course delivery can be achieved by the following suggestions.

- Industrial visit was planned to give a walk-through of the software engineering principles. Business models were built based on understanding of the object oriented mechanisms.
- Problems related to analyzing the models and applying the design techniques can be solved more rigorously. Industrial visit was conducted to get better observations of the principal activities of a software engineer.

5 Conclusions

Project Based Learning allows students to associate engineering concepts. Problem analysis, design/development of solutions, prepares students to be better players in solving real-world problems. Skill developed among students focuses on modern tool usage, individual team work, communication, project/engineering management. Finally the attitude oriented group comprises the engineer in the society, environment and sustainability, ethics and lifelong learning. Activity based learning is furthering the learning process in engineering in the following ways.

- a. The industry and academia when they work together provide an open platform for the engineering students to excel in their education.
- b. Students' soft skills are well built, starting from their presentation, their behavior dynamism, their email etiquette and finally positive outlook towards odds.

- c. There is a room for finding the various challenges which are unsolved and could be opened to find an optimized solution.
- d. Strong knowledge transfer network is framed, where job openings, internships or assisting on projects or freelancing for student communities can be chained for further links.

Project based learning helps in the following ways

- a. Exposing engineering students to the field realities.
- b. Sensitizing them of their societal role as problem solvers.
- c. Bring in a perspective that engineering education could be used as a tool for societal upliftment through practical applications of engineering concepts.
- d. Enabling students to connect engineering concepts with possible solutions.
- e. Preparing students as better team players.
- f. Emphasizing the importance of recording and documentation of processes.

References

1. R.B. Vaughn, J. Carver, Position paper: the importance of experience with industry in software engineering education, in *Software Engineering Education and Training Workshops, 2006. CSEETW'06. 19th Conference on IEEE* (2006)
2. O. Gol, A. Nafalski, K. McDermott, The role of industry-inspired projects in engineering education. in *Frontiers in Education Conference, 2001. 31st Annual*, vol. 2 (IEEE 2001)
3. Rick Stephens, Aligning engineering education and experience to meet the needs of industry and society. *Bridge* **43**(2), 31–34 (2013)
4. A. Smith Karl et al., Pedagogies of engagement: classroom-based practices. *J. Eng. Edu.* **94**(1), 87–101 (2005)
5. M. Perkmann, K. Walsh, University–industry relationships and open innovation: towards a research agenda. *Int. J. Manage. Rev.* **9**(4), 259–280 (2007)
6. C. De Fuentes, and G. Dutrenit, Best channels of academia–industry interaction for long-term benefit. *Res. Policy* **41**(9), 1666–1682 (2012)
7. B. Harper et al., Liberal education for the engineer of 2020: are administrators on board? in *2010 Annual Conference & Exposition* (2010)
8. Phase II, *Educating the Engineer of 2020: Adapting Engineering Education to the New Century* (National Academies Press, Washington, DC, 2005)
9. I.B. Mena, S.E. Zappe, T.A. Litzinger, Preparing the engineer of 2020: analysis of alumni data, in *2012 ASEE Annual Conference & Exposition* (2012)
10. D.A. Yates, C. Ward, Staying on the cutting edge of student active learning through real world engagement. *Am. J. Bus. Educ.* **2**(1), 101–114 (2009)
11. S. Karunasekera, K. Bedse, Preparing software engineering graduates for an industry career, in *Software Engineering Education & Training, 2007 CSEET'07 20th Conference on IEEE* (2007)
12. M. Felder Richard et al., The future of engineering education II. teaching methods that work. *Chem. Eng. Educ.* **34**(1), 26–39 (2000)
13. J.W. Thomas, A review of research on project-based learning (2000)

ML Suite: An Auto Machine Learning Tool



Nilesh M. Patil , Tanmay P. Rane , and Anmol A. Panjwani

Abstract In today's age, it is important for some businesses to upgrade to machine learning techniques. The aim of this project is to create an autonomous platform for researchers/laymen who operate on data, which would auto-clean the data and suggest machine learning approach to understand and get better value out of data. The proposed project includes a minimalistic user interface and complex backend to cater to user's needs such as better data visualization and summarization, using pandas and matplotlib operations in python. The ultimate goal is to obtain an autonomy such that the user does not have to write a single line of code. The data cleaning has been generalized for different structures of datasets in Django backend. The algorithms are allotted according to target columns which will be set by user in UI. User can try different ML approaches by just choosing options in UI. The project also includes complementary features such as auto web scraping to gather similar advancements in user's research area, Automatic report generation using semantics of natural language processing and dynamic file management. The project also features a chatbot which can be used to query various document and can give answers using semantic querying. This way a layman can be introduced to power of computing using machine learning to speed up businesses and scholar work.

Keywords NLP (Natural language Processing) · Auto-ML · Chatbot · Regression · Classification · Data visualization · Data cleaning · Matplotlib

N. M. Patil · T. P. Rane · A. A. Panjwani

Department of Information Technology, Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai, India

e-mail: nilesh.patil@fragnel.edu.in

T. P. Rane

e-mail: rane.tanmay950@gmail.com

A. A. Panjwani

e-mail: anmolpanjwani0105@gmail.com

1 Introduction

The power of machine learning has outperformed human pattern recognition speed and statistical superiority because of the computation speed of processors. But being a concept which recently came into boom, many people are skeptical to explore its power. There needs to be a platform which simplifies use of machine learning, where the users just have to upload the data, rest is handled by the platform.

Efforts are made in the field of machine learning to stack best solutions by random search, AutoML tools such as Auto-WEKA and Auto-sklearn operate on the known datasets and use Bayesian optimization. The working of machine learning model depends on proper pre-processing of data and tuning of hyperparameters accordingly. AutoML is to free the data scientists from burden of repetitive and time-consuming tasks (e.g. machine learning pipeline design and hyperparameter optimization) so they can better spend their time on tasks that are much more difficult to automate.

The proposed project in this paper is inspired from AutoML and will enable a special suite of interaction with data and streamlining the process of user. The project is designed using completely open source technology which will enable modifications by community. The most complex feature is data cleaning which takes up almost 75–85% of the efforts from user. Because trying to achieve cleaned data without programming would be a time consuming task for layman. If the data cleaning is automated user can focus more on achieving higher accuracies on prediction of data. This enables fast development of ML models to be used on daily basis.

The user creates a simple account on the web portal, which will be used for file management operations such as saving datasets, saving cleaned data, saving trained models for prediction in future. If users allow to track their work the model with highest accuracies and their approaches will be saved and suggested to users with similar datasets. Users can upload and download content project wise in their profile. The web portal consists of auto-clean data command, train command at user's fingertips.

Furthermore, after files are managed user can proceed to gather information about the data uploaded, user can effectively analyze data using analytics provided by pandas and graphs can be plotted on just a click using matplotlib library of python which works at backend of the project. This clears the data understanding part to the user to have a better grasp on data properties, extremities, etc. The target column selection is the most important step which is basically user's wish to choose, from all the columns.

The best feature is one touch button which would clean the data for user by use of custom built framework of suitable cleaning techniques, and in no time user will have hold of cleaned data. The cleaning of data will consist of null removal/substitution, one hot encoding, normalization, feature selection (how much a column affects target column).

The training is based on target column's values; the algorithm automatically selects correct module from sklearn for training the data. User can get accuracy/confidence score of trained model and can save trained model for future

purposes. User can enter other column values in text boxes in frontend and click on predict button by which the target column value will be computed using trained model in backend and showed in frontend.

For the users in Research Area, it is obvious to go through multiple research papers again and again before finally choosing one. In order to avoid the time consumption, there is a conversational chatbot, a tool that uses (NLP) for human-machine interaction (HMI). The chatbot is a document search type, where the intents could be questions regarding a paper and response is obtained by returning the most relative document.

2 Literature Review

The following are brief Review of Literatures On ‘ML Suite: An auto machine learning tool’:

2.1 Auto-WEKA

Auto-WEKA [1] considers the problem of simultaneously selecting a learning algorithm and setting its hyperparameters, going beyond previous methods that address these issues in isolation. Auto-WEKA does this using a fully automated approach, leveraging recent innovations in Bayesian optimization. This is obtained by considering large number of feature selection techniques such as three search and eight evaluator methods with classification approaches from WEKA. Given that it can generate 786 hyperparameters the compute power is needed is very high [2]. Although being a promising technique it does leave a space of overfitting of data as the hypothesis space is large in size. The Auto-WEKA software is in beta mode to find more sophisticated methods for dealing with overfitting. The cross validation technique seems pretty good for a start but can be replaced by better approach. It is also seen that size of dataset is a major factor which influences working of Auto-WEKA, so testing the proposed project on different size of datasets will give clearer idea about the feasibility. The Benchmarks [2] released by open source benchmarking tool indicate Auto-WEKA shows signs of overfitting when running longer and specially seen on multi-class problems. Auto-WEKA shows poorest performances when compared to AutoML in the benchmarks.

2.2 Auto-SKLEARN

It is another AutoML technique consisting of 15 classifiers, 14 feature processing methods and 4 data preprocessing methods. The main trick in this technique is to

ensemble of models by taking into consideration their previous performance on similar datasets. It is the winner of ChaLearn AutoML challenge [3]. The proposed project in this paper uses this auto sklearn technology to provide robust backend and also creates ensembles of models which come from similar datasets. The reason to use this technology is, the Auto-SKLEARN model performs better than Auto-WEKA model in 6 out of 21 cases meanwhile, Auto-WEKA only succeeds in 3 out of 21 cases. The 3 cases were lost by Auto-SKLEARN were because the classifiers chosen by Auto-WEKA aren't implemented in scikit-learn [3]. The Auto-SKLEARN uses meta-learning to improve its accuracy. The meta-learning technique uses main factor as performance of learning algorithms across different datasets. Whenever a new dataset is fed it computes the meta-features of the dataset, then it computes the distance between the meta features of previous trained datasets and just like that the correct classifier is selected for the project.

The scikit learn can provide a wide range of state of art machine learning algorithms using high level language and is easy to use for the task, the module has very less dependencies and can be installed using a simple pip command from python. The documentation is fairly elaborative and simple to reference including a lot of examples to refer. The computation time of sklearn stands apart from mlp, pybrain, shogun and other similar modules when tested on madelon dataset [4]. The model estimators provide an easy yet accurate way to determine the accuracy of models or even error rates using different math concepts.

2.3 Document Search

Usually, people end up spending a lot of time searching documents for their reference work. First, they go to a document store and then they search for relevant documents. To save this time, the chatbot [5] aims to perform document search using Text feature of Amazon Lex [6]. A document store using Amazon S3 is created, where the files will be stored. Using a Lambda function, it is linked with Amazon ES such that whenever a file is uploaded, it gets triggered and the file gets indexed. Another Lambda function is used by Amazon Lex for Fulfillment activity of the chatbot, which too is indexed with Amazon ES. After creating the chatbot, it is integrated [7] with the web application. Amazon Cognito is used to ensure access to the Amazon Lex services through the website. Amazon CloudWatch monitors the AWS services in use. The design [6] below shows the workflow, to implement the Document Search (Fig. 1).

The major AWS services used, along with their purposes, are explained as follows:

AWS S3: Amazon Simple Storage Service (Amazon S3) is the object storage service. A document store [6] is created using Amazon S3 to store all the documents. Also, it is used to host the website where the chatbot appears.

AWS ES: Amazon Elasticsearch Service (Amazon ES) is used for the indexing [6] of documents, which helps in faster access.

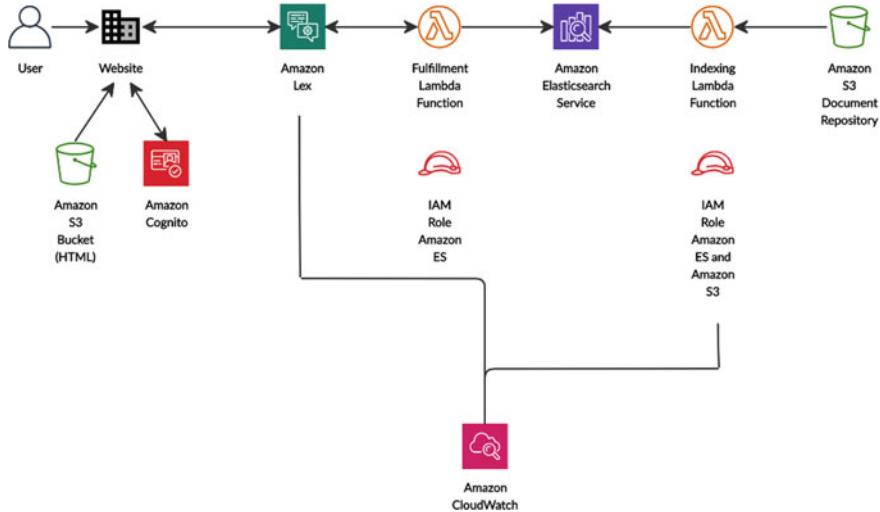


Fig. 1 Design of workflow to build the document-search bot [6]

AWS Lex: Amazon Lex is the service for building conversational interfaces with advanced deep learning functionalities like Natural Language Understanding (NLU). The Bot [5] in this proposed project can be easily used by the user as it contains all the components of a conversation. The user has an Intent [5] to achieve some desirable goals, for which the chatbot is used. For e.g. it can be (“Find answers”, “know meaning”). The Utterance [5] that is a spoken or typed phrase, invokes an intent. So the user can say “I want to know something about (topic name)” or “I want to check a pdf”; these are some simple utterances. And FulfillmentActivity [5] is the business logic that carries out the user’s intent.

AWS Lambda: Amazon Lambda is used to trigger other AWS services. There are 2 Lambda functions used. There’s one Lambda function that triggers Amazon ES whenever an object is created [6] i.e. a document is uploaded in the S3 bucket (document store). Another Lambda function is used for fulfilling the Amazon Lex [6] intent which helps to build the chatbot.

AWS CloudWatch: Amazon CloudWatch is the monitoring service used to ensure that all other AWS services are running smoothly. Every action and its outcome is monitored [6] on Amazon CloudWatch.

AWS CloudFormation and Amazon Cognito: First the AWS CloudFormation is used to launch a stack [7] containing the chatbot parameters and the website URL where it is to be hosted; which is then ultimately embedded [7] in the main web application. Thereafter, the Amazon Cognito service authenticates and authorizes calls to the Amazon Lex service from the web application.

2.4 Text Summarization Using NLTK

Natural language processing can do many tasks like machine translation, email spam filtering, information extraction, question answering system but the most useful application of NLP is Summarization [8]. Given a document to the suite established in this paper it can produce short summary of that document for the user. This can be achieved by using term frequency method to score each sentence after tokenization and creating word frequency table [8]. The method requires user to set a threshold by which the amount of summary needed can be controlled dynamically. NLTK has many functions such as stopword elimination, PorterStemmer, word_tokenize and sentence_tokenize which are building blocks of text summarization.

3 Conclusion

The proposed system has scope in terms of ease of use, better processing speed, and integration with wide range of services. In the wake of examining different techniques about the information mining ideas, the strategies, for example, smoothing, standardization, speculation, total can be utilized for pre-processing the information given in the dataset by client. For dynamic recommendation of research papers, the text mining systems, for example, stop words evacuation and stemming can be utilized. Accordingly, web scratching can be utilized to concentrate papers from Google Scholar utilizing BeautifulSoup. The system is capable of training and predicting from raw data with a layman operating it. The right of choosing the correct algorithm can be given to the system so that layman user can get effective results with real-time training and auto-cleaning of data. The system can effectively answer user's queries on documents and this is useful in today's fast world. The ability to answer the question raised by user regarding a topic in a book or pdf files uploaded by user is time saving and adds weightage to feature applicability to users who are in research or are students. Also adding a touch of text summarization will improve system's suite of tools which are at user's fingertips without explicitly coding.

References

1. C. Thornton, F. Hutter, H. Hoos, K. Leyton-Brown, Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. J. KDD (2012). [arXiv:1208.3719v2](https://arxiv.org/abs/1208.3719v2) [cs.LG]
2. P. Gijsbers, E. LeDell, J. Thomas, S. Poirier, B. Bischl, J. Vanschoren, An open source AutoML benchmark (2019). [arXiv:1907.00909v1](https://arxiv.org/abs/1907.00909v1) [cs.LG]
3. X. He, K. Zhao, X. Chu, AutoML: a survey of the state-of-the-art. Mach. Learn. (2019). [arXiv:1908.00709v4](https://arxiv.org/abs/1908.00709v4) [cs.LG]
4. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher,

- M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
5. R.W. Schmidt, *Learning System Customer Service Chatbot* (Georgia Institute of Technology, Atlanta, 2018)
 6. AWS Machine Learning Blog, <https://aws.amazon.com/blogs/machine-learning/build-a-document-search-bot-using-amazon-lex-and-amazon-elasticsearch-service/>. Last accessed 2018/08/01
 7. AWS Machine Learning Blog, <https://aws.amazon.com/blogs/machine-learning/deploy-a-web-ui-for-your-chatbot/>. Last accessed 2020/04/01
 8. D. Yogish, T.N. Manjunath, R.S. Hegadi, Review on Natural language processing trends and techniques using NLTK, in *2nd International Conference of Recent Trends in Image Processing and Pattern Recognition* ed. by K.C. Santosh, R.S. Hegadi (India, 2018), pp. 589–606

Survey and Gap Analysis on Event Prediction of English Unstructured Texts



Krishnanjan Bhattacharjee, S. ShivaKarthik, Swati Mehta, Ajai Kumar, Rushikesh Kothawade, Paritosh Katre, Piyush Dharkar, Neil Pillai, and Devika Verma

Abstract In the age of big data analytics, text mining on large sets of digital textual data does not suffice all the analytic purposes. Prediction on unstructured texts and interlinking of the information in various domains like strategic, political, medical, financial, etc. is very pertinent to the users seeking analytics beyond retrieval. Along with analytics and pattern recognition from the textual data, there is a need to formulate this data and explore the possibility of predicting future event(s). Event prediction can best be defined as the domain of predicting the occurrence of an event from the textual data. This paper spans over two main sections of surveying technical literature and existing tools/technologies functioning in the domain of prediction and analytics based on unstructured text. The survey also highlights fundamental

K. Bhattacharjee (✉) · S. ShivaKarthik · S. Mehta · A. Kumar
Centre for Development of Advanced Computing (C-DAC), Pune, India
e-mail: krishnanjanb@cdac.in

S. ShivaKarthik
e-mail: shivakarthiks@cdac.in

S. Mehta
e-mail: swatim@cdac.in

A. Kumar
e-mail: ajai@cdac.in

R. Kothawade · P. Katre · P. Dharkar · N. Pillai · D. Verma
Vishwakarma Institute of Information Technology, Pune, Maharashtra, India
e-mail: rushikesh.kothawade@viit.ac.in

P. Katre
e-mail: paritosh.katre@viit.ac.in

P. Dharkar
e-mail: piyush.dharkar@viit.ac.in

N. Pillai
e-mail: neil.pillai@viit.ac.in

D. Verma
e-mail: devika.verma@viit.ac.in

research gaps from the reviewed literature. A systematic comparison of different technical approaches has been listed in a tabular form. The gap analysis provides future scope of more optimized algorithms for textual event prediction.

Keywords Event prediction · Natural Language Processing (NLP) · Text-based prediction · Artificial Intelligence (AI) · Analytics · Unstructured corpora

1 Introduction

Big data can be considered as the field of data science that helps us understand about how large datasets can be processed and analysed in order to extract and link the information from them. It is associated with 5 V's: volume, velocity, variety, veracity and value. Thus, to handle such large sets of data, analysis of this textual data can be considered as an important aspect in predicting the textual data from various sources and systematically link them. Hence, prediction can be defined as a phenomenon which comprises clues and information about possible future events or inference. There are different types of prediction; amongst all the types, focus of this paper is on prediction of events from textual data which have potential to contribute greatly to the various domains. In certain fields like medicine or finance where cause–effect relationship can be established more profoundly, albeit, political domain is such that the causal relationships are not fairly obvious, where event prediction proves to be a difficult task. Again, event prediction can be broadly classified on the basis of temporal, sequential, inductive and deductive types as per types of information clues that lead to probable event. Therefore, algorithms which could achieve this result would have an overall profound impact on predictive text analytics. Experiments in event prediction are being done in different domains like politics, economics, finance, defence, crime, sports, etc. In this way, prediction can serve as a valuable tool for human experts, who can employ different methodologies to determine events prior to their occurrence. However, in order to be credible and determinative in preventing the predicted events from occurring and alter the future, it is essential for such a system to provide plausible reasoning for its forecast.

In this paper, the existing algorithms for political event prediction have been reviewed, and furthermore, certain algorithms for evidence retrieval and scoring have also been surveyed with the aim of elucidating the current state of the field and describe the gaps in current solutions as well as propositions of future work.

Predicting the occurrence of events over political or security domains on textual data has always been an interesting and challenging problem. Linguistically, it is a computing challenge to churn out a prediction from text. Initially, a number of studies have been conducted on different social media feeds, especially Twitter has been explored in multiple ways to predict the outcomes. As proposed by Achrekar et al. [1], Twitter has been used to predict flu or disease trends. Twitter is a platform which provides opportunity for members to address their views. Therefore, as mentioned in [2–4], a significant amount of tweet data was used to predict political

election outcomes. Along with these, tweet data has been used to predict stock market movements [6].

Forecasting and prediction on large amounts of text is particularly a challenge because it is unstructured and ambiguous. Perera et al. [7] have used Probabilistic Soft Logic to represent ontological knowledge and tracked potential cyber-attacks by analysing large volumes of webpage text. In the research work [8–10], large amounts of text have been analysed for prediction of political conflict and political violence. The paper is organized as follows: Sect. 2 discusses the overall goals of the study. In Sects. 3 and 4, the existing literature and tools are described and classified. Finally, Sect. 5 discusses the gaps which highlight the research issues from the surveyed material. Finally, the survey has been concluded and the related future work is laid out in Sect. 6.

2 Goals of the Study

This survey paper focuses upon the future event prediction from unstructured data and analytics over the events. The goals pertain to:

- Gist of existing technical approaches and algorithms for future event prediction from texts
- Identifying the specific features used in these algorithmic implementations
- Review of the existing technical tools, systems and libraries of event prediction and provide gap analysis

3 Literature Survey

This section provides analysis of relevant published papers and technical approaches related to event prediction. The predictive approaches (Refer Fig. 1) are effectively clustered into high-level group of techniques, viz. probabilistic logic, rule-based and machine learning-based approaches. Table 1 summarizes the sources of data and the accuracy results of the approaches considered. It also lists down the important

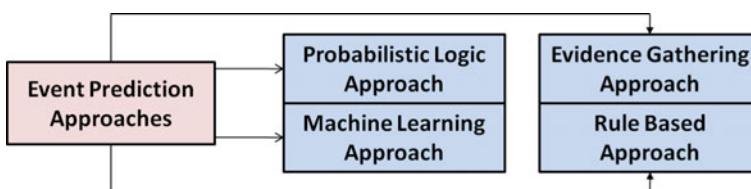


Fig. 1 Event prediction approaches

Table 1 Summary table of approaches

| Approach | Dataset | Features | Result |
|--|---|---|---|
| Probabilistic logic approach (MLN) | DBpedia API | <ul style="list-style-type: none"> • System—Markov Logic Network (MLN) • Event representation—OWL (Web Ontology Language) • Causality inference • First-order logic and Markov logic | <ul style="list-style-type: none"> • Precision: 89.74% • Coverage: 92.26% • Intra-diversity: 84.47% |
| Rule-based approach (Pundit algorithm) | 150 years of news articles, 200 world knowledge ontologies from linked data | <ul style="list-style-type: none"> • System—Pundit Algorithm • Event extraction • Semantic representation • World knowledge • Generalization • Similar event clustering • Builds an abstraction tree • Causality rules generated for prediction | <ul style="list-style-type: none"> • Accuracy: 63% • Human accuracy: 42% • Event extraction precision: 74-100% |
| Machine learning approach (nMIL) | News documents from 6000 agencies, labelled gold standard report (GSR) | <ul style="list-style-type: none"> • System—Nested Multiple Instance Learning (nMIL) • Event detection • Feature extraction and data labelling • Cross-bag similarity • Sequential model • Precursor identification | <ul style="list-style-type: none"> • Average accuracy: 0.709 • Average F1 score: 0.702 |
| Machine learning approach (topic modelling and regression) | 70,000 articles from LexisNexis dataset (January 1975 to December 2015) | <ul style="list-style-type: none"> • System -regression • Topic modelling • Latent Dirichlet Allocation (LDA) • Linear regression model • Neural networks with 10 hidden layers | <ul style="list-style-type: none"> • Civil War ROC curves: 82% • Armed conflict ROC curves: 73% |

deep features that contributed to the overall model building. The outcomes of the approaches are also enumerated together in the same table.

3.1 Probabilistic Logic Approach

The probabilistic logic approach is a technique used to combine the probability theory along with the deductive logic for exploiting the formal arguments. As described in [11], a method has been proposed based on Markov Logic Networks (MLN) [12], wherein complex events are represented by First-Order Logic (FOL). News events are extracted using event representation models and then transformed into Web Ontology Language (OWL). A set of domain-specific causal rules in FOL are also fed into the system, in addition to large-scale ontologies such as DBpedia or WordNet as contextual knowledge. All the information is thus integrated into OWL and used with an MLN to predict news events.

The outcomes derived from this work are:

- Effective method of combining Probability Theory and First-Order Logic
- OWL format-based integration of various types of extracted static and dynamic knowledge from heterogeneous information sources for prediction
- Enrichment of the textual event representation through a deeper semantic structure providing better predictive capability

3.2 Rule-Based Approach

Rule-based approach is traditional linguistics-based method using rules to create predictive models. Pundit algorithm [13] predicts future events based on an input event extracted from news data. The system constructs a knowledge graph using concepts from Wikipedia, ConceptNet [14], WordNet [15], Yago [16] and OpenCyc. It constructs a causality graph, and rules for predicting causality are then generated by finding the predicate projections from cause event to effect event. During prediction, the algorithm searches the causality graph to find the possible effect events for the input event. Lastly, the implausible events are pruned.

The outcomes that can be derived from the above approach are:

- Efficient merging of dynamic, raw and unstructured form of textual data over 150 years via a concept graph
- Proposition of large-scale learning using novel inference-based technique

3.3 Machine Learning Approaches

Machine learning approaches are the techniques that give computers the capability to learn without being explicitly programmed. Based on the fact, Ning et al. [17] explores the use of the nested Multiple Instance Learning, and as further described in [18–20] approach, it not only forecasts significant political event from news articles, but also identifies precursors to those events. The news articles were considered from three countries—Argentina, Brazil and Mexico. A labelled data set about protest events was provided in the form of a Gold Standard Report (GSR).

The outcomes derived from the above approach are:

- A novel application of MIL algorithms requiring a two-level nested structure for event forecasting and precursor modelling
- Strong forecasting performance with varying leading time and historical data
- Demonstration of qualitative richness of identified precursors across different protests in different cities

The paper [5] aims to predict public protests, using the anti-Trump protest as a case study. Tweets are identified using hash-tags calling for demonstration from trending topics on Twitter. The models are validated by predicting protests held in US airports, following President Trump's executive order banning citizens of seven Muslim countries from entering the USA. A major contribution of this paper is the inclusion of event-specific features for prediction, which helps to achieve a high accuracy.

The outcomes derived from this work are:

- High accuracy of all models having TP and TN rate
- Adding event-specific features improved the performance
- Model training on anti-trump protests and testing on Muslim ban protests
- Potential for protest prediction using Twitter features and models

A method has been proposed to utilize Latent Dirichlet Allocation (LDA) [22] to summarize newspaper text on the basis of topic models in the approach described in [21]. Subsequently these interpretable topics are used in panel regressions to predict the onset of conflict. The data has been used from a data set called LexisNexis, which contains 700,000 articles published in three major newspapers, viz. Economist, New York Times (NYT) and The Washington Post (WP).

The outcomes derived from this work are:

- Negative association-based topic models and conflict risk prediction
- Auto-learning based on the changing association of terms adds enrichment of topic models

3.4 Evidence Gathering Approaches

Along with prediction, an important aspect is gathering specific supportive evidence for the predicted event. Popat et al. [23] proposed authors a model for debunking false claims. In order to determine the veracity of a claim, it first searches for articles across the Web relating to that claim. The model then uses the words of the article, the words in the claim, the relevance of the article to the claim, the source of the claim and the source of the article to arrive at a credibility score. Finally, annotated snippets are returned from the articles containing evidence for the conclusion.

Supporting Evidence Retrieval (SER) [24] is a technique in which a query is formed using the initial question and a candidate answer to obtain passages which contain evidence for the candidate solution. It describes the methods by which candidate answers generated by the DeepQA pipeline are scored. After this, four methods are used to score the passages—Passage Term Match, Skip-Bigram, Textual Alignment and Logical Form Candidate Answer Scorer (LFCAS).

The paper [17] focuses on the problem of identifying precursors (evidence) for forecasting significant societal events. It synthesizes event forecasting and precursor identification as a multi-instance learning problem with a nested structure by cities within a country. It detects substantial precursors for different events by calculating a prediction score for each instance in the historic data. Further, it determines a probability for an individual instance occurred to signal a protest (by showing evidence).

4 Technology Survey

4.1 DoWhy by Microsoft

DoWhy [25, 26] is a Microsoft library for causal inference. Causal inference refers to the process of drawing a conclusion from a causal connection which is based on the conditions of the occurrence of an effect. Initially, DoWhy develops a causal graphical model for every problem provided. Secondly, DoWhy uses Bayesian graphical model framework to identify and represent assumptions formally. DoWhy uses Python 3 and uses various supportive libraries like scikit-learn, pandas, networkx, pygraphviz, sympy and matplotlib.

4.2 Tuffy

Tuffy [27] is an Markov Logic Network inference engine focused on data management techniques. It is written in Java and utilizes PostgreSQL and performs the tasks of MRF partitioning (for improving the performance), MAP inference (finding the

highest possibility world state), marginal inference (to estimate marginal probabilities) and weight learning (to learn the weights of MLN rules using training data). Additionally, Tuffy provides functionalities beyond the domain of MLN's such as datalog rules, arithmetic manipulation functions and predicate scoping to name a few.

4.3 IBM Watson

IBM Watson [28] combines Artificial Intelligence (AI) and advanced analytical technology to achieve optimal performance for various analytics tasks. The DeepQA architecture behind Watson consists of syntactic and semantic structure mining, weighted relationship graphs, hypotheses generation and ranking. The results of primary search (candidate answer) are used to generate hypotheses. A hypothesis is a query consisting of a candidate answer and keywords of the original question. This query is used for Supporting Evidence Retrieval (SER). The passages returned through SER are graded according to four evidence scoring algorithms. After merging equivalent hypotheses, the resultant hypotheses are ranked on the basis of the system's confidence and the hypothesis with the greatest confidence is output to the user.

4.4 Event Registry

Event Registry [29] is a tool that analyses news articles and identifies the world events highlighted. The system has the capacity to cluster the articles that describe the similar events. From the articles, the event's core information is extracted such as event, date, location and other named entities. It provides an interactive user interface, search options and visualization dashboard. The event clustering module implements the algorithms mentioned in [30, 31]. All the extracted event information stored in the system database can be searched using the developed API.

5 Gap Analysis

In this section, fundamental research gaps from the survey have been identified. All the approaches and proposed frameworks do not include the adaptability and scalability to parallel and distributed environments. The Markov Logic Network (MLN) framework relies on domain-specific rules devised by human experts. Therefore, the accuracy result drops when non-domain data is provided to the model. In the Pundit algorithm, the events are extracted only from news headlines which pose a requirement of considering data having elaborate news headlines. Also, the algorithm does not consider the effect of time over the causal relations and the forecasted effects.

Overall, the approaches do not consider multiple and heterogeneous data sources which in turn reduces the real-life reliability. For example, the novel nMIL approach was only tested on articles selected from Latin-American countries. Further, the nMIL approach does not utilize regularized multi-task learning. Doing so could have had a significant impact on the performance of the system. Also, the proposed models need to be validated with actually occurred events. Furthermore, extracting event-specific features and inculcating these into the methodology could also improve the accuracy.

In case of social media like Twitter, the location of the tweet or tweeter is also an important feature which should have been considered. The addition of this specific feature might give interesting insights from the analytics. In the case of public protest prediction from tweets, the hash-tags under which the tweets are collected cannot be guaranteed to be relevant. In addition to existing topic modelling techniques, dynamic topic models or structural topic models can also be used to extract events and related probabilistic keywords. It has been shown that validation of data and approaches is an important shortcoming in the surveyed approaches because of high processing needs and costs.

6 Conclusion and Future Work

The comprehensive survey presented here on event prediction from unstructured text falls in the realm of AI-NLP domain. Previously, a lot of work on text prediction has been done on social media like Twitter, but qualitative work on predictive analytics on unstructured sources of streamed data like articles, blogs, news feeds etc. are still necessary to be explored. The lack of structured presentation of data and the need to identify the hidden structure within the data pose huge research challenges.

In the literature survey, a gist of the papers has been presented which proposes novel technical methods to predict events from unstructured text. The reviewed approaches from the papers have been logically organized into four high-level clusters for effective understanding. Further, the approaches like Markov Logic Network (MLN), nMIL and LDA are described according to the clusters. An important aspect in event prediction is to give a prediction that is logically accurate and close to infallible. Therefore, providing specific evidence is a factor that portrays the role of support behind a prediction. Accordingly, papers have been reviewed which propose evidence gathering approaches. In the end, the summary table enumerates the accuracy obtained from the approaches and mentions the datasets used in the work.

The technology survey summarizes existing services, tools and technologies that support predictive analytics. The library Tuffy achieves greater scalability and shows good quality results in a short duration of time. DoWhy stands as a strong base for doing causal inference using traditional graphs.

In the gap analysis section, fundamental research gaps and limitations have been identified and summarized in terms of their scalability, adaptability and effectiveness. The gaps and limitations should be addressed through revisions of the existing frameworks and approaches and by exploring different research methods.

Acknowledgements Extending this study and learning for gap analysis, a new approach is being developed in a research project in Applied AI Group of Centre for Development of Advanced Computing (C-DAC) which aims at predicting events from unstructured text and provides substantial supportive evidence for the same targeting better efficiency and accuracy in textual event prediction.

References

1. H. Achrekar, A. Gandhe, R. Lazarus, S.H. Yu, B. Liu, Predicting flu trends using twitter data, in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (IEEE, Piscataway, NJ, 2011), pp. 702–707
2. B. O'Connor, R. Balasubramanyan, B.R. Routledge, N.A. Smith, From tweets to polls: linking text sentiment to public opinion time series, in *Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM)* (AAAI Press, Menlo Park, 2010)
3. A. Tumasjan, T. Sprenger, P. Sandner, I. Welpe, Predicting elections with twitter: what 140 characters reveal about political sentiment, in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (AAAI Press, Menlo Park, California, 2010), pp. 178–185
4. K. Wegrzyn-Wolska, L. Bougueroua, Tweets mining for French presidential election, in *Proceedings of International Conference Computational Aspects of Social Networks (CASoN)* (IEEE, Piscataway, NJ, 2012), pp. 138–143
5. M. Bahrami, Y. Findik, B. Bozkaya, S. Balcioglu, Twitter reveals: using twitter analytics to predict public protests. Preprint [arXiv:1805.00358](https://arxiv.org/abs/1805.00358) (2018)
6. J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market. *J. Comput. Sci.* **2**(1), 1–8 (2011)
7. I. Perera, J. Hwang, K. Bayas, B. Dorr, Y. Wilks, Cyberattack Prediction through public text analysis and mini-theories, in *2018 IEEE International Conference on Big Data (Big Data)* (IEEE, Piscataway, NJ, 2018), pp. 3001–3010
8. P.A. Schrodte, J. Yonamine, B.E. Bagozzi, Data-based computational approaches to forecasting political violence, in *Handbook of Computational Approaches to Counterterrorism*, ed. by V.S. Subrahmanian (Springer, Berlin, 2013), pp. 129–162
9. M.D. Ward, B.D. Greenhill, K.M. Bakke, The perils of policy by *p*-value: predicting civil conflicts. *J. Peace Res.* **47**(4), 363–375 (2010)
10. M.D. Ward, N.W. Metternich, C.L. Dorff, M. Gallop, F.M. Hollenbach, A. Schultz, S. Weschle, Learning from the past and stepping into the future: toward a new generation of conflict prediction. *Int. Stud. Rev.* **15**(4), 473–490 (2013)
11. S. Dami, A.A. Barforoush, H. Shirazi, News events prediction using markov logic networks. *J. Inf. Sci.* **44**(1), 91–109 (2018)
12. M. Richardson, P. Domingos, Markov logic networks. *Mach. Learn.* **62**(1–2), 107–136 (2006)
13. K. Radinsky, S. Davidovich, S. Markovitch, Learning causality for news events prediction. in *Proceedings of the 21st International Conference on World Wide Web*, (ACM, New York City, 2012), pp. 909–918
14. H. Liu, P. Singh, ConceptNet: a practical commonsense reasoning toolkit. *BT Technol. J.* **22**(4), 211–226 (2004)
15. G. Miller, Wordnet: a lexical database for english. *CACM* **38**(11), 39–41 (1995)

16. F.M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in *Proceedings of WWW* (ACM, New York City, 2007), pp. 697–706
17. Y. Ning, S. Muthiah, H. Rangwala, N. Ramakrishnan, Modeling precursors for event forecasting via nested multi-instance learning, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (ACM, New York City, 2016), pp. 1095–1104
18. S. Andrews, I. Tsachantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in *Advances in neural information processing systems*, ed. by M.I. Jordan, Y. LeCun, S.A. Solla (The MIT Press, Cambridge, 2002), pp. 561–568
19. Z.H. Zhou, J.M. Xu, On the relation between multi-instance learning and semi-supervised learning, in ed. by Z. Ghahramani, *Proceedings of the 24th International Conference on Machine Learning (ICML)*, vol. 227 (ACM, New York City, 2007), pp. 1167–1174
20. Z.H. Zhou, Z. Hua, Multi-instance learning: a survey. Department of Computer Science & Technology, Nanjing University, Tech. Rep (2004)
21. H. Mueller, C. Rauh, Reading between the lines: prediction of political violence using newspaper text. *Am. Political Sci. Rev.* **112**(2), 358–375 (2018)
22. D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
23. K. Popat, S. Mukherjee, A. Yates, G. Weikum, DeClarE: debunking fake news and false claims using evidence-aware deep learning. Preprint [arXiv:1809.06416](https://arxiv.org/abs/1809.06416) (2018)
24. J.W. Murdock, J. Fan, A. Lally, H. Shima, B.K. Boguraev, Textual evidence gathering and analysis. *IBM J. Res. Dev.* **56**(3, 4), 8–1 (2012)
25. A. Sharma, E. Kiciman, DoWhy—A library for causal inference (2019). <https://www.microsoft.com/en-us/research/blog/dowhy-a-library-for-causal-inference/>
26. Microsoft dowhy (2019). <https://github.com/microsoft/dowhy>
27. F. Niu, C. Ré, A. Doan, J. Shavlik, Tuffy: scaling up statistical inference in markov logic networks using an RDBMS, in *Proceedings of the VLDB Endowment*, vol. 4, no. 6 (ACM, New York City, 2011), pp. 373–384
28. R. High, *The Era of Cognitive Systems: An Inside Look at IBM Watson and How it Works* (IBM Redbooks, Mechanicsburg, Pennsylvania, 2012)
29. G. Leban, B. Fortuna, J. Brank, M. Grobelnik, Event registry: learning about world events from news, in *Proceedings of the 23rd International Conference on World Wide Web* (ACM, New York City, 2014), pp. 107–110
30. C.C. Aggarwal, S.Y. Philip, On clustering massive text and categorical data streams. *Knowl. Inf. Syst.* **24**(2), 171–196 (2010)
31. C.C. Aggarwal, P. Yu, A framework for clustering massive text and categorical data streams, in ed. by J. Ghosh, D. Lambert, D. Skillicorn, J. Srivastava, *Proceedings of the Sixth SIAM International Conference on Data Mining*, vol. 124 (SIAM, Philadelphia, Pennsylvania, 2006), pp. 479–483

Human Action Detection Using Deep Learning Techniques



Vedantham Ramachandran, Peddireddy Janaki Rani, and Kalavathi Alla

Abstract The need for automation is exponentially growing in this digital world. The automated detection of human activity has shown profound influence in varied mundane applications in the field of defense, patient monitoring, public security, and computer vision while imparting artificial intelligence. The intent of this work is to analyze the performance of different deep learning algorithms like logistic regression, random forest, decision tree, linear SVC, kernel SVM, and gradient boosted decision tree with grid search for the detection of basic human activities like laying, sitting, standing, walking, walking_upstairs, and walking_downstairs. An experimental set-up made for doing human activity recognition and a comprehensive comparative analysis of results is done. After applying suitable deep learning algorithms, a scrutiny was done by testing the system. The publicly available datasets are used for evaluation of human activity after a significant exploratory data analysis.

Keywords Human activity recognition · Classical machine learning · Deep learning algorithms · Logistic regression · Random forest · Decision tree · Gradient boosted decision tree · Linear SVC · Kernel SVM

1 Introduction

Human action acknowledgment is the issue of ordering successions of accelerometer information recorded by particular PDAs into known all around characterized movements. Classical ways to deal with the issue include hand creating highlights from the time arrangement information dependent on fixed-sized windows and preparing AI models, for example, troupes of choice trees. The trouble is that this element building requires profound skill in the field. Recently, insightful learning strategies [1], for example, repetitive neural systems and one-dimensional convolutional neural

V. Ramachandran (✉) · P. J. Rani · K. Alla

Information Technology Department, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India
e-mail: vrc.bhatt@gmail.com

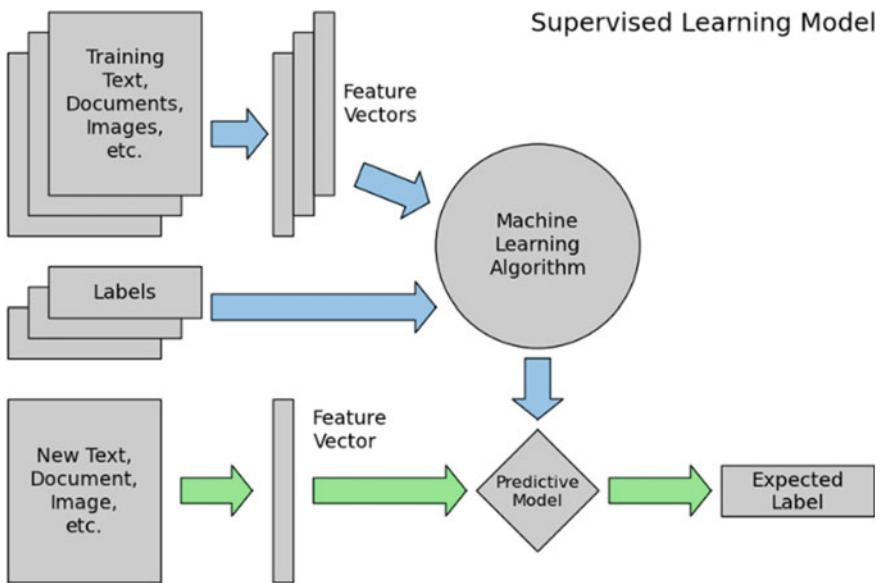


Fig. 1 Supervised learning model

systems, or CNNs, have been appeared to give best in class results on testing movement acknowledgment errands with next to zero information highlight designing, rather utilizing highlight learning on crude information.

Dataset consists of signals from a smartphone carried by 30 individuals performing 6 different basic activities. Activities performed are listed below (Fig. 1).

- Walking
- Climbing_upstairs
- Climbing_downstairs
- Sitting
- Standing
- Laying.

2 Related Work

Machine learning is a utilization of man-made cognizance also called as Artificial Intelligence (AI) that enables systems to normally take in and improve actually without being explicitly modified. Machine learning (ML) being a variant of Artificial Intelligence lets typical applications of software to be more precise in outcome envisaging sans being explicitly burdened to program. Machine learning algorithms use historical data as input to predict new output values. The gait recognition [2] and limb-based recognition [3] works have shown significant importance of machine

learning algorithms. Such Artificial Intelligence routines are incorporated in work for human activity recognition.

The path toward learning begins with recognitions or data, for instance, models, direct understanding, or direction, in order to look for plans in data and choose better decisions later on subject to the models that we give. The fundamental point is to allow the PCs adjust normally without human intervention or help and alter activities as requirements are.

2.1 Logistic Regression

Calculated relapse is a machine logistic regression that is a machine learning calculation which is utilized for the characterization issues, and it is a prescient investigation calculation and dependent on the idea of likelihood. For human pose tracking [4], some strategies explained very well by earlier authors (Fig. 2).

We can consider a logistic regression a linear regression model, yet the logistic regression utilizes an increasingly intricate cost work, this cost capacity can be characterized as the ‘sigmoid capacity’ or otherwise called the ‘strategic capacity’ rather than a straight capacity. The speculation of calculated relapse tends to constrain the cost work somewhere in the range of 0 and 1. Along these lines, straight capacities neglect to speak to it as it can have a worth more prominent than 1 or under 0 which



Fig. 2 Training and testing strategy

is beyond the realm of imagination according to the speculation of strategic relapse.

$$0 \leq h_0(x) \leq 1 \quad (1)$$

2.2 Kernel SVM

In AI, part techniques are a class of calculations for design examination, whose most popular part is the Support Vector Machine (SVM). Part works have been presented for succession information, diagrams, content, pictures, and just as vectors. SVM calculations utilize a lot of scientific capacities that are characterized as the piece. The capacity of part is to accept information as info and change it into the necessary structure. Distinctive SVM calculations utilize various sorts of piece capacities. These capacities can be various sorts, like direct, nonlinear, polynomial, outspread premise work as in Radial Basis Function (RBF) and Sigmoid. Present kernel capacities for grouping information, charts, content, pictures, and just as vectors. The most utilized kind of piece work is RBF. Since it has restricted and limited reaction along the whole x-hub.

The SVM algorithm works by characterizing a thought of similitude, with minimal computational expense even in extremely high-dimensional spaces.

2.3 Decision Tree

Choice tree algorithm is made by Quinlan. In the decision tree procedure, information gain approach is normally used to choose sensible property for each center of a created decision tree. The multi-view Cauchy estimator feature embedding [5] work was effectively used in several works. Right now, it can pick the characteristic with the most raised information gain as the test nature of current center point. At the present time on decision tree algorithm for classification two ways, the information expected to bunch the arrangement test subset obtained from later on isolating will be the most diminutive. As such, the use of this property to divide model set contained in current center point will make the mix level of different sorts for all delivered test subsets lessen to a base. Right now, usage of such an information theory approach will reasonably lessen the required isolating number of thing course of action.

2.4 Random Forest

Random forest is a gathering learning procedure for portrayal, backslide, and various assignments that work by building countless decision trees at getting ready time

and yielding the class that is the strategy for the classes (plan) or mean conjecture (backslide) of the individual trees. Unpredictable decision woods directly for decision trees' affinity for overfitting to their readiness set. The group sparse-based mid-level representation [6] was done in earlier works.

An augmentation of the calculation was created by Leo Breiman and Adele Cutler, who enrolled "arbitrary forests" as a trademark. The expansion joins Breiman's "stowing" thought and irregular choice of highlights, presented first by Hoand later autonomously by Amit and Gemanto build an assortment of trees with controlled fluctuation.

2.5 *Linear SVC*

The target of a linear support vector classifier (SVC) is to fit to the information you give, restoring a "best fit" hyperplane that separates, or arranges, your information. From that point, in the wake of getting the hyperplane, you would then be able to take care of certain highlights to your classifier to perceive what the "anticipated" class is. This makes this particular calculation somewhat reasonable for our utilizations. However, you can utilize this for some circumstances.

2.6 *Grid Search*

Grid search is used to find the optimal hyperparameters of model which results in the most 'accurate' predictions. Grid searching is the process of scanning the data to configure optimal parameters for the given model. Contingent upon the sort of model used, certain parameters are essential. Matrix looking does not just apply to one model sort. Lattice looking can be applied across AI to compute the best parameters to use for some random model. Note that grid looking can be very computationally costly and may take your machine a serious long time to run. Lattice search will manufacture a model on every parameter blend conceivable. It repeats through each parameter mix and stores a model for every mix (Fig. 3).

2.7 *Gradient Boosted Decision Tree*

Inclination boosting is an AI strategy for relapse and arrangement issues, which delivers a forecast model as a troupe of powerless expectation models, normally choice trees. Gradient boosted decision trees (GBDT) is one of the most broadly utilized learning calculations in AI today. It is versatile, simple to decipher, and delivers profoundly precise models. The earlier works that used dense trajectories [7] and group trajectories [8] have shown that the most used methods today are

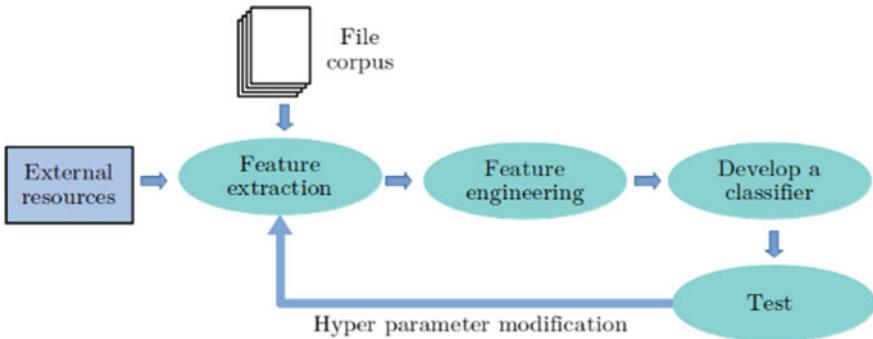


Fig. 3 HAR system architecture

computationally costly and require all preparation information to be in principle memory. As preparing information turns out to be ever bigger, there is inspiration for us to parallelize the GBDT calculation. Parallelizing choice tree preparing is instinctive and different methodologies have been investigated in existing writing. The fisher vector-based strategy for classifying the human action recognition [9] and the semantic works such as action recognition using bag of words [10] stood as other observed works that motivated the selection of acute model in this survey of experiments.

3 Results and Analysis

An extensive set-up for prototyping these proposed methods were done in the constrained environment and they are illustrated down under in this section. The logistic regression, linear SVC, RBF SVM classifier, decision tree, random forest, and gradient boosted DT were experimented in this work, and an impeccable assessment was done to compare their performances.

3.1 Logistic Regression

Through logistic regression, we can strictly classify laying and walking with 100% accuracy and precision, but 12% of sitting points are classified as standing, that is, the reason that recall for sitting and precision for standing is low (Fig. 4).

Fig. 4 Accuracy and confusion matrix obtained after applying logistic regression

```

training_time(HH:MM:SS.ms) - 0:01:31.880353
Predicting test data
Done

testing_time(HH:MM:SS.ms) - 0:00:00.014916

| Accuracy |
-----
0.9626739056667798

| Confusion Matrix |
-----
[[537  0  0  0  0  0]
 [ 1 428 58  0  0  4]
 [ 0 12 519  1  0  0]
 [ 0  0  0 495  1  0]
 [ 0  0  0   3 409  8]
 [ 0  0  0  22  0 449]]

```

3.2 Kernel SVM

Through kernel SVM, 10% of sitting points are classified as standing points. There is also deterioration in walking accuracy when compared to logistic regression (Fig. 5).

3.3 Decision Tree

Through decision tree, 21% of sitting points are classified as standing points (Fig. 6).

3.4 Random Forest

By applying random forest algorithm, 13% of sitting points are classified as standing points (Fig. 7).

Fig. 5 Accuracy and confusion matrix obtained after applying kernel SVM

```
training_time(HH:MM:SS.ms) - 0:03:32.741339
Predicting test data
Done

testing time(HH:MM:SS:ms) - 0:00:03.403906

|-----| Accuracy |-----|
0.9626739056667798

|-----| Confusion Matrix |-----|
[[537  0  0  0  0  0]
 [ 0 441 48  0  0  2]
 [ 0 12 520  0  0  0]
 [ 0  0  0 489  2  5]
 [ 0  0  0  4 397 19]
 [ 0  0  0 17  1 453]]
```

Fig. 6 Accuracy and confusion matrix obtained after applying decision tree

```
training_time(HH:MM:SS.ms) - 0:00:11.165610
Predicting test data
Done

testing time(HH:MM:SS:ms) - 0:00:00.013963

|-----| Accuracy |-----|
0.8632507634882932

|-----| Confusion Matrix |-----|
[[537  0  0  0  0  0]
 [ 0 385 106  0  0  0]
 [ 0  93 439  0  0  0]
 [ 0  0  0 471 17  8]
 [ 0  0  0  16 343 61]
 [ 0  0  0  78 24 369]]
```

Fig. 7 Accuracy and confusion matrix obtained after applying random forest

```

training_time(HH:MM:SS.ms) - 0:05:21.604999
Predicting test data
Done

testing_time(HH:MM:SS:ms) - 0:00:00.245944

-----| Accuracy |-----
0.9141499830335935

-----| Confusion Matrix |-----
[[537  0  0  0  0  0]
 [ 0 428 63  0  0  0]
 [ 0  53 479  0  0  0]
 [ 0  0  0 481 10  5]
 [ 0  0  0  34 339 47]
 [ 0  0  0  35  6 430]]

```

3.5 Linear SVC

By applying linear SVC algorithm, 1% of walking_upstairs points and walking_downstairs are overlapped. Over all accuracy of the system is similar to the logistic regression's accuracy (Fig. 8).

3.6 Gradient Boosted Decision Tree

By using gradient boosted decision tree, 19% of sitting points are classified as standing points (Fig. 9).

3.7 Model Comparisons

See Figs. 10 and 11.

Fig. 8 Accuracy and confusion matrix obtained after applying linear SVC

```

training_time(HH:MM:SS.ms) - 0:00:32.724270
Predicting test data
Done

testing_time(HH:MM:SS:ms) - 0:00:00.007938

-----| Accuracy |-----
0.9674244994910078

-----| Confusion Matrix |-----
[[537  0  0  0  0  0]
 [ 2 431 55  0  0  3]
 [ 0 10 521  1  0  0]
 [ 0  0  0 496  0  0]
 [ 0  0  0  3 412  5]
 [ 0  0  0 17  0 454]]

```

4 Conclusion and Further Work

The automation of human activity recognition under a constrained environment was experimented in this work. The expert feature engineering model was played with classical machine learning algorithms, and the real-time series analysis was experimented with deep learning algorithms. The experiments elicited that, it is better to use linear regression or kernel SVM algorithms for detecting the human activities as it produces more accurate results with less error rate. The metrics used for measuring the accuracy are classification accuracy, confusion matrix, F1-score, and logarithmic loss. Classification accuracy is the ratio of correct prediction to the total number of input samples. F-score is given by $(2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$. It conveys the balance between the precision and recall. Logarithmic loss measures the performance of classification model and the prediction input is a probability value between 0 and 1.

The further work on this model aimed at considering more varieties of human activities, varying the scale and shape of HAR databases and inclusion of optimization techniques for better accuracy.

Fig. 9 Accuracy and confusion matrix obtained after applying gradient boosted decision tree

```
training the model..
Done

training_time(HH:MM:SS.ms) - 0:28:03.653432

Predicting test data
Done

testing_time(HH:MM:SS.ms) - 0:00:00.058843

-----| Accuracy |-----
0.9222938581608415

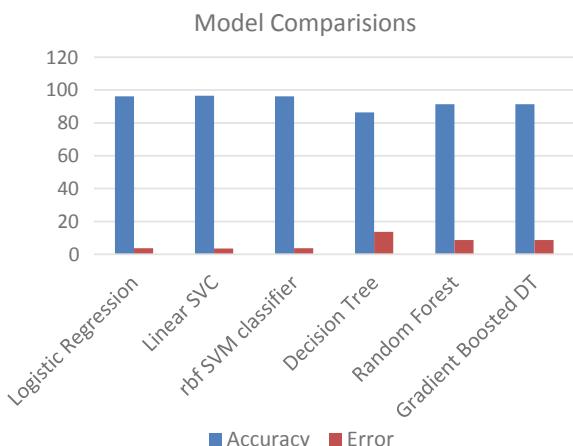
-----| Confusion Matrix |-----
[[537  0  0  0  0  0]
 [ 0 396 93  0  0  2]
 [ 0  37 495  0  0  0]
 [ 0  0  0 483  7  6]
 [ 0  0  0 10 374 36]
 [ 0  1  0 31  6 433]]
```

Fig. 10 Comparison of models

Models Comparisons

| model | Accuracy | Error |
|---------------------|----------|--------|
| Logistic Regression | 96.27% | 3.733% |
| Linear SVC | 96.61% | 3.393% |
| rbf SVM classifier | 96.27% | 3.733% |
| Decision Tree | 86.43% | 13.57% |
| Random Forest | 91.31% | 8.687% |
| Gradient Boosted DT | 91.31% | 8.687% |

Fig. 11 Graph showing comparative study of models



References

1. C. Chen, R. Jafari, N. Kehtarnavaz, Action recognition from depth sequences using depth motion maps-based local binary patterns, in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 1092–1099 (2015)
2. W. Chi, J. Wang, M. Q.-H. Meng, A gait recognition method for human following in service robots. *IEEE Trans. Syst. Man Cybern. Syst.*
3. G. Liang, X. Lan, J. Wang, J. Wang, N. Zheng, A limb-based graphical model for human pose estimation. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(7), 1080–1092 (2018)
4. J. Yu, J. Sun, Multiactivity 3-D human pose tracking in incorporated motion model with transition bridges. *IEEE Trans. Syst. Man Cybern. Syst.*
5. Y. Guo, D. Tao, W. Liu, J. Cheng, Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(4), 617–627 (2017)
6. S. Zhang, C. Gao, F. Chen, S. Luo, N. Sang, Group sparse-based mid-level representation for action recognition. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(4), 660–672 (2017)
7. H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3169–3176 (2011)
8. H. Wang, C. Schmid, Action recognition with improved trajectories, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558 (2013)
9. X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked fisher vectors, in *Proceedings of the European Conference on Computer Vision*, pp. 581–595 (2014)
10. X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition. *Comput. Vis. Image Understand.* **150**, 109–125 (2016)

Deep Learning Methods and Applications for Precision Agriculture



Nilay Ganatra and Atul Patel

Abstract Agriculture is the primary source of basic needs like food, raw material and fuel, which are considered as the basic building blocks for the economic growth of any nation. Agriculture products threatened by various factors including decline in pollinators, various diseases in crops, improper irrigation, technology, scarcity of water and many others. Deep learning has emerged as a promising technique that can be used for data intensive applications and computer vision tasks. It has a great potential and like other domains, it can also apply to agriculture domain. In this paper, a comprehensive review of research dedicated to applications of deep learning for precision agriculture is presented along with real time applications, tools and available datasets. The findings exhibit the high potential of applying deep learning techniques for precision agriculture.

Keywords Deep learning · Convolutional neural network · Precision agriculture

1 Introduction

The research shows that approximately 65% living beings are directly or indirectly depends on the agricultural products. This sector faces various changes to accomplish the needs of growing population which has almost doubled in last 50 years. Increased population and climate changes are considered as the detrimental factors for agriculture to fulfill its necessity. Moreover, the agriculture products threatened by various factors including decline in pollinators, various diseases in crops, improper irrigation, technology, scarcity of water and many others. Pre-emptive measures against

N. Ganatra (✉) · A. Patel
Faculty of Computer Science and Applications, Charotar University of Science & Technology,
Changa, India
e-mail: nilayganatra.mca@charusat.ac.in

A. Patel
e-mail: atulpatel.mca@charusat.ac.in

this factor could be possible with the introduction of the latest technology in agriculture domain. Applying computer vision along with the Internet of things, Deep Learning and Machine Learning, i.e., subfields of Artificial Intelligence, it has been possible to come up with sympathetic solution for the this problem area. Precision agriculture is a field that integrates all these technologies and considered them as an important aspect to deal with the agriculture field challenges like environmental hurdles, food productivity, sustainability, quality [1]. The most widely used sensing method is satellite-based, where infrared cameras and thermal are being used in a smaller but increasing range. Machine learning with image processing was the most popular technology used for analyzing image. Various algorithms of machine learning like K-means, SVM, Artificial neural networks (ANN) and random forest are most popular among others [2]. However, the latest technology used by number of researchers nowadays is Deep Learning (DL) [3]. The deep learning and machine learning are the subdomains of wider family artificial intelligence. It offers various advantages like more learning abilities and increased performance and precision over machine typical machine learning algorithms [4]. This paper covers a specific review of research applications dedicated to deep learning for precision agriculture along with real time applications, tools and available datasets.

2 Overview of Deep Learning

Deep Learning (DL) is an extension of classical Machine Learning by adding complexity into the model as well as data that can be transferred using various functions which allow hierarchical representation of the data through the number of levels of abstraction [5]. The major reason for increased usage of deep learning in various application domains is its feature learning capabilities. Figure 1 illustrates a working of CaffeNet, an architecture of CNN [6].

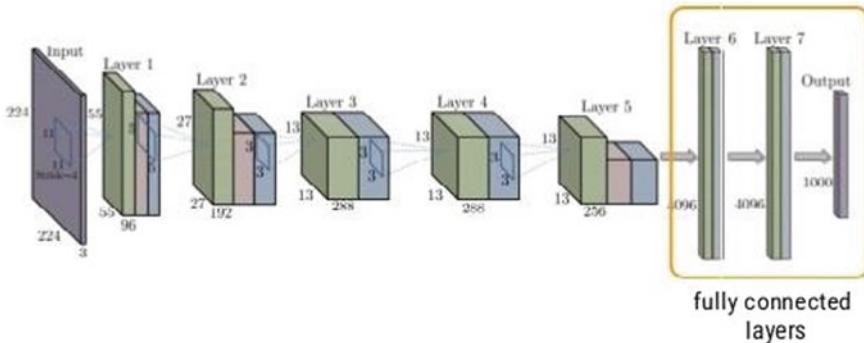


Fig. 1 CaffeNet: an example of CNN architecture [6]

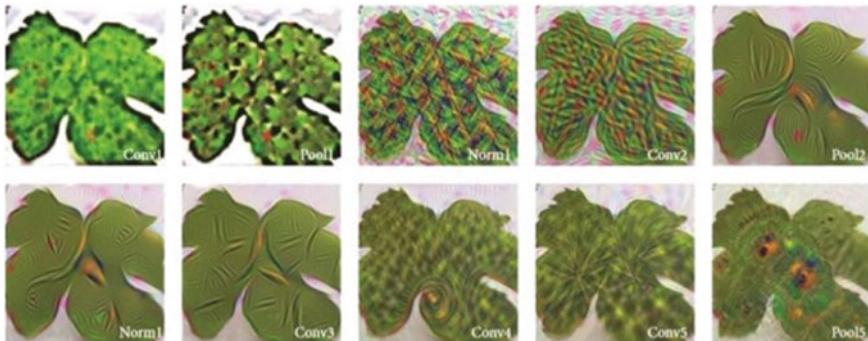


Fig. 2 Output layer visualization after processing respective layer of the CNN: caffenet for plant disease identification [6]

An automatic feature extraction from given raw dataset can accomplish by deep learning algorithms in which lower level features composition is used to form the higher level features hierarchy. Using deep learning algorithms complex problems can be solved more precisely and faster, because of complex structures of the modes used, using which immense parallelization can be done [5]. Convolutions, pooling layers, fully connected layers, activation functions, gates, memory cells etc. are the various components of the deep learning and usage of them depends upon the network architecture used, i.e., Convolutional Neural Networks, Recurrent Neural Networks, Recursive Neural Networks [3]. Convolutional Neural Network (CNN) is considered as most commonly used approach of the Deep Learning which is considered as deep, feed-forward Artificial Neural Network (ANN). As shown in Fig. 2, to create different representation of the given data set, beginning from more common ones at primary layers, fetching more precise at the deeper layers, various different convolutions are performed at required layers of network. The convolutional layers are used to extract the features from the input image and dimensionality of the image then reduced using pooling layers. Convolutional layer extract and encode the low-level features into more specific features for the given problem domain. They are considered as the filters which transform the input image to another image to understand the required pattern. In many cases fully connected layers are placed at the output of the model to apply the task of classification using more specific high-level features obtained from the network to classify the given image with predefined classes or used for regression tasks to produce numerical predictions. Input vector passed as the input to each layer which in turn produces the output vector [7]. Figure. 2 shows the image representation at each layer to the problem of plant disease identification in the CaffeNet CNN.

Table 1 Open dataset of agriculture domain

| Dataset | Description |
|--|---|
| ImageNet dataset | Plants images like trees, vegetables and flowers |
| ImageNet large scale visual Recognition challenge (ILSVRC) | Object localization and identification can be applied on the images |
| EPFL, plant village dataset | Images of variety of crops and their diseases |
| Leafsnap dataset | 185 species of leaves from Northeastern United States |
| University of Arkansas, plants dataset | Dataset with herbicide injury images |
| MalayaKew dataset | 44 classes of species of images |
| Crop/weed field image dataset | Field images crop/weed type with labels |
| Flavia leaf dataset | Images of 32 leaf plants |
| PASCAL visual object classes dataset | Dataset contains animal images like birds, cats, crows, dogs and horses |

3 Deep Learning Architectures, Frameworks and Datasets

There are various successful, tested and trained architectures available which may be used by the researcher to start their model building instead of starting from scratch [8]. The popular architectures used by many researchers are AlexNet, LeNet, ZFNet, GoogLeNet and many more. Each of these architectures has their own benefits and drawbacks and presented in [9] more precisely. Each of the deep learning architecture come with their weights pre-trained, i.e., their network had already trained with some dataset and hence it provides accurate classification for some specific problem domain [5]. Moreover, various dataset available for agriculture are presented in Table 1.

There are various open source tools and frameworks available, using which researcher can perform their experiments. The commonly used frameworks are Theano, TensorFlow, Keras, PyTorch, Caffe and MATLAB tool kit for Deep Learning [10]. Many of these tools like Theano and Caffe incorporates commonly used architectures like AlexNet, GoogLeNet, VGG as either libraries or classes.

4 Deep Learning Application in Agriculture

Deep Learning can be applied to various areas of the agriculture like Species management, field condition management, crop management and livestock management. CNN is the most popular architecture used for such application as mentioned in the many papers done in this paper. However, some researchers have used FCN and RNN. Deep Learning could be applicable to whole growing and harvesting cycle. It starts

with the seed plantation (soil preparation, seeds breeding and water feed measurement) till the robot pick the harvest after confirming the ripeness using computer vision and machine learning.

4.1 Soil Moisture Prediction

Soil is heterogeneous natural resource which provides structural support and considered as the source of water and nutrients for the plant. Using Deep Learning algorithms it is possible to understand the soil moisture and temperature to understand the dynamics of ecosystem and its effect on the various processes of the agriculture.

4.2 Yield Estimation

Yield prediction is the most significant and popular among the researcher for the precision agriculture as it contains various areas like mapping crop supply and demand, crop management and mapping of yield with estimation. Machine Learning and Deep Learning approaches along with the computer vision provided state-of-the-art approaches over the simple prediction based on historical data which allows multidimensional comprehensive analysis of weather conditions, economic situations and crop quality and quantity to make the most of the crop for farmer and population.

4.3 Leaf Classification

There are hundreds of types of trees available in the ecosystem and identifying difference between them is quite difficult. However, for the Botanists it is quite easy to recognize the tree type by using the characteristics of the leaf.

4.4 Disease Detection

To control the pest and disease into the plants or crops pesticides required to spray uniformly over the cropping area. Machine learning and Deep Learning applications helps in making this approach effective by identifying the significant amount of pesticides required which results in improved financial and environment effects.

4.5 Weed Identification

Using computer vision and Deep Learning models it is possible to detect and discriminate weeds in cost effective manner and with no environmental side effects and issues. In many farms robots are available which destroy weeds and reduce the requirement of herbicides.

4.6 Plant Recognition

Compare to human being who uses color and shape for plant classification, Deep learning algorithms provides more effective, quick and accurate result by analyzing lead vein morphology which provides more information about leaf property.

4.7 Fruit Counting

Deep learning algorithms help farmers in knowing exact number of fruits, flowers and trees which is useful for cultivation practices, required labor force for harvesting and disease prevention. The manual counting of flowers and fruits is time-consuming and expensive process.

4.8 Animal Welfare

By observing animals chewing signals by connecting with the animal behavior classifier it provides the guidance to the farmers for need in change in diet or their movement patterns like standing, drinking and feeding. Also, the stress level measurement of the animal is also possible with such systems. Moreover, it predicts its weight gain and proneness to diseases.

Table 2 summarizes the applications of deep learning models in various areas which includes area of usage, dataset used, classes labels, model used, performance metric used and its value.

5 Limitations of Deep Learning

A major barrier to deep learning is the need of huge dataset, which would be used as the input to train any deep learning model. In spite of technique like data augmentation which augments the dataset in label-preserving mode, for real life problem minimum some hundreds of images are required, according to the complexity of the given

Table 2 Applications of deep learning in agriculture

| Agriculture area | Dataset used | Classes and labels | Model used | FW used | Performance metric used | Value (%) |
|---|--|---|--|-----------------------------|-------------------------|---------------------|
| Forecast the soil moisture data over an irrigated corn field [13] | Collected soil data from an irrigated corn field in Northwest China | Scalar value of percentage of soil moisture content | Deep belief network based macroscopic cellular automata (DBNMCA) | Custom developed by authors | RMSE | 6.77 |
| Mapping winter vegetation quality coverage considering time series [14] | Dataset of se Sentinel-1 13 acquisitions in TOPS mode from October 2016 to February 2017 with baseline of 12 days. Dual polarization (VV + VH) data in 26 images | Quality estimation of vegetative development in 5 classes like low, very low, bare soil, average and high | Five-unit LSTM, Gated recurrent unit (GRU) | Keras/heano | CA, F1 | CA-99.05 F1-0.99 |

(continued)

Table 2 (continued)

| Agriculture area | Dataset used | Classes and labels | Model used | FW used | Performance metric used | Value (%) |
|---|--|--|--|------------|-------------------------|-----------|
| Original article international journal of fuzzy logic and Intelligent systems Vol. 17, No. 1, March 2017, pp. 26–34 http://dx.doi.org/10.5391/IJFIS.2017.17.1.26 . ISSN (print) 1598-2645. ISSN(Online) 2093-744X | Dataset consists 8 classes of leaf type with 30% damaged images | Images lanceolate classes like lanceolate, oval, acicular, linear, oblong etc. | CNN using adjusting GoogleNet depth | Custom CNN | CA | CA-94 |
| Plant identification using natural environment [16] [15] | BJFU100 dataset with 10,000 images | 100 plant species | ResNet26 | Keras | PCA CA | CA-91.78 |
| 13 types of various plant diseases classification out of healthy leaves [17] | 4483 images author-created database | Total 15 classes | CaffeNet CNN | Caffe | CA | CA-96.30 |

(continued)

Table 2 (continued)

| Agriculture area | Dataset used | Classes and labels | Model used | FW used | Performance metric used | Value (%) |
|---|---|---|--------------------------|--|-------------------------|---|
| Banana leaves' disease classification [11] | 3700 images dataset from the Plant village dataset | C13 classes | LeNet CNN | deeplearning4j | CA, F1 | CA-96 F1-0.968 |
| 91 types of seed classification [18] | Dataset contains 3980 images of 91 different types of seeds | 91 classes of seeds commonly found in agriculture | PCANet + LMC classifiers | Author developed | CA | CA-90.96 |
| Classification of weed of to 22 different spice [19] | 0,413 images dataset, taken from BBCH | 22 classes of weed species and crops at early stage of growth | Variation of VGG16 | Theano based lasagnelibrary for python | CA | CA-86.2 |
| Identification of 7 different views of plant like entire plant, flower, fruit, leaf, stem and scan [20] | LifeCLEF 2015 dataset with 91,759 images with 13,887 plant observations | 1000 classes of species | AlexNet CNN | Caffe | LC | LC-48.60 |
| Localization and detection of root and shoot features [21] | The dataset with 2500 images with labels of root system and 1664 images of wheat plan with labels | 2 classes to predict root tip & 5 classes for Leaf tips, Leaf bases, ear tips and bases | Custom CNN | Caffe | CA | First dataset-98.4 Second dataset-97.3 |
| Recognition of 44 different plant species [12] | The leaf dataset of MalayaKew with 44 classes. | 44 classes of various spices | AlexNet CNN | Caffe | CA | 99.60 |

(continued)

Table 2 (continued)

| Agriculture area | Dataset used | Classes and labels | Model used | FW used | Performance metric used | Value (%) |
|--|--|--|--|---------|-------------------------|---|
| Fruit count based on apples and oranges images [6] | 71 orange images and 21 apple images | Count of total fruit (scalar value) | CNN and linear regression | Caffe | RFC, L2 | RFC-0.968 L2-13.8 for orange and RFC-0.913 L2-10.5 for apples |
| Sweet paper and rock melon fruit detection [22] | 122 images of type color and near-infrared | Bounding box around sweet red peppers and rock melon | Faster Regionbased CNN with VGG16 mode | Caffe | F1-IoU | 0.838 |

problem. Authors in [11] mentioned that some more number of images is required to improve the classification accuracy. Another limitation is that the deep learning algorithms cannot perform well beyond the boundaries of dataset's expressiveness used for training the deep learning model. For an instance plant recognition in [12] was affected by the environmental parameters like insect damage and wrinkled on surface. A more common issue with deep learning application in computer vision is data pre-processing becomes time-consuming task sometimes especially with the satellite or aerial images where data is high dimensional and training samples are limited. Finally, in agriculture domain very few open datasets are available on which deep learning application can be applied. So, many applications demands researcher to develop their own datasets which is time-consuming and demands lot of efforts.

6 Future Usage of Deep Learning Technology for the Advancement in Agriculture Domain

In this paper, we highlighted many existing application of computer vision where deep learning is already applied in agriculture. The various applications which are discussed here are like crop type classification, weed identification, fruit grading and counting and many more. However, there are many areas of agriculture where deep learning could be applied to better results in the domain, such as water stress detection in the plant, pest detection, deficiency in the food, monitoring of green-house, water erosion monitoring. Also, other possible area where deep learning could be applicable is image capturing via drones, i.e., aerial imagery to monitor the seeding process effectiveness, identifying right locations and moments for best maturity levels, animals and their movements. Moreover, application which can predict plant growth, yield prediction, water needs assessment to avoid the diseases, climate change prediction, and weather condition and phenomenon predication.

7 Conclusion

In this paper, we have performed a survey on effort applied in the field of agriculture by the deep learning. We have examine the models used, available data sources, performance measures, classes or regression values obtained from the various research papers. Based on the survey we identified that deep learning outperforms than other popular techniques like image processing and traditional machine learning. Moreover, we have presented the areas of agriculture where this modern technique has not yet been applied sufficiently. The main aim of the paper is to encourage the research community to experiment with deep learning to solve various problems of agriculture such as prediction, classification, computer vision and image

analysis and data analysis to provide the better solution in favor of smarter, sustainable and secure farming.

References

1. K. Jha, A. Doshi, P. Patel, M. Shah, A comprehensive review on automation in agriculture using artificial intelligence. *Artif. Intell. Agric.* **2**, 1–12 (2019)
2. L.P. Saxena, L. Armstrong, A survey of image processing techniques for agriculture, in *Asian Federation for Information Technology in Agriculture, Perth, W. A. Australian Society of Information and Communication Technologies in Agriculture*, pp. 401–413 (2014)
3. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–44 (2015)
4. D. Li, D. Yu, Deep Learning: methods and applications. *Found. Trends Sig. Process.* **7**(3, 4), 197–387 (2014)
5. S.J. Pan, Q. Yang, A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010)
6. S.W. Chen, S.S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C.J. Taylor, V. Kumar, Counting apples and oranges with deep learning: a data driven approach. *IEEE Robot. Autom. Lett.* **1**–1 (2017). <https://doi.org/10.1109/lra.2017.2651944>
7. A. Kamilaris, F.X. Prenafeta-Boldu, Deep learning in agriculture: a survey. *Comput. Electron. Agric.* **147**, 70–90 (2018)
8. M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, Deep learning applications and challenges in big data analytics. *J. Big Data* **2**(1) (2015)
9. A. Canziani, A. Paszke, E. Culurciello, An analysis of deep neural network models for practical applications (2016)
10. S. Bahrampour, N. Ramakrishnan, L. Schott, M. Shah, Comparative study of caffe, neon, theano, and torch for deep learning. *arXiv* (2015)
11. J. Amara, B. Bouaziz, A. Albergawey, A deep learning-based approach for banana leaf diseases classification, in B. Mitschang, D. Nicklas, F. Leymann, H. Schöning, M. Herschel, J. Teubner, T. Härdter, O. Kopp, M. Wieland (Hrsg.), *Datenbanksysteme für Business, Technologie und Web (BTW 2017)*—Workshopband. Bonn: Gesellschaft für Informatik e.V. (S. 79–88); X. Wang, C. Cai (2015). Weed seeds classification based on PCANet deep learning baseline. pp. 408–415 (2017). <https://doi.org/10.1109/apsipa.2015.7415304>
12. M. Rahnemoonfar, C. Sheppard, Deep count: fruit counting based on deep simulated learning. *Sensors* **17**, 905 (2017)
13. X. Song, G. Zhang, F. Liu et al., Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model. *J. Arid Land.* **8**, 734–748 (2016)
14. H.T. Dinh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, P. Maurel, Deep recurrent neural networks for mapping winter vegetation quality coverage via multi-temporal SAR Sentinel-1 (2017). *arXiv arXiv:abs/1708.03694*
15. W.-S. Jeon , S.-Y. Rhee, Plant leaf recognition using a convolution neural network, in *Int. J. Fuzzy Logic Intell. Syst.* **17**, 26–34 (2017)
16. Y. Sun, Y. Liu, G. Wang, H. Zhang, Deep learning for plant identification in natural environment. *Comput. Intell. Neurosci.* **4**, 1–6 (2017)
17. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* **6**, 1–11 (2016)
18. M. Dyrmann, H. Karstoft, H.S. Midtbø, Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **151**, 72–80 (2016)
19. M. Šulc, J. Matas, Fine-grained recognition of plants from images. *Plant Methods* **13** (2017)
20. M.B. Pound et al., Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *GigaScience* **6**, (2016)

21. S.H. Lee, C.S. Chan, P. Wilkin, P. Remagnino, Deep-plant: plant identification with convolutional neural networks, in *IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, 2015, pp. 452–456 (2015)
22. Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* 7(6), 2094–2107 (2014)

Object Detection with Convolutional Neural Networks



Sanskruti Patel and Atul Patel

Abstract During the last years, a noticeable growth is observed in the field of computer vision research. In computer vision, object detection is a task of classifying and localizing the objects in order to detect the same. The widely used object detection applications are human–computer interaction, video surveillance, satellite imagery, transport system, and activity recognition. In the wider family of deep learning architectures, convolutional neural network (CNN) made up with set of neural network layers is used for visual imagery. Deep CNN architectures exhibit impressive results for detection of objects in digital image. This paper represents a comprehensive review of the recent development in object detection using convolutional neural networks. It explains the types of object detection models, benchmark datasets available, and research work carried out of applying object detection models for various applications.

Keywords CNN · Single-stage object detection · Two-stage object detection

1 Introduction

During the last years, a noticeable growth is observed in the field of computer vision research. Employing machine learning methods provides robust solution to solve computer vision tasks. In computer vision, object detection deals with detecting instances of objects from a particular class in a digital image or video [1]. It is a task of classifying and localizing the objects in order to detect the same. It determines the location where the object is presented in the image and scales one or more objects [2].

S. Patel (✉) · A. Patel

Faculty of Computer Science and Applications, Charotar University of Science and Technology,
Changa, India

e-mail: sanskrutipatel.mca@charusat.ac.in

A. Patel

e-mail: atulpatel.mca@charusat.ac.in

Object detection pertains to identify all objects presented to an image irrespective of that location, size, rendering, etc. Further information like class of an object, recognition of an object, and object tracking is gained once the object is detected accurately.

Object detection mainly comprises of two tasks: object localization and classification. Object localization determines the location and scale of one or more than one object instances by drawing a bounding box around it. Classification refers to a process to assign a class label to that object. For detection, object detection systems construct a model from a set of training data and for generalization, it is required to provide huge set of training data [3, 4]. In last decade, artificial intelligence made an impact in every field of human life and deep learning is a field of artificial intelligence that uses artificial neural network for representation learning. In the wider family of deep learning architectures, convolutional neural network (CNN) made up with set of neural network layers is used for image processing and computer vision [5]. It is having an input layer, a set of hidden layers, and an output layer. CNN takes an image as an input, processes it, and classifies it under certain category. The application of CNN for object detection was applied on years where an arbitrary number of hidden layers used for face detection [6]. With the availability of large datasets, increased processing capabilities with availability of graphical processing unit (GPU), deep CNN architectures exhibits impressive results in the field of image classification, recognition, and detection of objects in digital image [7]. The object detection models briefly perform following operations: (a) informative region selection (b) feature extraction (c) classification. The paper represented the application of various deep learning techniques based on convolutional neural network (CNN) for object detection.

2 Object Detection Models

With the increase number of usage of face detection systems, video surveillance, vehicle tracking and autonomous vehicle driving, fast and accurate object detection systems are heavily required. The output of object detection normally has a bounding box around an object with the determined value of confidence. Object detection can be single-class object detection, where there is only one object found in particular image. In multi-class object detection, more than one object pertaining to different classes is to be found [8]. Object detection systems mostly rely on large set of training examples as they construct a model for detection an object class. The available frameworks for object detection can be categorized in two types, i.e., region proposal networks and unified networks [1]. The models based on region proposal networks are called as multi-stage or two-stage models. The unified models are called as single-stage models. The multi-stage models perform the detection task in two stages: (i) The region of interest (ROI) is generated in first stage and (ii) classification is performed on these ROI. Two-stage object detectors are accurate but somewhat slower. Single-stage detectors are faster than two-stage detectors as less computational work is needed

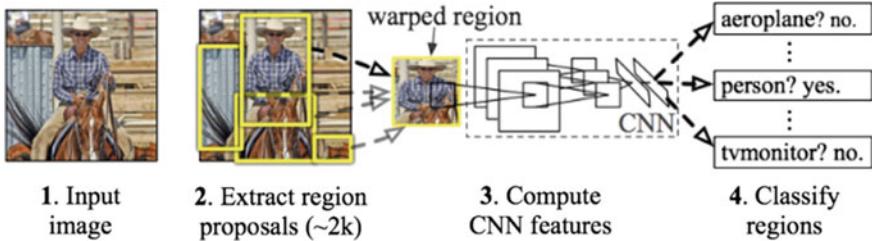


Fig. 1 R-CNN: a region-based CNN detector. *Image source* Girshick et al. [10]

to carry out. On the contrary, two-stage detectors are slow but more accurate. Mask R-CNN, R-CNN, Fast R-CNN, and Faster R-CNN are two-stage object detectors. YOLO and SSD are single-stage object detection models. Two-stage models detect the foreground objects first and then classify it to a specific class, whereas single-stage detector skips the detection of foreground objects and takes uniform samples of objects through a grid [9]. The following section describes the single-stage and two-stage detectors models briefly.

3 Two-Stage Detectors

3.1 Region-Based Convolutional Neural Network (R-CNN)

R-CNN, a short form of region-based convolutional neural network, is one of the most widely used object detection model that falls under two-stage object detectors. Girshick et al. [10] proposed R-CNN that is a first region-based CNN detector as shown in Fig. 1. R-CNN uses a selective search method that generates 2000 regions from the image, called region proposals. These regions are input to CNN that produces a 4096-dimensional feature vector as an output. SVM is applied on this generated vector to classify the objects. Moreover, the bounding box is also drawn surrounding to an object. The major problem with R-CNN is its speed as it is very slow. Also, selective search, the algorithm used to generate the proposals, is very fixed that discards the possibility of learning. The major drawbacks are selective search algorithm proposes 2000 regions per image; for each region of image, it generates CNN feature vector and there is no shared computation between these stages [11]. R-CNN obtained a value of 53.3% of mAP which is significant improvement over the previous work on PASCAL VOC 2012 [1].

3.2 Fast R-CNN

A faster version of R-CNN was released after a year by Girshick [12]. R-CNN uses convolution on each region proposal and it takes more time to complete the detection

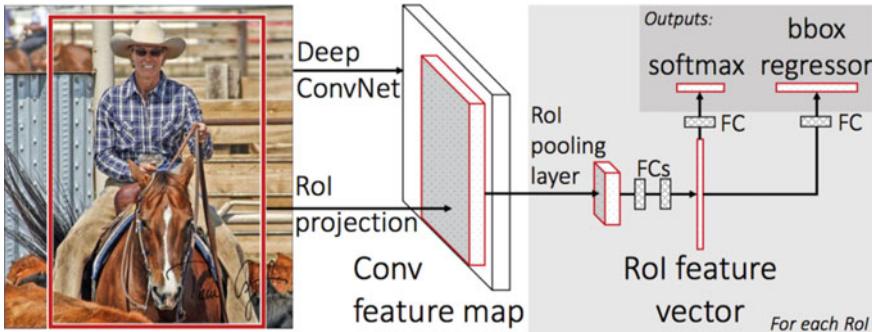


Fig. 2 Fast R-CNN: a faster version of R-CNN. *Image source Girshick [12]*

process. Fast R-CNN uses the concepts of region of interest (ROI) that reduced the time consumption. ROI pooling layer is used to extract a fixed-sized feature map with fixed height and width. For converting features from the regions, it performs the max pooling operation. Before applying a max pooling operation, the $h \times w$ ROI window is divided into set of small and fixed sub-windows, i.e., $H \times W$. The size of each generated sub-window in the grid is $h/H \times w/W$. Experimental results showed that Fast R-CNN obtained a mAP score of 66.9% whereas; R-CNN obtained 66.0%. The experiment was conducted on PASCAL VOC 2007 dataset [13]. It uses VGG16 as a pre-trained network model and follows the shared computation that makes the Fast R-CNN fast. It combines three independent processes that shared computational task. R-CNN extracts CNN feature vector from each region proposal, whereas Fast R-CNN group them and only one CNN forward pass is generated. Moreover, the same feature map is used by classifier and regressor, i.e., bounding box shown in the following Fig. 2.

3.3 Faster R-CNN

Faster R-CNN introduced a concept of a region proposal network (RPN) introduced by Ren et al. 2016 [14]. It replaces the slow selective search algorithm used in Fast R-CNN with RPN, a fully CNN that predicts the region proposals. First, a set of anchor boxes, which are in the form of rectangle, are generated around the object. In second step, loss functions are applied to calculate the likelihood of an error. At last, the feature map is generated by backbone network and RPN proposes set of region proposals. These set of proposals are sent as an input to the next layer, i.e., RoI pooling layer. ROI pooling layer converts the features obtained from the fine-tuned CNN layer to a fixed-sized of feature maps. At last, classification layer predicts the class, while bounding box regression creates the rectangular box surrounded to an object for localization. A mAP score of 69.9% is achieved on PASCAL VOC 2007

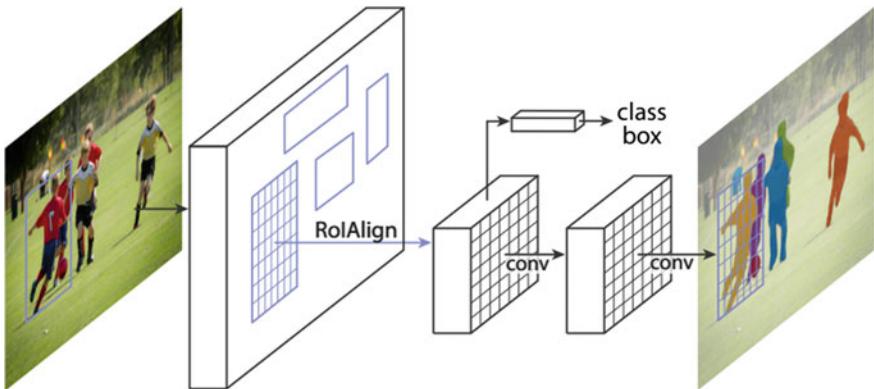


Fig. 3 Instance segmentation using mask R-CNN. *Image source* He et al. [15]

test set, which is a significant improvement over Faster R-CNN for both detection and prediction efficiency [14].

3.4 Mask R-CNN

Faster R-CNN is extended by Mask R-CNN that focuses on instance segmentation from an image and introduced by He et al. [15]. It is an extension of Faster R-CNN and in addition to the class label and bounding box, it also generates the object mask. The accurate detection is essentially required in an instance segmentation task. Therefore, it combines the two important aspects of computer vision task, object detection, which classifies and localizes the objects from an image and semantic segmentation, which classifies and assigns each pixel into a fixed set of category. Mask R-CNN introduced RoIAlign layer that maps the regions more precisely by fixing the location misalignment. The following Fig. 3 is a simple illustration of Mask R-CNN model.

4 Single-Stage Object Detectors

4.1 You Only Look Once (YOLO)

YOLO falls under single-stage object detection models and widely used for real-time object detection task and introduced by Redmon et al. [16]. It generates the bounding boxes and class predictions in single evaluation. It is widely known as unified network and very fast compared to Faster R-CNN and runs using single convolutional neural network. The CNN used in YOLO is based on GoogLeNet model originally and

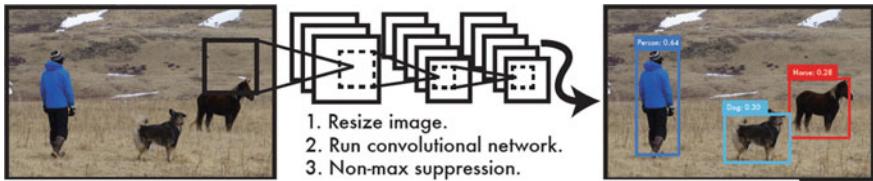


Fig. 4 YOLO: a single-stage object detector model. *Image source Redmon et al. [16]*

the updated version is called DarkNet based on VGG. As per shown in Fig. 4, it splits the input image into a grid of cells, where each cell directly classifies the object and predicts a bounding box. As a result, there are large numbers of bounding boxes generated that are integrated to a final prediction. The variations in YOLO are YOLOv1, YOLOv2, and YOLOv3, where YOLOv3 is the latest version. YOLO is a fast and good for real-time object detection tasks. It is possible to train it end-to-end for accuracy improvisation as it uses a single CNN for prediction. It is more generalized and performs well with generalization of natural and artwork images.

4.2 Single-Shot Detector (SSD)

Like YOLO, SSD also falls under single-stage object detection model and introduced by Liu et al. [17]. It takes only single shot to detect multiple objects within the image. Object localization and classification both performed in a single pass only. For extracting useful image features, SSD uses the VGG-16 model pre-trained on ImageNet dataset as its base model. At the end of the base model, it has additional convolutional layers for object detection. While predicting, score is generated for each object category presented in an image using default box. Also, to improve matching of object shape, it produces adjustments to the box. The network of SSD also pools the predictions generated from multiple feature maps with different resolutions. This process helps to handle objects of different sizes.

The following Table 1 presents the features and limitations of the benchmark object detection models including R-CNN family, YOLO and SSD.

5 Benchmark Datasets for Object Detection

As object detection models required huge amount of data to be trained, dataset plays very crucial role in success of these models [2]. A generalized datasets available for object detection tasks are ImageNet [18], MS COCO [19], PASCAL VOC [20], and open images [21]. These dataset are in annotated form and used for benchmarking deep learning algorithms. The following Table 2 summarizes the available images and classes to each dataset and types of images pertain to that dataset.

Table 1 Benchmark single-stage and two-stage object detectors

| Model | Features | Limitations |
|--------------|---|---|
| R-CNN | Performs classification and localization for object detection using selective search algorithm | Extracted 2000 region proposals per image, huge amount of time taken to train the network, Real-time implementation not possible as testing is very slow |
| Fast R-CNN | Region of interest (ROI) pooling layer that proposed fixed-sized regions, combines three models used in R-CNN, incorporated softmax in place of SVM for classification, more accurate and faster than R-CNN | Selective search algorithm for finding proposal makes it slow and time consuming |
| Faster R-CNN | Introduced region proposal network (RPN) in place of selective search, emerged as a very precise detection model for recognition, much faster and accurate than Fast R-CNN | Finding proposals takes time, many passes are required to complete the entire process |
| YOLO | Remarkably fast compare to R-CNN family of algorithms, single CNN for localization and classification, used for real-time object detection, better generalization for object representation | Struggles with detection of small objects comprises in a group, incorrect localizations are the main reason of error, problem with generalization of objects with uncommon aspect ratio |
| SSD | End-to-end training is possible, small CNN filters for prediction of category | Accurate than YOLO but somewhat slower, faster than Faster R-CNN but lesser in accuracy |
| Mask R-CNN | Used for instance segmentation, introduced RoIAlign layer for accurate segmentation | Classification depends on segmentation |

6 Applications of Object Detection Models

Several researchers have applied object detection models so far for different application areas including agriculture, medical imaging, satellite imagery, transport system, etc. The following Table 3 summarizes the work done with different object models and pre-trained model, application area, dataset used, and accuracy achieved. To evaluate the effectiveness of any object detection model, various parameters are taken into consideration including average precision (AP), average recall (AR), mean average precision (mAP), and intersection over union (IoU).

Table 2 Benchmark datasets for object detection

| Dataset | Number of images | Number of classes | Types of images |
|--|------------------|-------------------|---|
| ImageNet—WordNet hierarchy | 14,000,000 | 21,841 | Animal, plant, and activity |
| Microsoft COCO (common objects in context)—Large scale and used for object detection and segmentation | 330,000 | 80 | 250,000 people with key points, more than 200,000 are labeled, 5 captions per image |
| PASCAL VOC (Visual object classes)—Standardized image data sets obtained through VOC challenges | 11,530 | 20 | Common images of daily life, annotations are in XML file |
| Open Images—OICOD (open image challenge object detection)—training – 9 million + images, validation – 41 k + images, test – 125 k + images | 9,000,000 | 6000 | Diverse images of complex scenes, all images are annotated |

7 Conclusion

With the increase number of usage of face detection systems, video surveillance, vehicle tracking and autonomous vehicle driving, fast and accurate object detection systems are heavily required. Object detection refers to locating and classifying object from digital image. With the progressive result from deep CNN architectures, CNN-based object detectors are used in variety of applications. Based on the methodology, it has been categorized as either single-stage or two-stage object detection model. This paper summarizes the different CNN-based models that include R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, SSD, and YOLO. Apart from this, it explains the different features of available datasets. It also covers the details of research work carried out so far that applied object detection models in various fields of applications.

Table 3 Applications of object detection models

| Description of problem | Dataset | Object detection method | Object(s) identified | Accuracy obtained |
|---|--|----------------------------|---|--|
| The soldered dots identification of automobile door panels [22] | 800 images collected using camera | Fast R-CNN and YOLOv3 | Rectangle dot, semicircle dot, circle dot | Fast R-CNN: mAP-0.8270 Recall-0.8993 YOLOv3: mAP-0.7992 Recall-0.9900 |
| Different scales face detection [23] | FDDB, AFW, PASCAL faces, Wider face | Improved Faster R-CNN | Human faces from different scales | Recall-96.69% |
| Using semantic segmentation pedestrian detection [24] | Caltech pedestrian dataset with 10 h of video consists 2300 pedestrian | Faster R-CNN + VGG16 model | Pedestrian segmentation and detection | IoU-0.75 |
| Segmentation and detection of oral disease [25] | MSCOCO dataset for pre-training and 30 training and 10 validation | Modified Mask R-CNN | Cold sores and canker sores | AP- 0.744 |
| Gastric cancer diagnosis [26] | 1400 images with 1120 positive sample and 280 negative sample | Mask R-CNN | Prediction with masking and bounding box | AP-61.2 |
| Ear detection [27] | AWE dataset | Mask R-CNN | Colored region on identified ear | Accuracy-99.7% IoU-79.24% Precision-92.04% |
| Abandoned baggage detection [28] | User created dataset | YOLO | Box around the person and abandoned bag | 11 cases are correctly identified |
| Fast vehicle detection [29] | KITTI (15,000 images) + LSVH (16 videos) | Improved Fast R-CNN | Bounding box around the vehicles | AP Easy-89.20% Moderate-87.86% Hard-74.72% |
| Ship detection and segmentation [30] | 42,500 images from the Airbus ship dataset | Mask R-CNN | Circle box and masking around the ship | mAP-76.1% |
| Bones detection in the pelvic area [31] | 320 monochromatic images created out of two CT datasets | YOLO | Bones detection with labels | Precision-99% Recall-99% |

References

1. Z. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: a review. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(11), 3212–3232 (2019)
2. L. Liu, W. Ouyang, X. Wang et al., Deep learning for generic object detection: a survey. *Int. J. Comput. Vis.* **128**, 261–318 (2020)

3. A. Opelt, A. Pinz, M. Fussenegger, P. Auer, Generic object recognition with boosting. *IEEE TPAMI* **28**(3), 416–431 (2006)
4. A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: a brief review. *Comput. Intell. Neurosci.* 1–13 (2018)
5. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015)
6. H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection. *PAMI* (1998)
7. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
8. A.R. Pathak, M. Pandey, S. Rautaray, Application of deep learning for object detection. *Procedia Comput. Sci.* **132**, 1706–1717 (2018)
9. C. Li, Transfer learning with Mask R-CNN, https://medium.com/@c_61011/transfer-learning-with-mask-r-cnn-f50cbbea3d29
10. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587 (2014)
11. L. Weng, Object detection for dummies part 3: R-CNN family. <https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html>
12. R. Girshick, Fast R-CNN, in *ICCV*, pp. 1440–1448 (2015)
13. M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **88**, 303–338 (2010)
14. S. Ren, K. He, R. Girshick, J. Sun, Faster RCNN: towards real time object detection with region proposal networks. *IEEE TPAMI* **39**(6), 1137–1149 (2017)
15. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask RCNN, in *ICCV* (2017)
16. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real time object detection, in *CVPR*, pp. 779–788 (2016)
17. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. Berg, SSD: single shot multibox detector, in *ECCV*, pp. 21–37 (2016)
18. J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, ImageNet: a large scale hierarchical image database, in *CVPR*, pp. 248–255 (2009)
19. T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, L. Zitnick, Microsoft COCO: common objects in context, in *ECCV*, pp. 740–755 (2014)
20. M. Everingham, S. Eslami, L.V. Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: a retrospective. *IJCV* **111**(1), 98–136 (2015)
21. A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset et al., The open images dataset v4: unified image classification, object detection, and visual relationship detection at scale. [arXiv:1811.00982](https://arxiv.org/abs/1811.00982). (2018)
22. W. You, L. Chen, Z. Mo, Soldered dots detection of automobile door panels based on faster R-CNN model, in *Chinese Control And Decision Conference (CCDC)* (Nanchang, China, 2019), pp. 5314–5318
23. W. Wu, Y. Yin, X. Wang, D. Xu, Face detection with different scales based on faster R-CNN. *IEEE Trans. Cybern.* **49**(11), 4017–4028 (2019)
24. T. Liu, T. Stathaki, Faster R-CNN for robust pedestrian detection using semantic segmentation network. *Front. Neurorobot.* (2018)
25. R. Anantharaman, M. Velazquez, Y. Lee, Utilizing mask R-CNN for detection and segmentation of oral diseases, in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (Madrid, Spain, 2018), pp. 2197–2204
26. G. Cao, W. Song, Z. Zhao, Gastric cancer diagnosis with mask R-CNN, in *11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Hangzhou, China, 2019), pp. 60–63
27. M. Bizjak, P. Peer, Ž. Emeršič, Mask R-CNN for ear detection, in *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (Opatija, Croatia, 2019), pp. 1624–1628
28. T. Santad, P. Silapasupphakornwong, W. Choensawat, K. Sookhanaphibarn, Application of YOLO deep learning model for real time abandoned baggage detection, in *IEEE 7th Global Conference on Consumer Electronics (GCCE)* (Nara, 2018), pp. 157–158

29. H. Nguyen, Improving faster R-CNN framework for fast vehicle detection. *Math. Prob. Eng.* 1–11 (2019)
30. N. Xuan, D. Mengyang, D. Haoxuan, H. Bingliang, W. Edward, Attention mask R-CNN for ship detection and segmentation from remote sensing images. *IEEE Access* 1–1 (2020)
31. Z. Krawczyk, J. Starzyński, Bones detection in the pelvic area on the basis of YOLO neural network, in *19th International Conference* (2020)

Autonomous Vehicle Simulation Using Deep Reinforcement Learning



Rishikesh Kadam, Vishakha Vidhani, Bhavika Valecha, Anushree Bane, and Nupur Giri

Abstract The reinforcement learning algorithms have been proven to be extremely accurate in performing a variety of tasks. These algorithms have outperformed humans in traditional games. This paper proposes a reinforcement learning based approach to autonomous driving. The autonomous vehicles must be able to deal with all external situations to ensure safety and to avoid undesired circumstances such as collisions. Thus, we propose the use of deep deterministic policy gradient (DDPG) algorithm which is able to work in a complex and continuous domain. To avoid physical damage and reduce costs, we choose to use a simulator to test the proposed approach. The CARLA simulator would be used as the environment. To fit the DDPG algorithm to the CARLA environment, our network architecture consists of critic and actor networks. The performance would be evaluated based on rewards generated by the agent while driving in the simulated environment.

Keywords Autonomous driving · Imitation learning · Reinforcement learning · Deep deterministic policy gradient · Simulation · CARLA simulator · Self-driving agent · Artificial intelligence

R. Kadam (✉) · V. Vidhani · B. Valecha · A. Bane · N. Giri
Vivekanand Education Society's Institute of Technology, Mumbai, India
e-mail: 2016.rishikesh.kadam@ves.ac.in

V. Vidhani
e-mail: 2016.vishakha.vidhani@ves.ac.in

B. Valecha
e-mail: 2016.bhavika.valecha@ves.ac.in

A. Bane
e-mail: 2016.anushree.bane@ves.ac.in

N. Giri
e-mail: nupur.giri@ves.ac.in

1 Introduction

The NHTSA report in 2015 indicated that 94% of the total 2.3 million road accidents in the United States happened because of human error [1]. An autonomous driving agent would reduce this number as it will take safe actions quickly to avoid undesired circumstances and would not be subjected to issues faced by human drivers such as fatigue and loss of concentration. While developing such an agent, hard-coding all the possible scenarios is a huge task that requires constant monitoring and updates. That is when artificial intelligence and deep learning techniques come into the picture. Although the reinforcement learning algorithms have been proven to be better than humans at basic tasks such as playing simple games, their implementation in complex environments is not yet conclusively perfect [2]. We propose the use of Deep Deterministic Policy Algorithm (DDPG) which is expected to work better than simple Deep Q-Learning networks.

We have used simulators instead of developing an actual vehicle to train in a real environment. The use of simulators reduces the cost as well as time as we do not need to create a vehicle having all the sensors. The simulators provide life-like graphics and vehicle dynamics along with various sensor APIs which allow us to perform all the steps in autonomous agent development without any hardware.

Imitation learning is a type of supervised learning. It is useful to generate the same behavior when it is easy to demonstrate desired behavior by an expert rather than specifying a reward function. It is also known as learning by demonstrations [3]. As it is easy to collect the demonstrations of human vehicles which is then used by imitation learning to train the model which maps the input to actions [4]. It determines steering and acceleration based on camera images which are taken as input.

In this paper, Imitation learning is performed on an Udacity simulator where the model is trained such that the vehicle does not go off the track. The supervised learning approaches like imitation learning require huge datasets that need to be exhaustive in order to cover the required number of data points of each possible scenario [5]. We have also used Deep Deterministic Policy Gradient (DDPG) algorithm which is a combination of deterministic policy gradient, an actor-critic algorithm, and deep Q-learning, to train the driving agent in continuous action spaces on CARLA simulator. CARLA is an open-source simulator for autonomous driving research [6]. It provides different urban layouts with different weather conditions, multiple vehicle models, buildings, pedestrians and it also gives support for training, prototyping, and validation of autonomous driving models. The reward function is used to calculate the rewards based on the input taken which then helps to avoid the collisions.

2 Algorithms Used

2.1 Imitation Learning

Imitation Learning uses a convolutional neural network to train the self-driving car model. CNN learns from the behavior of a human expert driver; thus this type of learning is called Imitation Learning. Goal of imitation learning is to perform as well as an expert with an error that has linear dependency [7]. The model takes image data along with control parameters such as steering angle, speed, throttle, and gear as input and predicts the vehicle control values. Thus, the model is trained on labeled data and is a type of supervised learning. Imitation learning is the most basic way of developing a self-driving car. Although a considerable amount of time is required for generating the dataset as the dataset needs to be exhaustive and be inclusive of all the possible scenarios or environmental conditions [5]. The agent simply tries to replicate the behavior of the human expert driver. Hence, the agent won't be able to make correct decisions if a certain situation arises which is much different from the ones in the dataset on which it was trained.

2.2 Deep Deterministic Policy Gradient (DDPG)

DDPG algorithm concurrently learns a Q -function and a policy. It uses the Bellman equation to learn the Q -function, and uses the Q -function to learn the policy. Bellman equation [8] with the discount rate γ , describing the optimal action-value function, $Q^*(s, a)$ is given by,

$$Q^*(s_i a_i) = E[r_i + \gamma \max Q^*(s_{i+1}, a_{i+1})] = |s, a| \quad (1)$$

The structure of the DDPG consists of an actor-critic neural network [9]. An actor represents the policy structure whereas the critic represents the Q -value function. The current environment state is taken as an input of the neural network. The actor predicts an action for the given input and this action goes to the critic to produce the action-value function $Q(s, a)$. The critic's output is responsible for learning in actor and critic networks. The weights in the actor-network are updated using the update rule from the deterministic policy gradient theorem. The gradients obtained from the error are used to update the critic network.

3 Methodology

3.1 Imitation Learning Using Udacity Simulator

Imitation learning uses a convolutional neural network to train the self-driving car model. The Udacity simulator has two tracks. It provides left, center, and right images and parameters such as steering angle, speed, throttle, and brake. We drove the car at a constant speed and kept it at the center of the track in the “Recording Mode” of the simulator. One dataset for each track was generated having nearly 15,000 images each (5000 images per camera angle).

Figure 1 shows an image captured by center camera. The model is trained on processed images. The image processing steps included random shearing, cropping, and flipping of images. The images captured by the simulator come with a lot of details irrelevant in the model building process like trees, sky, etc. The hood of the car also occupies the bottom area in the captured images. The extra space occupied by these details requires additional processing power. Thus, in the cropping stage, we remove 35% of the original image from the top and 10% from the bottom. In the flipping stage, we randomly (with 0.5 probability) flip images. This operation was especially important for track 1 as it had less right bends as compared to left bends. It would have created an unbalanced dataset which would have resulted in the agent unable to predict right turns. Thus, we flipped half of the images and adjusted the corresponding steering angles. Then we resized images to 64×64 in order to reduce training time. A sample resized image is shown in Fig. 2. Figure 3 shows the architecture of the system. The processed images are then fed to the neural network.

NVIDIA’s End to End Learning for Self-Driving Cars paper formed the basis of our convolutional neural network model [10]. We used MaxPooling layers after each Convolutional Layer in order to cut down training time [11]. The model consists of five consecutive Convolution and MaxPooling layers followed by four fully connected layers. We then split the dataset training and validation set in ratio 80:20. We used the `fit_generator` API of the Keras library for training our model which incorporated Adam optimizer to optimize the learning rates. The trained model then

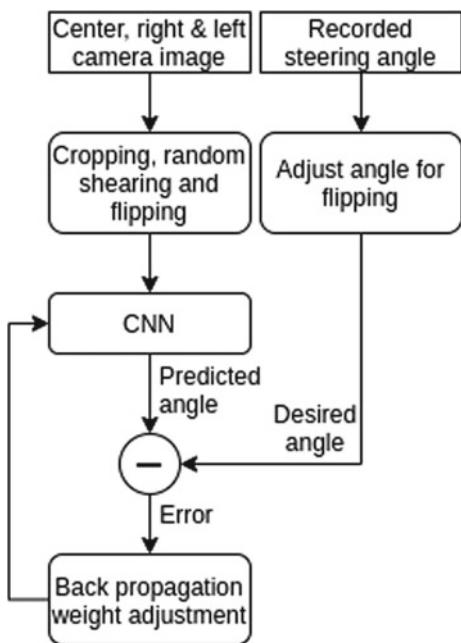


Fig. 1 Original image with steering angle 0.667015

Fig. 2 Processed image with steering angle -0.667015



Fig. 3 System architecture



predicted the steering angle. The speed of the vehicle was controlled from the code which was maintained around 22 kmph on track 1 and 27 kmph on track 2. The higher speed was maintained for track 2 as it had steep ascents compared to track 1 which had a plain road.

3.2 Reinforcement Learning Using CARLA Simulator

We used the DDPG algorithm to train the agent to control the vehicle in the CARLA simulator. The ‘state’ of the environment consists of the vehicle speed, throttle, steering angle, collision impact details, traffic light state, speed limit, percentage of vehicle body intersecting the lane, and RGB image captured by the camera. Figure 4 shows a sample RGB image. We used the `compute_feature()` function provided by CARLA which crops and resizes the image. The state is fed to the actor-network which predicts the action to be taken. The steering angle is in the range $[-1.0, 1.0]$ whereas the throttle and brake are in the range $[0, 1.0]$. Instead of making our actor-network predict three parameters (throttle, brake, and steering angle), we made it predict steering angle and effective acceleration, i.e., throttle minus brake. The range of this acceleration is $[-1.0, 1.0]$.

The first layer of the critic network takes the state as input. Its output and the actions predicted by the actor-network are fed to the second layer. The critic network predicts the Q -value which is to be maximized. We trained the agent with 4 different architectures of actor and critic networks as shown in Table 1. Figure 5 shows the structure of network N2. The ReLu activation function was used for all the layers.

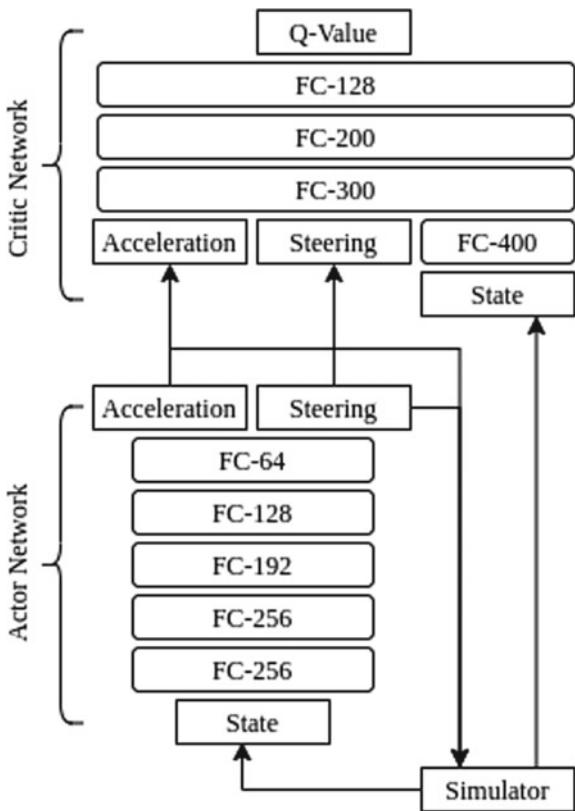


Fig. 4 Image captured by center camera in CARLA simulator

Table 1 Architecture of actor and critic network

| Network | Actor-network | | Critic network | |
|---------|-------------------------|------------------------------------|-------------------------|------------------------------------|
| | Number of hidden layers | Number of neurons in hidden layers | Number of hidden layers | Number of neurons in hidden layers |
| N1 | 4 | 256, 256, 128, 64 | 3 | 400, 300, 128 |
| N2 | 5 | 256, 256, 192, 128, 64 | 4 | 400, 300, 200, 128 |
| N3 | 5 | 256, 256, 192, 128, 64 | 5 | 400, 300, 200, 128, 64 |
| N4 | 3 | 256, 128, 64 | 3 | 400, 200, 128 |

Fig. 5 Structure of neural network N2



After training for a few episodes, the model tends to utilize the learnt actions again and again. This is known as exploitation which results in the agent missing out on other possible actions that would have produced better rewards. This is avoided by adding exploration noise to the predicted actions. To generate the exploration noise during training, the Ornstein–Uhlenbeck process was used [12]. The reward function utilizes the state of the environment to calculate the reward. For undesired conditions such as intersecting lanes, collision with other objects, moving with speed greater than the speed limit, negative reward is given. For maintaining speed below speed limit, positive reward is given. The reward is calculated for each action predicted by the actor-network. The total reward for an episode is the sum of rewards for all actions taken in that episode. An episode ends when the vehicle stands still when a collision occurs.

4 Results

4.1 Imitation Learning

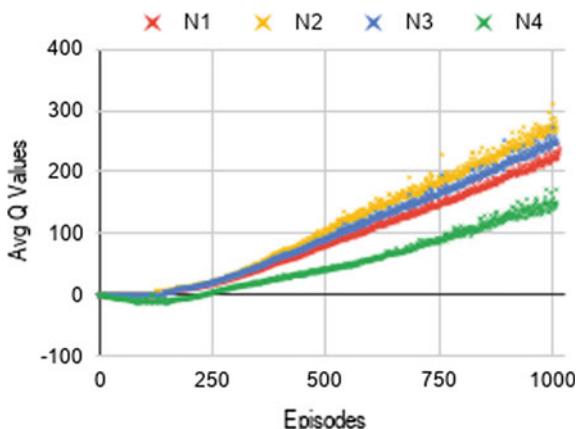
The Udacity simulator provides an autonomous mode where the vehicle control input predicted by the agent is sent via sockets to the simulator. We trained two models on datasets of separate tracks. Both models were able to control the vehicle without going off-road. As the models were trained on different datasets, the model trained on track 1 data was unable to drive properly on track 2 and vice versa. This is because the input images of the two tracks were completely different.

4.2 Reinforcement Learning

The multilayer networks were trained using NVIDIA GeForce 920M cards. First, all four networks were trained until 1000 episodes. Figure 6 shows the average Q values generated by these networks. Based on this performance, the network N2 was selected for further training. It was trained for more than 150,000 episodes. The learning rate for the actor-network was 0.0001 and that for the critic network was 0.001. The Adam optimizer was used for both the networks. After nearly 125,000 episodes, the agent was able to drive within the lane without colliding with other cars, pedestrians, and other objects.

There were few negatively rewarded episodes even after training for more than 150,000 episodes. These episodes are the ones where the vehicle was spawned too close to a red traffic signal thus not giving enough time for the agent to move forward and produce positive rewards. In all other episodes, the agent generated positive rewards.

Fig. 6 Comparative analysis of different neural network structures



5 Conclusion

Thus, we have used imitation learning and a deep deterministic policy gradient algorithm to train the autonomous driving agents. Imitation learning uses labeled data to predict the steering angle whereas the DDPG algorithm uses continuous action spaces ideal for autonomous driving scenarios. The actor-critic network in DDPG is used to predict the action according to the policy and the maximum Q -value for the action. We have trained our autonomous agent on Udacity simulator using imitation learning and on CARLA simulator using DDPG algorithm. Both the agents were able to control the vehicle in the simulated environments without going off-road or colliding with other objects.

6 Future Scope

We have used the traffic signal state and speed limit provided by the simulator API. In real-life scenarios, these need to be detected using the images captured by the cameras placed on the vehicle. This can be achieved by adding an object-detection module. But it might add unnecessary processing load. Another way of doing this is by implementing conditional affordance learning (CAL) [13]. The CAL algorithm focuses on the area of interest in the captured image to produce the output efficiently.

References

1. S. Singh, Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report, Traffic Safety Facts Crash Stats, National Highway Traffic Safety Administration (2015)
2. V. Mnith, K. Kavukcuoglu, D. Silver, A. Graves, L. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with deep reinforcement learning. Tech. Rep. (2013). <https://arxiv.org/abs/1312.5602>
3. H. Daume, A course in machine learning (2017). http://ciml.info/dl/v0_99/ciml-v0_99-ch18.pdf
4. F. Codevilla, M. Muller, A. Lopez, V. Koltun, A. Dosovitskiy, End-to-end driving via conditional imitation learning, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE Press, Brisbane, QLD, 2018), pp. 4693–4700
5. T. Osa, J. Pajarinen, G. Neumann, J.A. Bagnell, P. Abbeel, J. Peters, An algorithm perspective on imitation learning. Found. Trends Robot. 7(1–2), 1–179 (2018)
6. A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open driving simulator, in *1st Conference of Robot Learning (CoRL 2017)* (PMLR Press, Mountain View, USA, 2017), pp. 1–16
7. Y. Pan, C. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, B. Boots, Agile autonomous driving using end-to-end deep imitation learning. Tech. Rep. (2019). <http://arxiv.org/abs/1709.07174>
8. Open AI Spinning Up. <https://spinningup.openai.com/en/latest/algorithms/ddpg.html>. Accessed 10 Jan 2020

9. H. Yi, Deep deterministic policy gradient for autonomous vehicle driving, in *20th International Conference on Artificial Intelligence (ICAI'18)* (CSREA Press, Nevada, USA, 2018), pp. 191–194
10. M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zeiba, End to end learning for self driving cars. Tech. Rep. (2016). <http://arxiv.org/abs/1604.07316>
11. F. Chollet, *Deep Learning with Python* (Manning, New York, 2018)
12. E. Bibbona, G. Panfilo, P. Tavella, The Ornstein-Uhlenbeck process as a model of a low pass filtered white noise. *J. Metrol.* **45**, 117–126 (2008)
13. A. Sauer, N. Savinov, A. Geiger, Conditional affordance learning for driving in urban environments, in *2nd Conference on Robot Learning (CoRL 2018)* (PMLR Press, Zurich, Switzerland, 2018), pp. 237–252

Bitcoin Price Prediction Using Time Series Analysis and Machine Learning Techniques



Aman Gupta and Himanshu Nain

Abstract Cryptocurrencies are becoming a major moneymaker because of their high availability and abundance of easy investment platforms. In this paper, we have attempted to predict bitcoin value by taking into consideration various features that may affect its price. The amount of cryptocurrency in circulation, the volume of cryptocurrency exchanged in a day and the demand for cryptocurrency are a few of the factors that influence its cost. The forecasting is done using different time series analysis techniques like moving average, ARIMA and machine learning algorithms including SVM, linear regression, LSTM and GRU. Our goal is to compare all these models based on their observed accuracy. The dataset has been recorded daily over the course of three years.

Keywords Cryptocurrency · ARIMA · SVM · Linear regression · LSTM · GRU

1 Introduction

Over the years, the world has been progressing toward a cashless economy, with now only about 8% of the world's money being represented by physical notes. People have been shifting to cards and cheques due to the ease of money exchange, giving rise to cryptocurrency.

Cryptocurrency is a digital currency that allows strong safekeeping methods to secure financial transactions. Unlike the legal tender money declared by the government, cryptocurrency is not a fiat currency. Hence, it is not backed by any government, making it a decentralized system [1] and fairly independent of the failing economy of any particular country. This has led to a widespread adoption and use of cryptocurrency.

A. Gupta (✉) · H. Nain
Delhi Technological University, Delhi, India
e-mail: amangupta182@gmail.com

H. Nain
e-mail: himanshu10nain@gmail.com

Skepticism about cryptocurrency as a means of transaction has been mounting up lately due resource-intensive computations required to mine money and a lack of tether to reality. Despite this, the total amount of money invested in bitcoin as of March 4, 2020 is more than 160 billion dollars. It is estimated that more than 250 billion dollars are invested in all the cryptocurrencies around the world. For many people, cryptocurrency has been a major source of income. The buy and hold strategy and day trading are some of the really popular earning methods. These techniques are very similar to how people make profits in the stock market.

These vast money making opportunities gave rise to the development of stock market prediction using machine learning algorithms [2] (classification and regression) and neural networks [3]. This allowed people to have a mathematical surety to their investments. As people shifted from the stock market to cryptocurrency, the trend of prediction followed and gave rise to a huge industry of forecasting.

2 Related works

One of the earliest financial predictions was done by Ou and Penman [4], using the market price to determine a firms' value and analyzing financial statements to predict future returns on the stock. In 2001, Chen et al. [5] attempted to use a probabilistic neural network [6] to find if the future price of the Taiwan stock index would be higher or lower than the current value. This paper proved that the use of PNN wielded results much better than any methods in use until that point.

In 2004, Huang et al. [7] applied the support vector machine (SVM) algorithm to predict the movement of the stock market. The performance was compared with backpropagation neural networks & discriminant analysis and SVM outperformed all other classification methods used in the study. SVM continued to be one of the most popular techniques in use due to the high accuracy. The notion of “the survival of the fittest” was introduced by [8] as a prediction model when genetic algorithms were combined with SVM, creating a GA-SVM model that outperformed a standard SVM model.

As neural networks became more technologically advanced and easier to implement, McNally et al. [9] used LSTM [10] and RNN [11] algorithms to predict the future direction of bitcoin, whether the price will go up or down. As expected, LSTM performed the best. As a comparison, the time series ARIMA model was also implemented but the results were unsatisfactory. Development on this was done by Rane and Dhage [12] using RNN and combining it with Twitter sentiment analysis to see the influence of tweets on the value of bitcoin. With an accuracy of 77.62%, it can be categorized as a successful model. Further, refinement came from the creation of a new framework called C2P2 in [13] with a similar goal of predicting whether the cryptocurrency value would rise or fall. However, the C2P2 engine differed from the standard methods as it used feature extraction and similarity mapping and was hence an improvement on its predecessors. A comparison between long short-term memory (LSTM), deep neural network (DNN), convolutional neural network (CNN) [14] and

their combination was performed on bitcoin, where LSTM was used to predict the cost of bitcoin (regression) and DNN-based model to classify whether the price will go up and down. A systematic review was done in [12] on different machine learning techniques like ARIMA [15], regression, LSTM, SVM, BGML [16], etc. and found that Nonlinear Autoregressive with Exogenous Input Model (NARX) [17] gave the best results in predicting bitcoin price with an accuracy of 62%.

3 Methodology

Cryptocurrencies tend to follow a trend that can be studied and analyzed to predict future value. Many of the factors on which bitcoin value depends are unquantifiable and can make prediction difficult. So, an approach that recognizes and understands the underlying pattern in past data and makes predictions based on them is easier to implement. In such cases, time series analysis is used. In order to study the dependencies of cryptocurrency, we include some of the other attributes. These features, like the volume of cryptocurrency, exchanged on a particular day can prove to be a good factor to train different machine learning models. The next category of models that we looked at was deep learning. The algorithms included in deep learning can handle huge amounts of data and are capable of processing at a much faster rate. This allows deep learning methods to find correlations across multiple data points and make predictions of future values based on previous values. In this paper, we have studied and implemented a few algorithms from each model. The details of these methods and algorithms are as follows.

3.1 Time series analysis (TSA)

Time series analysis is the use of statistical techniques to deal with the time series data. A time series involves a set of data that is spread over time. This data generally consists of discrete data points that are equidistant from each other. The forecasting methods in TSA allow prediction of future data based only on the past observed data. Time series analysis comprises four components. They are seasonal variations, trend variations, cyclical variations and random variations. For cryptocurrencies, the analysis of trend variation is the most important part. Seasonal variations and cyclical variations rarely occur in cryptocurrency data. The TSA methods used are mentioned below.

- **Naïve approach**

Since the change in value across consecutive days is considerably small, this method assumes that the data of i th day is equal to the data of $i - 1$ th day.

- **Simple average**

The most basic technique of TSA is the simple average method. As the name suggests, this value is determined by calculating the average of data collected. Mathematically,

Where $i = 1, \dots, n$ and C is the value of past data.

A major drawback in simple average is that the value of n th day is considered to depend on the value of all the days.

$$\frac{\sum_{i=1}^n C_i}{n}$$

- **Moving average**

Moving average deals with the limitations of the simple average method by making a small change in the approach. Instead of taking average of all the data, a short constant span of time is selected and the average of only that span is taken. Assuming this span to be equal to 5 days, we can now say that the value of 100th day no longer depends on the value of first 94 days. This allows the old data to be forgotten with time.

- **Weighted average**

Moving average makes the assumption that all the days within a span contribute equally in predicting the price. A weighted average involves assigning different weights to the different values so that the most recent date is more involved in prediction as compared to the least recent date. Mathematically,

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

- **ARIMA**

The term autoregressive integrated moving average (ARIMA) is composed of two parts. The “autoregressive” part deals with prediction of future values by applying regression on its own past (lagged) values. The “moving average” deals with the combination of error terms encountered during regression. The integration of these two separate methods allows better understanding and prediction capability. The calculation of difference between consecutive values is done as follows:

$$y'_t = y_t - y_{t-1}$$

3.2 Machine learning

Machine learning has been used in the prediction industry for decades now. The use of mathematical computations to understand the structure behind the observed data and use of a model to forecast allows better understanding and predicting abilities. The finance sector has tried to stay ahead in terms of technology and thus used

all the methods at its disposal to get better and more accurate results. Determining creditworthiness and fraud detection are the most basic applications of machine learning. Eventually, machine learning was applied to the stock market and gave rise to algorithmic trading. In this study, machine learning methods are applied on cryptocurrency as follows.

- **Univariate linear regression**

A univariate linear regression is a linear approach to modeling the relationship between a single independent variable (volume of the cryptocurrency (x)) and a dependent variable (cost of the cryptocurrency (y)).

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

Mathematically,

$$\begin{aligned}\hat{\alpha} &= \bar{y} - \hat{\beta} \bar{x}, \\ \hat{\beta} &= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \\ &= \frac{s_{x,y}}{s_x^2} \\ &= r_{xy} \frac{s_y}{s_x}.\end{aligned}$$

- **Polynomial linear regression**

A relationship between the independent and the dependent variable can sometimes be modeled as a polynomial of n th degree. Hence, an approach beyond a linear relationship is provided by a Polynomial regressor. This can be represented as:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n$$

- **Support vector regression regression**

A support vector machine is generally used for classification problem by separating two entities with a hyperplane. A SVR is based on the same concept and has proven to be useful in real world estimation problems.

3.3 Deep learning

As the emphasis on machine learning grew, the amount of data to be studied increased. This led to the development of complex mechanisms called artificial neural networks (ANNs), an idea inspired by the information processing nodes found in any biological systems. Deep learning is the use of ANNs, generally on unstructured or unlabeled data, to determine the underlying pattern in the dataset. Complex mathematical formulae are used to form correlation between data points and make predictions.

The methods used are

- **Long short-term memory**

RNN is a category of artificial neural network that is used to deal with temporal data. Each node of an RNN is used to form a long sequence and pass on information. LSTMs are a kind of RNN that were made to deal with long-term dependencies. They are able to remember and connect patterns across huge datasets and are hence widely known. The main concept behind an LSTM is the cell state which acts as a conveyor belt, running down the cells and hence allowing unchanged information flow. LSTMs also contain gates, a way to add or remove stored information.

- **Gated recurrent unit**

GRU is a variation of the LSTM model with only two gates (reset gate and update gate). This makes it easier for the model to remember information for a long time and facilitates easier updation and deletion of data. GRUs use hidden state to pass information across cells and hence do not have output gate.

4 Data collection and refining

The data for this work has been collected from coinmarketcap.com. The dataset had the columns data, open, high, low, close, volume. For the time series analysis, only the close value was used for prediction. The amount of cryptocurrency bought and sold in a particular day was chosen as the independent variable for regression. For the LSTM and GRU model, all of the columns were used to train the model. The dataset consists of 1363 days of data starting from 11-Nov-13 to 4-Aug-17.

5 Implementation and results

The prediction of any mathematical data cannot be precise. There is bound to be some amount of difference in the actual data and the predicted data. This difference however needs to be within an acceptable limit for the prediction to be termed as successful. Taking this margin of error (MOE) as 5% of the actual value makes it easy to calculate the extent to which an algorithm performs correctly. For every algorithm, the result of application is as follow:

- **Simple average**

The average of first $i - 1$ values was taken as the prediction of i th value. Out of 1360 values, only 159 were under the margin of error. This model had the worst result with accuracy of only 11.69%.

- **Moving average**

For the moving average method, a window of 3 was taken, i.e., value of i th value is calculated as the average $i - 1$, $i - 2$ and $i - 3$ values. With 1098 cases within MOE, the model performed unusually well with an accuracy of 80.73%.

- **Weighted average**

In this method, the weights were given as: $i - 1$ value was given 0.7, $i - 2$ was given 0.2 and $i - 3$ was given 0.1. With an improvement over the moving average, an accuracy of 86.32% was observed.

- **Naïve approach**

The value of i th day was said to be equal to the value of $i - 1$ th day. Only a slight improvement over the previous method was seen, with an accuracy of 86.78%.

- **Linear regression**

The independent variable (volume) and the dependent variable (close) were used to create a linear model, taking only data of past three days to train the model so that past data does not interfere with future predictions. An accuracy of 77.72% was observed, ensuring the fact that the value of cryptocurrency does linearly depend to a certain extent on currency exchanged during that day.

- **Polynomial regression**

The independent variable was converted to higher degree terms with powers 2, 3 and 4 yielding an accuracy of 55.66%, 55.73% and 55.14%. Use of any power higher than this resulted in a sharp fall of accuracy.

- **Support vector regression**

The data was scaled using MinMaxScaler available in the sklearn library of Python. Linear and rbf kernels were used to get an accuracy of 63.75% for both kernels.

- **LSTM**

Out of the available data, 1166 values were taken as training data and 130 were taken as testing data. The data was scaled using MinMaxScaler. Out of these 130, 44 were within MOE giving an accuracy of 33.84%.

- **GRU**

Since GRUs are better at recognizing patterns, no scaling was used. With an accuracy of 78.46%, the model performed really well (Figs. 1 and 2).

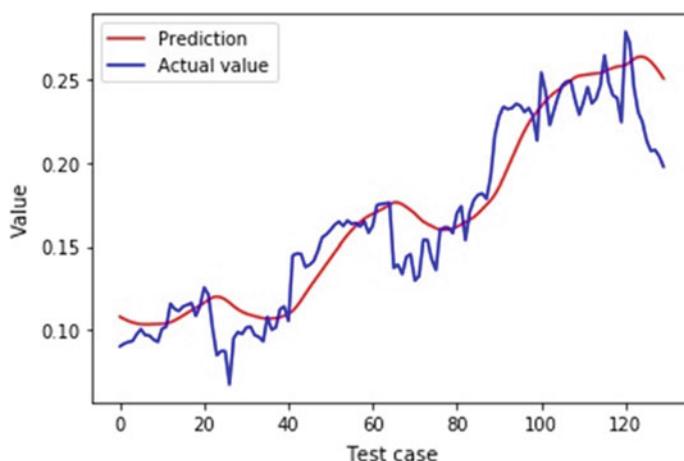


Fig. 1 LSTM

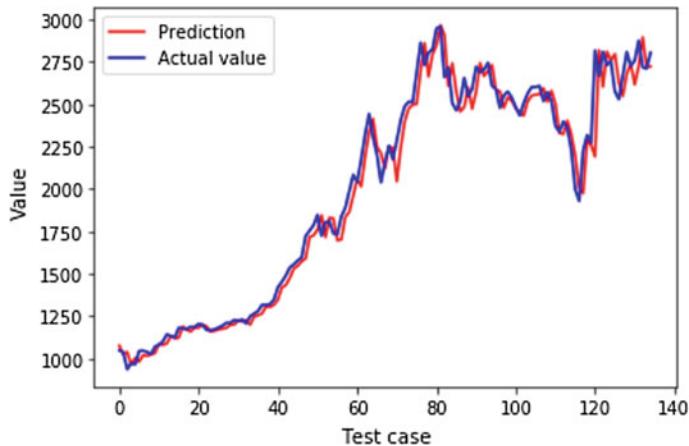


Fig. 2 GRU

6 Conclusion

Prediction has been in use for several millennia's. People have wanted to correctly predict the outcome of any future event, giving rise to an exhaustive study into forecasting. Over time, many different prediction concepts were developed and obviously some work better than the others, depending upon the domain in which they are applied. It has been noted on many occasions that SVM works better than most other classification methods for stock prediction (Fig. 3).

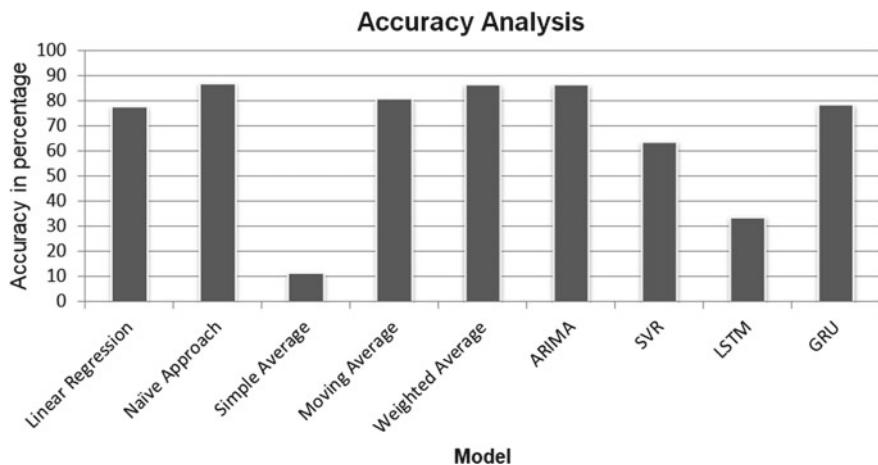


Fig. 3 Accuracy

The major contribution of this paper is the application of the different forecasting algorithms on bitcoin to predict its value on the next day with a fairly high accuracy. The paper also compares the aforementioned algorithms to find that the basic strategies like moving average outperformed complex LSTM architecture, suggesting that sometimes less is more.

Overall, it was seen that time series analysis, specifically weighted average model and the ARIMA model are better suited than machine learning or deep learning methods for bitcoin prediction. While the Naive approach did yield a slightly better accuracy of 86.78%, the results may change in case of a highly fluid market. Furthermore, a high accuracy obtained from the linear regression supports our assumption that bitcoin value depends on the amount of the cryptocurrency exchanged.

7 Challenges and future direction

Despite the various advancements in forecasting methods and their application in a number of fields, the work done for cryptocurrency prediction is far from complete. There exist a number of factors affecting price of a cryptocurrency that have either not been studied or discovered yet. As analyzed by Kristoufek [18], there are several drivers of bitcoin value, including the interest of people in a specific cryptocurrency, a factor that is difficult to quantize. The paper shows a direct relationship between bitcoin value and the Chinese market, explaining the events in China that caused sudden rise and fall in prices. An example to support this claim states that the acceptance of bitcoins by Baidu, a Chinese multinational technology company, in 2013, led to a sudden increase in bitcoin value, and subsequently, a drop in bitcoin value was observed when the Chinese government banned electronic purchases using bitcoin. The study suggests that a correlation between the value of American Dollar (USD) and the bitcoin also exists. Although this research gives a huge insight regarding the various drivers of bitcoin, it also establishes the fact an exhaustive study is needed to find out about the other determinants, since our current knowledge is lacking.

The work on forecasting will continue for a long time but the implementation of these predictions has not been done yet. Though the research studies allow people to gain an in-depth knowledge about the subject, they do not offer any practical use in the real world. The creation of a method to decide the correct time to buy or sell cryptocurrency will make it easy for the general people to make investments.

References

1. A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction* (Princeton University Press, USA, 2016)
2. S. Shen, H. Jiang, T. Zhang, Stock market forecasting using machine learning algorithms (2012)

3. T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka, Stock market prediction system with modular neural networks, in *1990 IJCNN International Joint Conference on Neural Networks*, vol. 1 (1990), pp. 1–6
4. J.A. Ou, S.H. Penman, Financial statement analysis and the prediction of stock returns. *J. Acc. Econ.* **11**(4), 295–329 (1989). Available at: <http://www.sciencedirect.com/science/article/pii/0165410189900177>
5. A.-S. Chen, M. Leung, H. Daouk, Application of neural networks to an emerging financial market: forecasting and trading the Taiwan stock index. *Comput. Oper. Res.* **30**, 901–923 (2001)
6. D.F. Specht, Probabilistic neural networks. *Neural Netw.* **3**(1), 109–118 (1990). Available at: <http://www.sciencedirect.com/science/article/pii/089360809090049Q>
7. W. Huang, Y. Nakamori, S.-Y. Wang, Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **32**(10), 2513–2522 (2005). Available at: <http://www.sciencedirect.com/science/article/pii/S0305054804000681>
8. R. Choudhry, K. Garg, A hybrid machine learning system for stock market forecasting. *World Acad. Sci. Eng. Technol.* **39** (2008)
9. S. McNally, J. Roche, S. Caton, Predicting the price of bitcoin using machine learning, in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (2018), pp. 339–343
10. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). Available at: <https://doi.org/10.1162/neco.1997.9.8.1735>
11. Z.C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning (2015)
12. P.V. Rane, S.N. Dhage, Systematic erudition of bitcoin price prediction using machine learning techniques, in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)* (2019), pp. 594–598
13. C. Bai, T. White, L. Xiao, V.S. Subrahmanian, Z. Zhou, C2p2: a collective cryptocurrency up/down price prediction engine, in *IEEE International Conference on Blockchain (Blockchain)* (2019), pp. 425–430
14. Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, Handwritten digit recognition with a back-propagation network, in *Advances in Neural Information Processing Systems* (1990), pp. 396–404
15. J. Contreras, R. Espinola, F.J. Nogales, A.J. Conejo, Arima models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **18**(3), 1014–1020 (2003)
16. M. Shirazi, D. Lord, S.S. Dhavala, S.R. Geedipally, A semiparametric negative binomial generalized linear model for modeling over-dispersed count data with a heavy tail: characteristics and applications to crash data. *Accid. Anal. Prev.* **91**, 10–18 (2016). Available at: <http://www.sciencedirect.com/science/article/pii/S0001457516300537>
17. T. Lin, B.G. Horne, P. Tino, C.L. Giles, Learning long-term dependencies in NARX recurrent neural networks. *IEEE Trans. Neural Netw.* **7**(6), 1329–1338 (1996)
18. L. Kristoufek, Ladislav kristoufek—what are the main drivers of the bitcoin price (2015)

The Detection of Diabetic Retinopathy in Human Eyes Using Convolution Neural Network (CNN)



Saloni Dhuru and Avinash Shrivas

Abstract A medical situation in diabetic patients is recognised as diabetic retinopathy, primarily involving human eye. Provenance of diabetic retinopathy is because of high blood glucose levels in past prolonged course of era called as diabetes mellitus. Diabetic retinopathy dataset has 5 levels of images present with levels numbered from 0 to 5 with initial level having mild signs of retinopathy to the last levels having no retinopathy. In high-resolution pictures of retina, system should segregate the pictures whether the patient has no diabetic retinopathy or has diabetic retinopathy. Originally, the pictures should be preprocessed by rotations and also need to be resized to standard image size so that the system can process the images with same efficiency. Then, deep learning approach of convolutional neural network (CNN) is applied to convert image which tells whether the patient is having diabetic retinopathy or not. The conclusions are alleged to conclude a sensitivity of 95% and a competence of 75%. So, this system can easily analyze retinal images whether healthy patients and diabetic patients diminishing the quantity of surveys of specialists.

Keywords Diabetic retinopathy · Greater pixels dataset images · Deep learning · Convolutional neural network (CNN)

1 Introduction

Diabetic retinopathy is an eye disorder mainly seen in diabetic patient's eyes. If patient has diabetes for a stretch of years, then his/her likelihood of establishing diabetic retinopathy. Person having diabetes for 20 years or more is said to have 85% chances of being affected by diabetic retinopathy. It causes spillage of veins

S. Dhuru (✉) · A. Shrivas

Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India

e-mail: salonidhuru26@gmail.com

A. Shrivas

e-mail: avinash.shrivas@vit.edu.in

and expanding, limiting blood from fleeting through and also constantly surge of irregular new blood vessels in the retina. Pigmentation or dull strands happen to make eyesight, fluctuating eyesight, hindered shading eyesight, obscured vision, dim or void regions in view and eyesight misfortune are few of typical side effects of the disease [1]. Regular clue and signals of the same are small-scale aneurysms, veins spillage, expanding of retina, irregular new blood vessels, and harmed nerve tendon. Diabetic retinopathy is efficient of evaluation with techniques, for example, focal laser therapy, scatter laser therapy, and vitrectomy. Medical procedure frequently deteriorates or disallows the advancement of diabetic retinopathy, yet it is anything but a total fix. It is an important factor, which can lead to retinal harm and eyesight depletion [2]. Key side effect of DR is exudates, which is shown in the retinal scan of the eyes and is a cause of generating DR within the patient.

Thus, a relevant location of this disorder is a compelling. Diagnosis strategies are fluorescein angiography and optical cognizance tomography that consist of outside liquid to cover on to the patient's eye afterwards the image is gathered. In any case, this framework can naturally and promptly foresee diabetic retinopathy with no outside operator, and that is a progressively advantageous technique both for specialists and patients [3].

In today's world, scanned images of the retina of the eye are playing an important role to carefully diagnose the problems associated with it. To differentiate the features such as blood vessels, fluid drip, exudates, haemorrhages, and micro-aneurysms within distinct grades, deep learning models are capable of measuring them efficiently. The system will tell whether the subject has diabetic retinopathy or not.

To address the above-stated problems, there is a need to develop a model that aims on introducing diabetic retinopathy diagnosis that naturally determines features which are crucial in diagnosing the stage of the disease without certain or manual feature extraction. For this, CNN is being used because of the fact that neural networks acquire the approach of biological brain. These processes are being pursued by image processing and also clustering for boosting the efficiency of next process steps. Preprocessing of image commenced by rotations and resizing the images to carry on the further analysis. To distinguish the images according to the listed features, the image classification steps work the best, thus getting the desired results. The following work should involve minimization of noise for obtaining an enhanced image and accurate data.

2 Literature Review

Many systems have been proposed earlier the revelation of diabetic retinopathy using machine learning algorithms, feature extraction, ensemble learning, neural networks, number of micro-aneurysm, digital image processing, edge detection, etc., are being studied and implemented.

- **Enrique V. Carrera** has proposed a computerized analysis established on the digitally altering process of retinal figure considering to encourage people for identifying diabetic retinopathy in before it propels. Their ambition is classifying naturally grade of non-proliferative diabetic retinopathy. Primarily original image before processing phase, they isolate blood vessel, micro-aneurysms, and hard exudates in an ordered way to select attributes which can be worn by SVM to consider the retinopathy grade of every single retinal figure. The technique was tried on almost 400 figures of retinal which are labelled according to scale of grades non-proliferative diabetic retinopathy. Final outcome obtained is 95% sensitivity and 94% predictive gap [4].
- **Mohamed Chetoui** proposed favour of distinct texture attributes mode for diabetic retinopathy, principally local ternary pattern (LTP) and local energy-based shape histogram (LESH). It does exhibit SVM for the distribution of refined histogram which is studied for binning scheme for representing attributes. The primary outcome provides that LESH is performing better with accuracy of 0.904. Analysing ROC curve that exhibits LESH including SVM-RBF gives finest area under curve giving performance of 0.931 [5].
- **Asti Herliana** proposed research which was conducted by implementing particle swarm optimization (PSO) mode to prefer the finest diabetic retinopathy attribute establishing on diabetic retinopathy data. Next, selected aspect is again derived using any classification mode considering NN (neural network). Outcome of study exhibits increase in outcome by applying NN-based PSO of 76.11%. The outcome also exhibits building up in outcome by emphasizing 4.35% of selection phase as to old outcome of 71.76% [6].
- **Shailesh Kumar** proposed two attributes, i.e., number and area of micro-aneurysm. Techniques exhibit initially as extraction of green channel, equalization of histogram and method of morphological is exhibited. It exhibits morphological process, PCA, contrast limited adaptive histogram equalization (CLAHE), and averaging filter phase to expose micro-aneurysm. Sensitivity is 96%, and specificity is 92% to DR identifying system [7].
- **Kranthi Kumar Palavalasa** initially exposed the achievable candidate exudate lesions by using the background subtraction approach. In completion stage of algo, they cut out the false exudate lesion detections exhibiting de-correlation stretch-based method. Test was performed public accessibly DiaretDB database, which encloses reality of images. Finest outcome of sensitivity is 0.87 and F-Score of 0.78 and Positive Predict Value of 0.76 [8].

3 Problem Statement

Diabetes arises when our body is not being able to produce acceptable insulin; therefore, it leads to high glucose level which causes damage to various body organs like heart and brain and also leads to slower healing of any wounds. Also, diabetes can affect the eye and be seen in the blood vessels of retina, which may cause blindness,

and this process is known as diabetic retinopathy. Diabetic retinopathy is a condition which can be seen in human eyes who are suffering from diabetes over a long period of time. Often, the eye gets affected after 20 years of prolonged diabetes. Diabetic retinopathy can be classified as NPDR (known as non-proliferative diabetic retinopathy) and PDR (known as proliferative diabetic retinopathy). NPDR (known as non-proliferative diabetic retinopathy) can be again further broken down into mild, moderate, and severe diabetic retinopathy.

Objectives:

In this system, the user can give input to the system in the form of scanned retinal images and the system tells us whether the patient has diabetic retinopathy or not. This helps the patients to know what is the condition of their vision and also gets to know the exact remedy to be taken.

- provides unique approach to concealed patterns in the data.
- helps avoid human biasness.
- implements neural network that classifies the disease as per the input of the user.
- reduces the cost of medical tests.

4 Proposed System

The neural network system uses the training data to train models to observe patterns and uses the test data which evaluate the predictive nature of the trained model. The data are split into two ways—one for training and other for testing sets which is an important part of assessing neural network models. When classifying a dataset in two parts, 80% of the data is used for training purpose, and 20% data is used for testing. The known output can be used as training set, and the model begins to learn on this dataset in order to induce to other data afterwards. The model is tested by predictions against the testing set after the model has been processed by using the training set. It is easy to determine whether the model that will be guessed are correct or not because the testing set data already contains familiar values for the attribute that needs to be predicted.

In most of the studies systems lack the dataset preprocessing steps which have a potency to give faulty outputs. For studying purpose, preprocessing of images is done through fewer rotations of images and also resizing them which enlarges the features of the image. In addition, preprocessing done is of resizing the image to a pixel of 200×200 which makes the images to a standardized size which makes it easier for the system to process the images with same efficiency.

a. The Dataset:

The dataset used is the diabetic retinopathy fundus images [9] of retina having pixels over varied resolutions. The dataset is downloaded from Kaggle website, which is openly available in it, having around 32,000 images for training set and 18,000 images for testing set. The inconsistency in the website's data lies in the

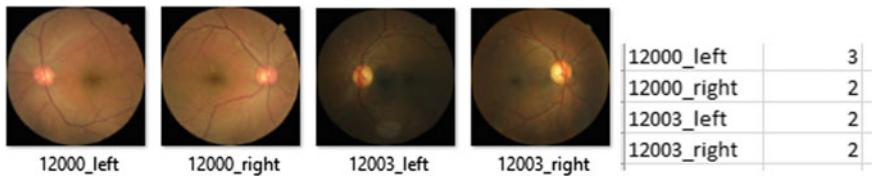


Fig. 1 Samples of dataset

image resolution that can vary for distinct figures, and the figures enclose noise; therefore, proper filtering is necessary to get proper information from the dataset. For this study, all the images have been chosen at 6:2 training and testing ratio. The dataset has labels assigned to them as per the severity of the retinopathy. Figure 1 shows the images which are present in the dataset. As it can be seen in the samples of dataset, all images are taken of different people, using distinct cameras, and of distinct sizes. Pertaining to the preprocessing section, these data are outrageously disrupted and wish for numerous preprocessing steps in order to train the model and to get all figures to a processable form.

b. Data preprocessing:

- Rotate and Resize All Images: All images were scaled to 256×256 . Despite taking longer to train, the detail present in images of this size is much greater than at 128 by 128. Moreover, 403 images were deselected from the training set. Scikit-Image prompts multiple warnings during resizing, due to these images having no colour space. Because of these problems, any images that were completely black were removed from the training data.
 - Rotate and Mirror Images: All images were rotated and mirrored including retinopathy images; no retinopathy images and also middle-staged images were mirrored and rotated at 90° , 120° , 180° , and 270° .
- In Fig. 2, the first images show two pairs of eyes, along with the black borders. Notice in the resizing and rotations how the majority of noise is removed.

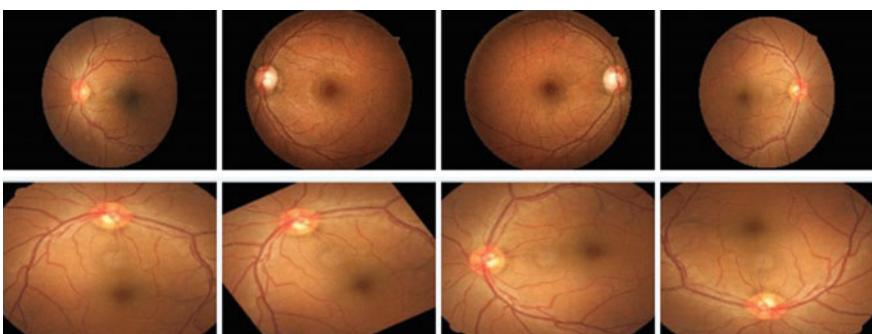


Fig. 2 Sample of data preprocessing

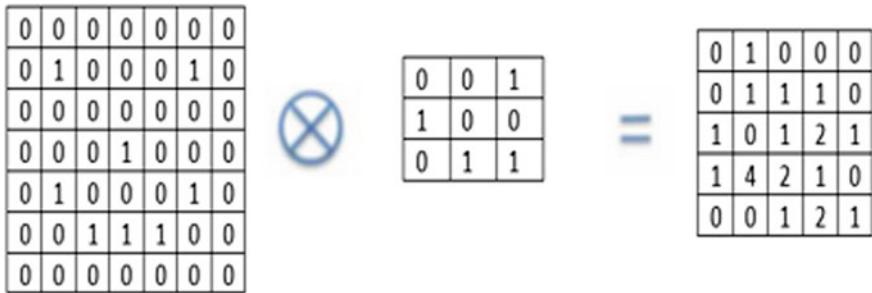


Fig. 3 Convolutional layer example

For rectification of class imbalance, a few rotations and mirroring are done with a few more images having retinopathy. In total, there are 106,385 images being processed by the neural network.

- The Convolutional Neural Network:** The network comprises of an input layer which accepts preprocessed images of resolution 200×200 pixels as an input. Then, there is combination of 3 sets of convolution; each set consists of a convolution layer, a rectified linear unit (ReLU) layer, and a max-pooling layer. Then, flattening or unrolling is performed on the last set of feature maps (pertaining to a single image) obtained after the 3 set convolution into a single feature vector in the flattening layer. This single feature vector is then given into an artificial neural network forming the dense layer of the convolutional neural network.
 - The Convolutional Layer:** This is used to select features from an input image. The relationship between the pixels is preserved by convolution function by learning the image features using matrix [12]. In the convolutional layer, each in all of the 3 sets has feature detectors of dimensions, 3×3 (Fig. 3).
 - ReLU Layer or Rectification Layer:** ReLU stands for rectified linear unit for performing a nonlinear operation. The purpose of ReLU is to propose nonlinearity in the convolution's result. Since the data would want the convolution's result to learn non-negative linear values, in the matrix all the negative numbers would be brought to 0.
 - Max-Pooling Layer:** It is used to decrease the number of parameters where the image is too large but it retains the important features of the map. The maps are obtained in all the feature of the ReLU layer; to preserve the features of spatial independence, max-pooling is done where the pooling stride is of dimensions (2×2), for making the convolutional neural network (Fig. 4).
 - Flattening Layer:** Flattening transforms a 2D matrix of features into a vector that can be fed into a fully connected neural network classifier [10]. In this layer, the end feature map achieved after 3 sets is flattened into a unique feature vector by taking an example in Fig. 5.
 - Dense Layer:** The structure of the neural network was decisive later studying the literature of another image processing tasks. An increase in convolution

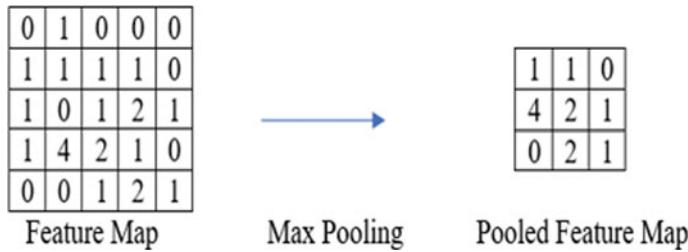


Fig. 4 Max-pooling layer taking a pooling stride of 2×2 dimension

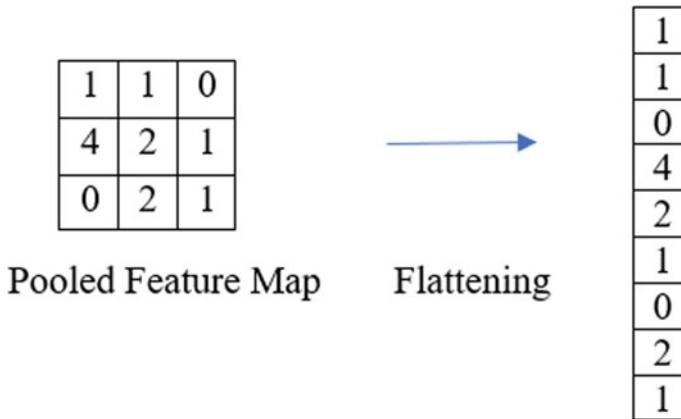


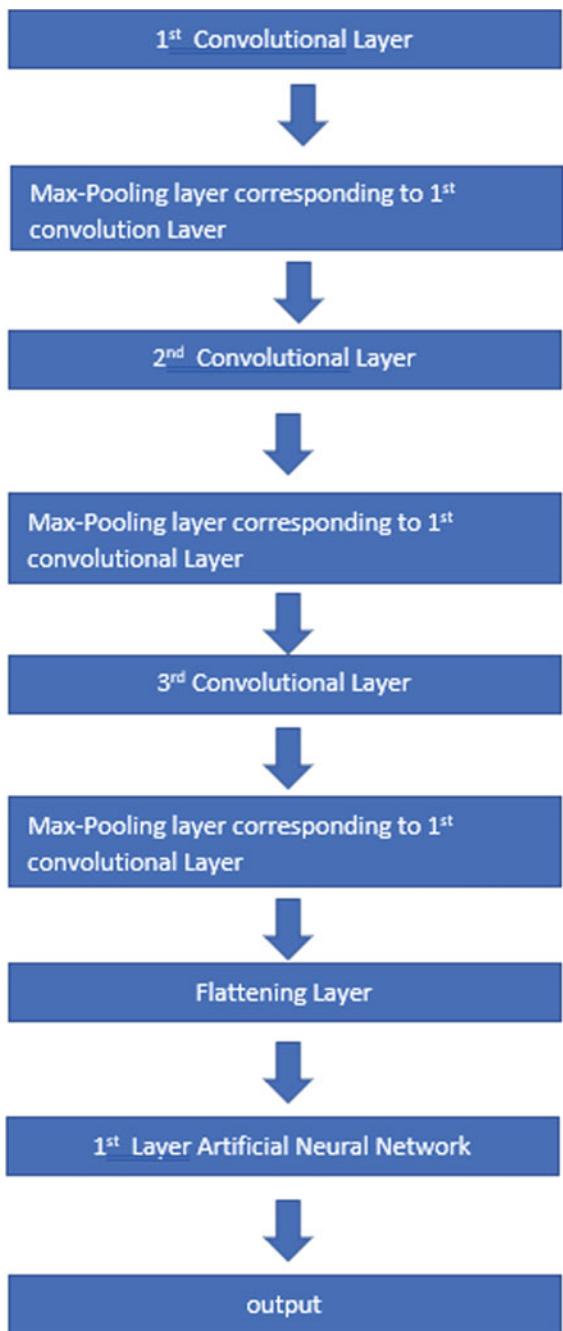
Fig. 5 Flattening layer example

layers perceives to let the network learn the broader features. For an instance, the first convolutional layer learns the edges and the final convolutional layer should determine the classification of diabetic retinopathy such as hard exudate which mainly differentiates the two types of diabetic retinopathy. The system initiates with convolution blocks with activation and then after each corresponding convolution layer. All max-pooling is achieved with kernel size 3×3 and 2×2 strides. After the last convolutional block, network is smoothed to 1D. Then, dropout is performed on dense layers (Fig. 6).

5 Conclusion

The proposed paper is on artificial neural network approach to diabetic retinopathy with the use of typical convolutional neural network (CNN) architecture. Alterations in diabetic retinopathy dataset images are fundamental to get fitting lineaments. Also resizing these images makes the system to efficiently and uniformly process

Fig. 6 CNN architecture summary



the images. Statistical values predict the severity level, but when the data is noisy or inconsistent leading to the chances of having poor dataset, it will let down the accuracy and would give incorrect results. Accuracy will be obtained with the best of the knowledge, and the aim is to have a high rate of accuracy reached by the automation of diabetic retinopathy detection system.

Limitations: This project does not differentiate the levels of diabetic retinopathy. For future work, the model can not only differentiate the image of being diabetic retinopathy or but also the system will also predict the levels of diabetic retinopathy which is: no retinopathy, mild, moderate, severe, and proliferate retinopathy with efficiency using GPU systems. A single system with higher accuracy is proven to be good recognition of the disease.

Acknowledgements I want to stretch out my genuine gratitude to all who helped me for the undertaking work. I want to earnestly express gratitude towards Prof. Avinash Shrivastava for his guidance and steady supervision for giving vital data with respect to the undertaking likewise, for their help in completing this task work. I would like to offer my thanks towards my mates and individuals of Vidyalankar Institute of Technology for their thoughtful co-activity and support.

References

1. <https://www.mayoclinic.org/diseases-conditions/diabeticretinopathy/symptoms-causes/syc-20371661>
2. <http://www.advancedeyecareny.com/retinopathy/>
3. N. Chakrabarty, A deep learning method for the detection of diabetic retinopathy, in *5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (2018), p. 978
4. E.V. Carrera, A. Gonzalez, R. Carrera, Automated detection of diabetic retinopathy using SVM, in *IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)* (IEEE, 2017)
5. M. Chetoui, M.A. Akhloufi, M. Kardouchi, Diabetic retinopathy detection using machine learning and texture features, in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)* (IEEE, 2018)
6. A. Herliana, T. Arifin, S. Susanti, A.B. Hikmah, Feature selection of diabetic retinopathy disease using particle swarm optimization and neural network, in *The 6th International Conference on Cyber and IT Service Management (CITSM 2018) Inna Parapat Hotel—Medan*, IEEE, 7–9 Aug 2018
7. S. Kumar, B. Kumar, Diabetic retinopathy detection by extracting area and number of microaneurysm from colour fundus image, in *5th International Conference on Signal Processing and Integrated Networks (SPIN)* (IEEE, 2018)
8. K.K. Palavalasa, B. Sambaturu, Automatic diabetic retinopathy detection using digital image processing, in *International Conference on Communication and Signal Processing*, IEEE, India, 3–5 Apr 2018
9. High resolution fundus retinal image database: <https://www.kaggle.com/c/diabetic-retinopathy-detection/data>
10. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

Breast Cancer Classification Using Machine Learning Algorithms



Simran Sharma and Sachin Deshpande

Abstract Breast Cancer is a disease that is depicted as anomalous and unwanted cells that increase and affect regular cells in the body. Breast cancer occurs initially in the cells of the breast as malignant cells meet that can capture neighboring flesh or disperse to different areas of the body. As per the registered record provided by the World Health Organization (WHO), cancer affects 2.1 million women every single year. According to the latest report a year ago, 268,600 women expired from breast cancer which is approximately 13% of all deaths caused due to cancer. Different Machine Learning Algorithms are been handled for the detection and analysis of breast cancer and also to provide high classification and accuracy and effective diagnostic capabilities. The actual scope of the proposed design is to reach up the machine learning algorithm and to determine initial detection model for breast cancer which can predict the type of cancer, i.e., benign and malign, Doctor will examine patient based on physical examination process and then various input parameters are checked using the Random Forest algorithm and ensemble learning algorithm in which we have proposed bagging sampling and Decision Tree using CART. The Doctor will first examine breast cancer physically and secondly based on various input parameters. The data set here used will be UCI data-set and it will be dynamic.

Keywords Breast cancer · Random forest · Decision tree · Ensemble learning · Bagging sampling · CART

S. Sharma (✉) · S. Deshpande

Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India
e-mail: Simran.mars96@gmail.com

S. Deshpande

e-mail: sachin.deshpande@vit.edu.in

1 Introduction

Cancer cells start to invade in the body when healthy cells within the breast transform uncontrollable, which results in the formation of a mass or sheet of cells called a tumor [1]. A tumor is often cancerous (malign). A cancerous tumor is malignant, this means that it can breed largely and fast in the body and will spread to other organs or the neighbor tissues around the breast. A benign tumor means the tumor which will only grow in its size and it will not disperse to other parts or tissues of the body. Formation of Carcinoma is because of the mutations during a single cell which may cause unstoppable division of cells in a small amount of time [2]. Proper and expert detection and physical examination by oncology experts can help in time and proper diagnosis of the patient. This study addresses to create a computer-aided Prediction scheme to differentiate between benign and malignant. The fast evolution of computing algorithms has allowed for a novel approach to getting early prediction and diagnosis within the medical field, the fastest and popular field of technology is machine learning techniques. Machine Learning is predicated on four steps: Assembling data, proper model, training the model testing the model. To detect carcinoma accurately and mainly the sort (benign or malign) is that the main aim to make this model. The proposed model has machine learning algorithms; the algorithm such as the basis of random forest, a combined classifier that consists of multiple decision trees, which may figure out problems like classification and regression Algorithm (CART) [1]. The random forest is an ensemble method that consists of essential multiple algorithms. These models consist of a dynamic data set and have a feature selection of the input parameters. The Random Forest method selects variables randomly from many separate variables, to enhance the assessment growth, bagging sampling technique is employed to generate subsets then Decision Tree with CART algorithm is used for this model. Trees are grown with none shorten process just like the Decision Tree during which the CART method, which is the tree which is chosen gives the best certain results. Random forest method leans on the variation of classifiers [3].

This study uses the Wisconsin Database (WBCD) data considering UCI Repository. Information is based on classification of, benign and malignant. Data in it typically employed by analysts who use machine learning schemes and methods for the examination of cancer, exclusively for carcinoma.

2 Literature Review

Sara Alghunaim, Heyam H. Al-Baity scaled up the machine-learning model that is trained to classify by applying each data separately and then combine and hence used Apache Spark as a platform. It has used algorithms, i.e., SVM, Random forest, and also decision tree, and implemented few systems that help in the prediction of breast cancer. It gives a contingent study using three scenarios with the GE, DM, and

GE and DM together to produce the best outcome in terms of efficiency and error rate. It has compared different platforms to deal with a huge unwanted dataset which shows the experimental results as scaled SVM classifier is more efficient than the other classifier, as it shows the maximum efficiency and the minimum error rate with the GE database [4].

Shubham Sharma, Archit Aggarwal, Tanupriya Choudhury has proposed Machine learning models by using a set of tools used for developing and evaluating the algorithms that give a prediction, pattern recognition, and deep classification, Therefore there are two algorithms used, i.e., Naïve Bayesian Classifier and kNN. By using these two algorithms they had checked for benign and malign tumor. The purpose is to give an effective machine learning approach for the classification of cancer using two derivate classifiers in a data set. The act of each derived classifier will be analyzed and accuracy, and thereafter the training process and testing process are verified. Based on the comparison the paper has shown KNN has a higher efficiency but NB also performs well [3].

Ebru Aydindag Bayrak, Pinar Kirci, Tolga Ensari Proposed SVM and ANN techniques which are used for prediction of breast cancer by using machine learning methods to analyze betterment of performance. SVMs use different classes and samples. SVM can classify tumors as benign or malignant placed on the patient's age and tumor size [5]. Artificial Neural Network (ANN) has powerful functions which are non-linear. It consists of many separate networks of neurons. It is based on a local mathematical function which can be a weighted summation of inputs and generates an output of given threshold value. Effectiveness of the applied ML approach is correlated in terms of performance metrics [6].

Erwin Halim, Pauline Phoebe Halim, Marylise Hebrard stated the methods which are used in parallelism for designing the research including MRF (MMRF) segmentation for mammography, Histologic examination, and k-NN for detecting identification of the gene. A combination of the following gives the initial image and unites it with a refined image according to the closeness of pixel. MLP histopathology process is assumed to handout a positive implementation to breast cancer detection and also DWT-MMRF segmentation [1].

Mamatha Sai Yarabarla, Lakshmi Kavya Ravi, Dr. A. Sivasangari proposed to make use of the recent advances in the buildup of Computer-aided detection or diagnosis system and more techniques. The mainstay of the project is to check whether the person is having breast cancer or not. Random Forest algorithm gives a better outcome in terms of detection and accuracy because it uses regression as well as classification therefore if the user enters data it goes through different parameters and gets validated and gives the best accuracy [2].

3 Proposed System

Machine Learning uses the training data to train models to gauge patterns and uses the test dataset to evaluate the divining quality of the trained system. The proposed

model will evaluate the performance of the algorithms that are used in the prediction of breast cancer using the UCI data set. Segregation of data set in the training and testing model is the most elementary and important part of the evaluation of data mining models. Generally, when a dataset is distanced into two parts, much of the data is employed for training, and a minor part of data is employed for testing. Next, the model is tested by building predictions against the test model. As the data in the testing set consist of known values for that attribute it is easy to determine whether the predicted output of the model is correct or not. In this proposed model, 80% of our data for training and 20% for testing.

The Random forest algorithm adopts the approach of bagging sampling to get K training subgroup from the initial dataset. The dimensions for every training subgroup are about two-third times of the first database, as well as each sampling is random and replaced into sampling [3].

Over the bagging sampling, the constructed K training subgroups form K decision trees. For the decision tree algorithm, the CART algorithm is presently used. Method of node sampling within the CART algorithm is the most crucial and important step of the algorithm. The CART algorithm uses the GINI coefficient method to perform node splitting.

Random forest is a supervised learning algorithm the elementary methodology of this algorithm is recursion, which may figure out problems such as classification and regression. The random forest algorithm uses an ensemble method that contains numerous fundamental algorithms. An ensemble of decision trees is created and then the bagging method trains the whole system the random forest can combine the calculated results by decision trees to optimize and give the desired output.

The CART algorithm is employed for building both Classification and Regression Decision Trees. The impurity (or purity) measure utilized in generating decision trees in CART is the GINI Index. Tree developed by the CART algorithm continually develops a binary decision tree with two child nodes of the main node [7].

Following are the input parameter which has been used in this proposed model which has been acquired from WDBC UCI:

ID

Diagnostics (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

- (a) radii
- (b) textures
- (c) perimeter
- (d) area
- (e) smoothness
- (f) compactness
- (g) concavity
- (h) concave points
- (i) symmetry
- (j) fractal dimension.

Algorithmic Descriptions

Bagging sampling:

Bagging is the simple and powerful ensemble method used in random forest to get training subsets from the original datasets. Bagging (Bootstrap Aggregation) is employed to reduce the variance of a choice tree. Set D of d tuples, at every repetition i, a training set D_i of d tuples is examined with restoration from D (i.e., bootstrap method). Then the classifier model M_i is accomplished for every training set D < i. Each classifier M_i return its class prediction. Bagged classifier M* calculates the votes and appoint the category with primary votes to X (Unknown sample) [8].

1. Various subgroups are generated from the first data set with equal tuples, choosing observations with restoration. 2. Base model is formed on all of those evaluated subgroups. 3. Each model is learned parallel from each training set and all the sets are independent of each other. 4. The final prediction is decided by linking together the entire prognosis from total evaluated models (Figs. 1 and 2).

Decision Tree:

The aim of adopting Decision Tree is to invent a training model that may figure class or value of destination variables by learning rules implied from preceding data. Each subjective node of the tree belongs to an aspect, and each leaf node correlates to a category label concerning the Bagging sampling, then constitutes K training subsets

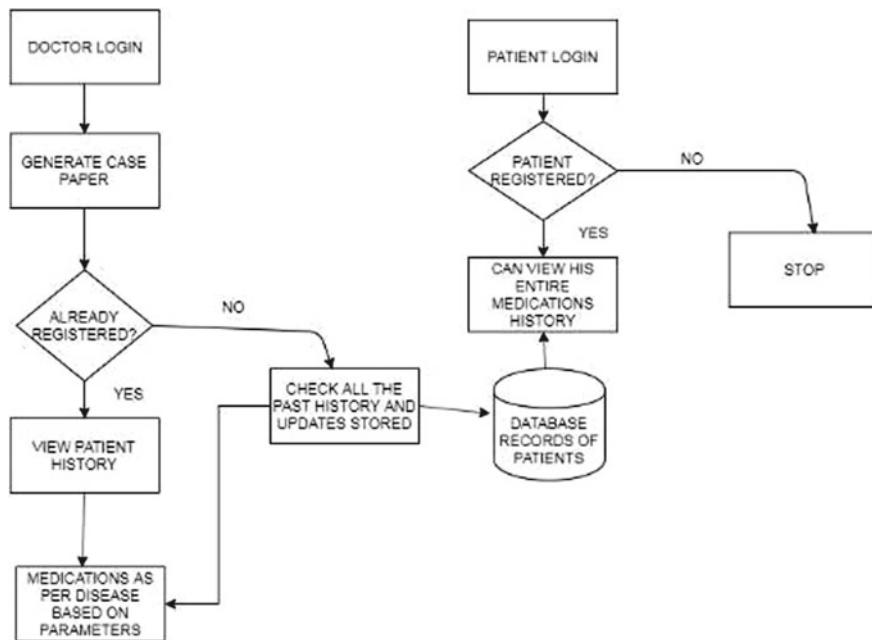


Fig. 1 Block diagram

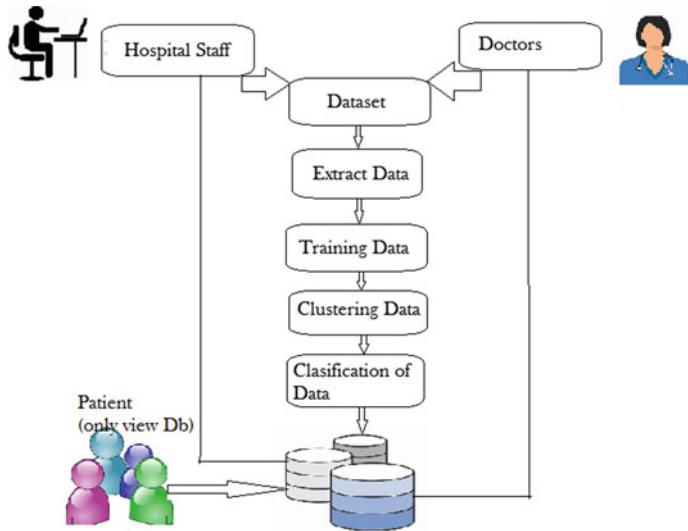


Fig. 2 Work flow model

from K decision trees. Random forest algorithm also uses CART (Classification and Regression) algorithms which are widely used in prediction models [8]. The frequently used members of CART (and any decision tree algorithm) are as follows:

1. Rules to split data at a node supports the value of that one variable;
2. Deciding of terminating criteria for a branch which is at the terminal node whether to separate further or to terminate; and
3. After evaluating these steps a prediction for target value is completed at each terminal node.

4 Conclusion

The proposed model in this paper presents a provisional study of different machine learning models, for disclosure and prediction of carcinoma performance comparison of the machine learning models, which has been administered using the UCI carcinoma dataset. Different reports of the algorithm determine that every one of the algorithms has observed the precise accuracy.

To find out different terms and know it during medicated statistics, many approaches to knowledge mining and machine learning models possible is higher. A crucial challenge in machine learning is to build certainty and decrease computing costs for classifiers to be exhibit in medical applications.

Random Forest outperforms other used algorithm during this sort of conditions thanks to its efficient ensemble learning algorithms and maybe authentically classifier. Random Forest algorithm, i.e., it has features which sort of a classification in

its family that builds upon a mixture of various decision trees. The random forest likewise works expertly in bulkier databases.

These are a few phases for processing:

- Define n tree bootstrap specimen through data.
- On every data, develops a tree. Randomly then select them from fluctuating tree for segregation on each tree node, tree segregates in a distinctive terminal node not as.
- Gather the data into n trees to evaluate the newest data like mass voting for distribution.
- Lastly, consider the extent of error out-of-bag (OOB) using data rather in bootstrap specimen. Thus have restricted carcinoma adopting a random forest algorithm. The expected output in research is more efficient than-off any other ML algorithm. The supervised machine learning algorithms are helping to be very supportive in early diagnosis and prognosis of cancer type altogether fields and types of cancer and in its further research (Fig. 3).

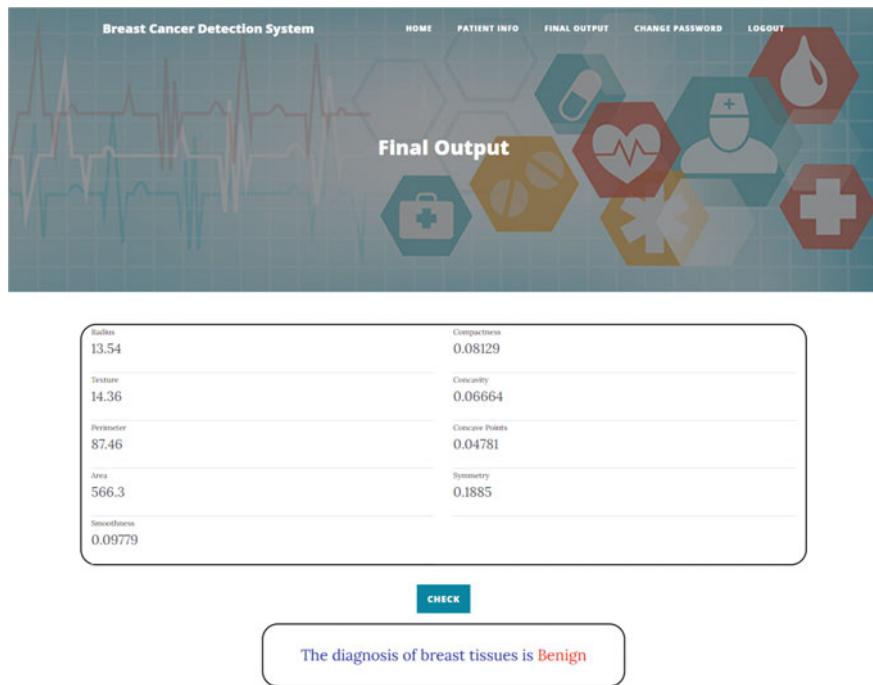


Fig. 3 Desired outputs of the model

Acknowledgements I want to stretch out my genuine gratitude to all who helped me with the undertaking work. I want to earnestly express gratitude toward Prof. Sachin Deshpande for their guidance and supervision at every step of the project. I want to offer my thanks to my folks and individuals from the Vidyalankar Institute of Technology for their thoughtful co-activity and support.

References

1. E. Halim, P.P. Halim, M. Hebrard, Artificial intelligent models for breast cancer early detection, in *2018 International Conference on Information Management and Technology* (2018)
2. M.S. Yarabarla, L.K. Ravi, A. Sivasangari, Breast cancer prediction via machine learning, in *Third International Conference on Trends in Electronics and Informatics (ICOEI 2019)* (IEEE Xplore, 2019). Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8
3. S. Sharma, A. Aggarwal, T. Choudhury, Breast cancer classification using machine learning, in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (2018)
4. S. Alghunaim, H.H. Al-Baity, On the scalability of machine-learning algorithms for breast cancer prediction in big data context. *IEEE Access* 7 (2019) <https://doi.org/10.1109/access.2019.2927080>
5. P. Sathiyanarayanan, S. Pavithra, M. Sai Saranya, M. Makeswari, Identification of breast cancer using the decision tree algorithm (2019)
6. E.A. Bayrak, P. Kirci, T. Ensari, Comparison of machine learning methods for breast cancer diagnosis, in *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science* (IEEE, 2019). 978-1-7281-1013-4/19/\$31.00
7. A. Bharat, N. Pooja, R.A. Reddy, Using machine learning algorithms for breast cancer risk prediction and diagnosis, in *IEEE Third International Conference on Circuits, Control, Communication and Computing* (2018)
8. Y.-Q. Liu, C. Wang, L. Zhang, Decision tree based predictive models for breast cancer survivability on imbalanced data of Biomedical Engineering School of Basic Medicine, Shanghai JiaoTong University Shanghai, China. yaqinliu@sjtu.edu.cn
9. M.S.M. Prince, A. Hasan, F.M. Shah, An efficient ensemble method for cancer detection, in *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019 (ICASERT 2019)* (2019)

A Strategic Approach to Enrich Brand Through Artificial Intelligence



Pranav Desai

Abstract The competition in market makes the scope broader for developing a strong brand for developing, gaining, and maintaining customer satisfaction and loyalty. The new-age technology makes it possible. Artificial Intelligence and Machine Learning are to be considered as most recent technologies and having several applications for effective filtration of pro data customization. This application can better contribute to developing a strong brand. Based on the inputs of such technology the ways get broaden to develop an effective branding strategy. The clear need for such analysis is the clear need of today's marketers. AI also aims to precise automated solutions developed without the help of much human interactions. The modern customer needs to get sensitized towards their specific preferences, rather to just purchase a product/services. For developing such customer-preferred experience, AI and ML make a significant differentiation to understand the needs of prospected clients. The behavior of the prospected client is to be recorded/analyzed by the AI and ML for making it as much as customized as per the preferences of a customer. This chapter explores the possibilities of developing better branding strategies with the help of AI and Machinery Learning driven branding experience.

Keywords Branding · Brand development · Applications of ICT · Machine learning · Artificial intelligence

1 Introduction

The new age technologies like Artificial Intelligence (AI) and Machine Learning (ML) have governed features and have been poor in various applications, most interestingly, they are capable to upset to versatile areas like analysis Media, Advertising, right to Law and also even for Medicine. Profoundly, major organizations

P. Desai (✉)

Indukaka Ipcowala Institute of Management, Faculty of Management Studies, Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat, India
e-mail: Pranavdesai.mba@charusat.ac.in

have straightened their gatherings around man-made consciousness study, instruments, and applied AI are centered around advancing the essential stages, to smear the man-made reasoning to items variety of transactions and consumer behavior and its administrations. The best performing companies across the globe are engaged for the most part putting intensely in AI to upgrade their whole biological system of items [1]. Practically, all shoppers have a tiny enabled device with them, AI in. The chance of clever contexts is significantly more intense as great analytical sensors become more astute. Truth be told, investigation into significant registering can possibly open an exponential jump forward as far as handling abilities [2].

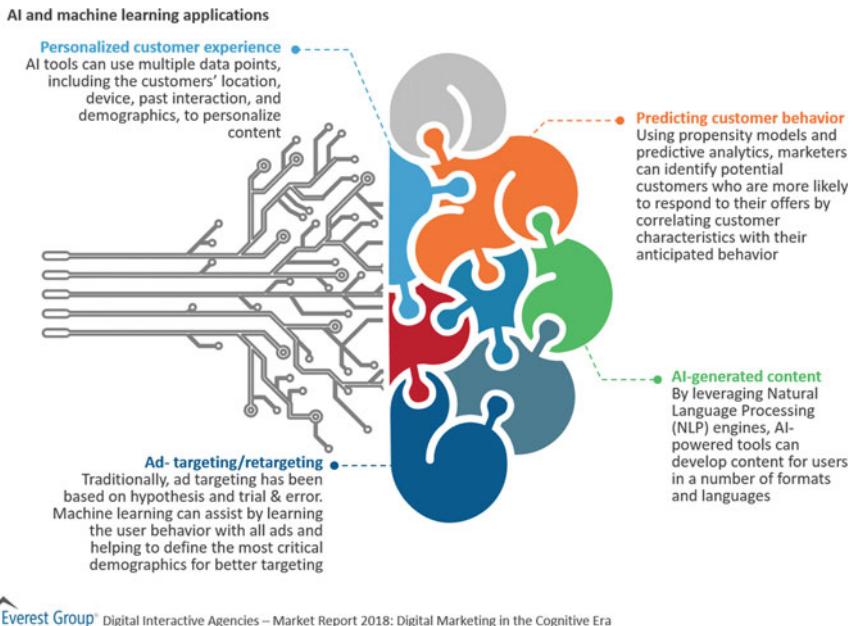
2 Applications of AI in Branding

Advanced Marketing has experienced remarkable development in the most recent decade and has become a basic piece of the general business procedure. Undertakings are putting resources into highlighting innovation, ability, apparatuses, and innovative practices to drive business development and enhance consumer loyalty through [3] advanced technology-based promoting activities. This expansion popular from ventures has changed the computerized promoting administration landscape with an assortment of players entering this space [4]. Advanced intelligent offices are confronting extraordinary rivalry from the conventional showcasing firms, consultancies, and even IT specialist co-ops. These players have extended their capabilities through different acquisitions and interests in intuitive offices. With the developing use of AI and psychological, advanced promoting will turn out to be increasingly customized and keen. Undertakings are quickly embracing AI and cognitive innovations to create advanced advertising methodology and give consistent Omnichannel client experience to customers with customized and portable prepared substance. These rapid changes are making enterprises rethink their strategy and leadership structure, with the CMO role facing pressure to adapt and perform. Digital interactive agencies are in turn capitalizing on the changing enterprise requirements by expanding their service portfolio to offer end-to-end digital marketing services (Fig. 1).

2.1 Strengthening Consumer Engagement

2.1.1 Developing Consumer Insights

In addition to the ability of machine learning's to gauge massive formless details, the most significant benefit against all other applications is its ability to unearth tacit, spontaneous consumer insights. With the use of algorithms, for frictions, investigators can link these findings to behavioral consequences [5]. The key strategic consequences of the same is to develop the effective process framework for providing a



 Everest Group® Digital Interactive Agencies – Market Report 2018: Digital Marketing in the Cognitive Era

Fig. 1 The integration of AI and ML for better understand the needs of customers

structured classification to amorphous data and categorize the specific personality traits that are linked to spectators prompts and conducts. There is a various ways to division, so devouring the right plan and creative mate in place is vital.

2.1.2 Inculcation of Creativity

The presence of deep learning and several kinds of neural nets encouraged a interspersed progression in system developing and discourse interpretation of consumer actions. This made the way for interpretation of the various cases and to offer influence of specific brands to upgrade their imagination and structure. With goal crowd bits of knowledge, the brand develops can use AI to analyze and describe gigantic outside formless evidence to comprehend what kind of informing and inventive components request to their objectives. This will help to develop the intelligence plan components for possessed and paid chattels [6].

2.1.3 Enrichment of Experience Design

The capacity to construe the amorphous information appraises new information sinks with expert system, progression, and to design the brand strategy. The most important phase is to distinguish the precise happenstances inside the area to constre and

recognize matter of identification and arrangement for smart suggestions. Additional important constituent is working with the precise picture processor to distinguish the correct pictures and plan the actions to feel provide necessary set of information to enhance the experience ought to look like from a structured way of decision making process.

2.1.4 Developing the Brand Advocacy

The extended strength of machine learning is the ability to distinguish and regulate innovative influencers way of understanding the consumer insight. With unstructured social information, the sturdy brand developers are able to distinguish major influencing factors within the development territories just as sympathetic the kind of substance to reverberate with their follow set of people [7]. This makes it possible to brand architect to contribute with focusing on the accurate influencers with the right ingredient to build mindfulness and diffusion.

2.1.5 The Ability to Improve Unknown Personalization

The blend of AI and proprietary data can help to categorize mysterious clients visiting your site by their personal conduct standards. With these bits of knowledge, it has been believable to focus on these clients with the correct and precise information and work upon the same to take necessary actions based on the expectations of the prospected customers.

2.2 Stirring the Brand Planning

Planning for developing a strategy for a brand is intended to develop a correct brand stage that can keep up authority as per the precise categorization. Various components are required to study that goes into building up a solid stage in support of a solid brand identity, reason, guarantee, qualities, informing tone, and maybe above all, a pertinent incentive. In any case, more significantly, they have to persist in first vivible of marketplace powers and disturbances from intensifying players. Before the new age technologies, the learning was getting the hang of, developing, and reexamining situating was compelled to the exclusive information and market knowledge information foundations, that are con-stressed themselves by how quickly they make it possible to distribute the information.

Machine learning and Artificial Intelligence are working effectively to make it able to add unstructured information sources to outline the categorized prospected client into the target audience over a predetermined period to distinguish them for better coverage to them. The noteworthy point here is to point a precise understanding to see how customers inside the class see the brand image, comprehend their worth,

and whether they are in aligning to the predefined goals. Such practices deal with the specific nature of the case; as well the specific requirement to see how your item has been seen with passage of time to differentiate growth and openings progression of new purchases.

2.3 Corporate Strategy

Like same as AI helps to change brand stages, organizations can use AI to drive an incentive over the bunch of the selection of a customer, overcome developing market area/products, and improve the method of targeting and dealing as per the values and beliefs of the customers.

2.3.1 The Enchantment of Automation

AI became key for creating a brand, with the expectation of effectively handling the high volume details that are involved daily schedule and repeating work. The outlines can identify designs that ultimately lead to decrease the cost and increments in productivity. Prime outcome of this process is inclinations to recognize a target, for example, exceptionally monotonous assignments that have a squat interest in understanding the behavior. Procedures for developing a decision trees looks like a perfect mechanization focus for insightful frameworks.

2.3.2 Leveraging the Mergers and Acquisition

With the correct space of information and scientific classification, organizations utilize chronicled and present state execution information to advice choices based on which the tangential markets to enter, and within that, organizations, in view of explicit signs, to air conditioning quire. The significant contribution of the same is to distinguish the correct information yourintakes to develop the calculations to cause the correct forecasts on which chances to line up with your plan of action and show the correct development potential.

2.3.3 Enhancement of Product Portfolio

Organizations can make use of AI frameworks to identify item upgrade and progress preambles inside the portfolios. By making scientific classifications around purchaser discussions, you can pick up experiences into new items or recognize future headings for highlights and structures of items inside your current portfolio. Added way is to apply AI to the particular handpicked first gathering information to see how

the existing clients are discussing the items to know the preferences as well as improvement in new openings.

2.3.4 Effective Acquisition of Talent

Finding and procuring ability can be the most troublesome undertaking for any administrator or pioneer inside any organization. Simulated intelligence's capacity to perceive designs through support learning can assist directors with understanding execution indicators on how well new ability will perform and build up the correct designs for the ability to learn and develop inside the weights of that particular condition.

2.3.5 Interrogating the Consumer State

The best innovation distortions into shoppers' experiences, possibly to bring when they need it. Computer-based intelligence will help drive this move by insight-fully associating with different gadgets to figure out which innovation, information, and substance ought to be available at that point. This implies customers will designate the medicinal less significant choices and errands to their own partners and spotlight on progressively significant choices or exercises like making content or encountering new conditions or places by means of blended reality. For architects of a brand, this implies that it is a requirement to open the opportunities where we can practice evidence and revolution in processes to affect specific contacts, and distinguish the initiatives to engage makeconsumers to set the strong brand belief and/ or conjure association through intense encounters [4]. The key here is building a system that recognizes what undertakings clients have assigned to their boots and characterizing a procedure that uses receptive information for the bot to accomplish logical mindfulness.

3 Strategic Mapping of Brand Through AI

The advanced algorithms and calculations through advanced software are used to screen and analyze with the objective to get better insights to comprehend and better trap the sources of information to work upon the expectations of the customers [5]. Most marketers are having a perception to use AI for media purchasing. Advertisers trust AI can improve focusing on and personalization for media situations in this way helping decline the degree of recurrence required (Fig. 2).

Phase I-Reach To start to understand a buyer's journey, "Reach" is the first step. This phase aims to aware of the maximum number of prospected customers and provides an appealing involvement to transform them into purchasers.

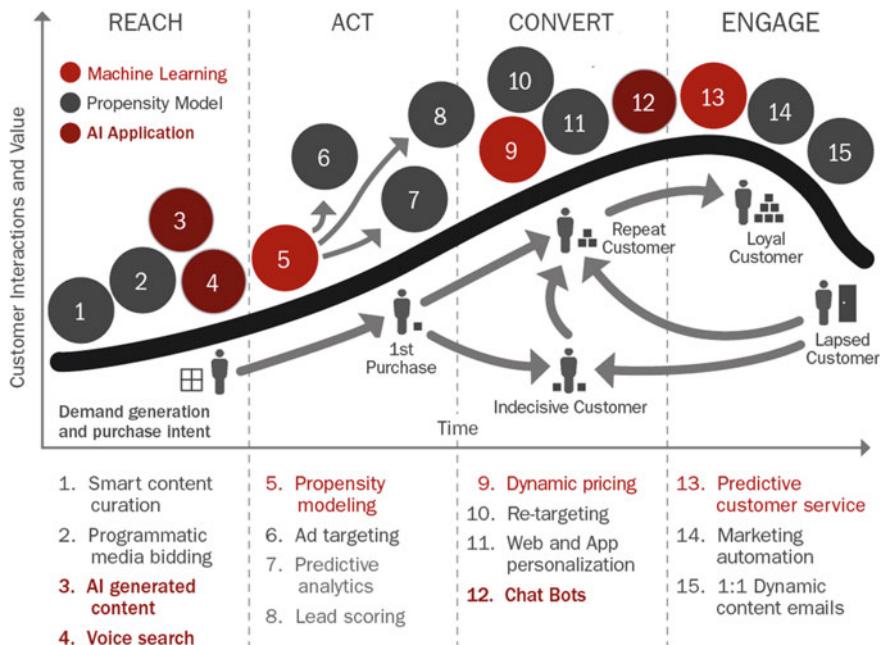


Fig. 2 Applications of ICT in branding through the propensity model. *Source* Smart Insights–Financial Brands

- Curation of Smart Content:** In this phase, visitors content including the demographic profile and the prospect client details are gathered. This includes reference suggestion that includes the products, offers, and based on the preferences of a customer.
- Procurement of Programmatic Media:** This identifies with the utilization of propensity models to better understand target advertisements for better coverage of the targeted customers. AI and machine learning make it possible to select possible media vehicles to be select to display/broadcast web banners/advertisements.
- System Generated Content:** AI content composing projects can choose components from various audio string collection and structure them for identifying the combination of other demographic details and target them for cold calling.
- Speech Hunt:** To better coverage of the prospected customers, the revolution focused by AI is tied in with using the innovation created by the significant outsourcing organizations having mastery in it to assist increment natural pursues.

Phase II-Act The features of the product/brand influence a lot in the next phase of the journey of Consumers.

- Shaping the Propensity:** With the use of large amounts of historical data to forecast the recent trend, Dr. Dave developed modeling and working effectively

for predicting the present reality. AI at this phase assists with guiding prospected clients through the correct posts that have been developed through the outbound customized content.

6. ***Steering of Advertisements:*** Based on the authentic information to select the better media vehicle, that makes the positive impact of a brand on explicit individuals for motivating in the purchasing procedure. Such activity results in better compelling promotion situations and substance than conventional strategies [3].
7. ***Prognostic Analytics:*** Using inclination models can help decide the probability of an offered client to change over, anticipating what value a shopper is probably going to change over, or which clients are destined to make rehash buys. The key here is exact information [2].
8. ***Prime Counting:*** Scoring leads is the way toward utilizing prescient examination with the goal that a business group can set up how ‘hot’ a given lead is, and on the off chance that they merit they may further be processed on the same.

Phase III-Transform (Convert) In this phase, the prospected clients/customers are transforming/converted to the purchaser or a member.

9. ***Active Assessing and Pricing:*** The changing valuing utilizes AI to create extraordinary offers just for those possibilities prone to require them so as to convert. This implies you can build deals without lessening your benefit damage gins by a lot, in this manner expanding benefits.
10. ***Re-Aiming:*** This optimizes the advertisements for making customers influence repurchasing based on the past/demographic details/purchase patterns.
11. ***Web and App Personalization:*** An incredible asset, you can utilize penchant models to customize a website page or application dependent on where the purchaser is on their excursion to buy.
12. ***The effective use of Chatbots:*** Chatbots use AI to emulate human knowledge, deciphering customer requests, and finishing orders. On the off chance that you are keen on building a chatbot for your image inside the Messenger stage, Facebook has created directions for how to do as such.

Phase IV-Engaging the Customers After a purchase/deal, it is critical to continuously build engagement and faithfulness with the goal of extending the relationship and possibly producing referral business [8].

13. ***Prescient Customer Service:*** Predictive examination driven by AI can be utilized to figure out which clients are well on the way to either get lethargic or leave by and large. With this knowledge, you can connect with these clients with offers, prompts, or help to keep them from beating.
14. ***Automation in Operations of Marketing:*** Machine learning can and prescient examination can be utilized to decide the best occasions to reach a client, what words ought to be utilized in the correspondence, and significantly more. These insights can improve the adequacy of your promoting customization endeavors.
15. ***Customized Emails:*** Predictive examination utilizing a penchant model can utilize past conduct to advance the most pertinent items and administrations

in email correspondence as a feature of the onboarding process. The outcomes from these interchanges are then taken care of into the models to improve brings about what's to come [4, 8, 9].

4 Discussion

Retail and shopper items associations are entering another era of technological innovation—with canny mechanization at its center. The consequences are significant, offering a large group of beforehand incomprehensible abilities—from consequently rerouting shipments to sidestep competitive environment, to customizing in-store administrations dependent on the investigation of a client's outward appearances and facial expressions. The most up to date look into shows that two of every five retailers and brands are now utilizing clever computerization, and that number is on target to twofold in upcoming years.

References

1. C. Sengupta, Y. Joshi, Digital interactive agencies—market report 2018: digital marketing in the cognitive era (2017)
2. J. Hull, A. Wilson, C. Olson, Digital assistants: reordering consumer lives and redefining digital marketing (Rep.) (2017)
3. D. Jurafsky, J.H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Dorling Kindersley Pvt., India, 2014)
4. G. Mazurek, Virtualization of marketing—conceptual model, in *Proceedings of the 2011 International Conference on Marketing Studies (ICMS2011), Academy of Taiwan Information Systems Research (ATISR)*, Kuala Lumpur, 9–11 Sept 2011, pp. 220–229
5. D. Forsyth, J. Ponce, *Computer Vision: A Modern Approach* (Prentice Hall, Upper Saddle River, NJ, 2011)
6. M. Ehsan Malik, B. Naeem, Aaker's brand personality framework: a critical commentary. *World Appl. Sci. J.* **24**, 895–899 (2013)
7. J. Kantola, P. Lillrank, E. Pöyry, The effectiveness of retargeting in online advertising. Unpublished master's thesis, Aalto University School of Science, 2014
8. D. Court, D. Elzinga, S. Mulder, O.J. Vettvik, The consumer decision journey. *McKinsey Quarterly* (2009)
9. G. Mazurek, Network value creation through marketing, management & business administration. *Cent. Eur.* **22**(4), 70–77 (2014)

Real Estate Price's Forecasting Through Predictive Modelling



Nitin Sharma, Yojna Arora, Priyanka Makkar, Vikas Sharma,
and Hardik Gupta

Abstract The intelligence and the power of making decisions by the machines and devices are the future of computer technologies and software industry. It includes analytics, predictions and forecasting. A correct and complete set of data and information are required for making predictions accurate. Data is not worthy until it provides us proper patterns and insights. A process of scrubbing is required in order to transform the data, suitable for analytics. Thus, data analytics sometime proven as a challenging job. For handling these challenges, a proper cleaning and transformation are required. In this research, I am going to tackle these problems and show all the phases of data analytics.

Keywords Real estate · Predictive modelling · Data science life cycle · Correlation · Supervised learning · Unsupervised learning · Web scrapping · House value · Exploratory data analytics

1 Introduction: Data Mining

Machine learning is about performing variety of tasks on the behalf of intelligence of the machine, without help of a human. In nearly all the fields of engineering and technology, machine learning is giving contribution to saving the cost of labours, improving quality standards and production statistics. This paper is about helping people by forecasting the prices of lands through predictive modelling. We are predicting the household values in different locations by the use of machine learning and the structural idea of machine learning is to develop programme in such a way so that computers can learn from the given resources and can take decisions on its own. Here, learning represents that by the use of various statistical techniques by which we

N. Sharma (✉) · Y. Arora · P. Makkar

Department of Computer Science & Engineering, Amity University, Gurugram, India

e-mail: wy.e.sharma@gmail.com

V. Sharma · H. Gupta

Department of Computer Science & Engineering, Chandigarh University, Mohali, India

are improving the accuracy and quality of the decision. This is an automatic process in which expenses are very less so people do not need to invest their time and money. So it will help people in saving cost and time. Data analytics is a part of machine learning in which machines learn from the data and predict some important aspects called insights for the business. This is called as ‘predictive analytics’ in the industry.

2 Data Science and Analytics

2.1 Exploratory Data Analytics (EDA)

EDA is a technique for exploring the data in such a way that we can finalise the key terms for the specified data set. It consists of cleaning, analysing and visualising the patterns and insights driven by the data. It helps us in encapsulate important information and characteristics of data. For doing this, we create various kinds of charts such as—bar graph, pie charts and scatter plots.

2.2 Types of Machine Learning

A. Supervised Learning

As the name suggest, an instructor or a teacher will be present in this kind of machine learning algorithm. We train our model with proper labelled data. Supervised learning algorithms try to make relationships and dependencies between variables, respectively, called ‘features’ and ‘labels’. On behalf of these relationships, algorithms learn from the data and predict the output.

B. Unsupervised Learning

There will not be any trainer or instructor, therefore, no training or teachings will be providing to the machine. Unsupervised learning deals with unlabelled data in which we cannot measure relationships and dependencies in the data. In this kind of machine learning, machine tries to group the unsorted data into clustered based on the patterns and similarities in data.

2.3 Life Cycle of a Data Mining Project

A. Data Acquisition

It is the process of collecting and gathering the data from multiple heterogeneous sources. Data can be gathering from data warehouses, data marts, collective online

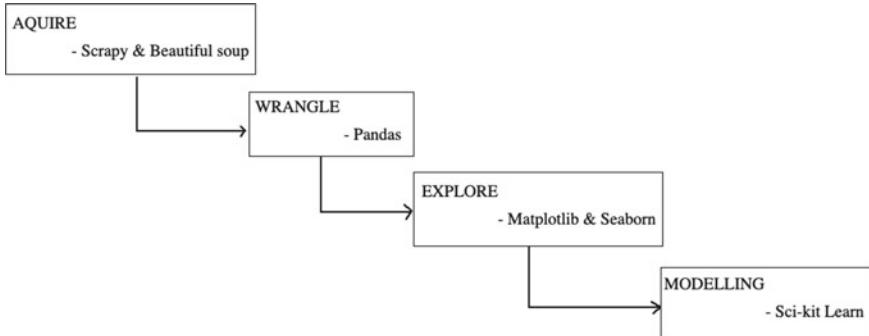


Fig. 1 These are the four steps that we need to follow in order to complete a data science project

and online surveys. I have gathered the data from Internet via <https://www.kaggle.com>. We can gather data sets from various resources like arranging an application programme interface (API) and organisational databases. There are various types of data available like csv (comma separated values), Excel files and zip files consisting of other formats (Fig. 1).

Web Scraping Internet is a very rich source of data and web scrapping is the best way for doing that. Web scraping also known as data fetching or web harvesting is a process or technique to pull out data from the websites. It can be done manually but regularly the term use for automatic process using programming or a bot (automation scripts). Web scrapping can be done in Python using a framework called Scrapy. It provides a complete bundle of techniques without manually writing the code. Beautiful soup is another framework that can be used for web scraping for parsing XML as well as HTML recodes. We can speed up this process by automated scrapping.

B. Data Wrangling

Data wrangling also called data cleaning is a very important step in data analytics and machine learning. Data cleaning is the procedure in which we spot incomplete, irreverent and incorrect set of entities and have to deal with this accordingly. Simply the data is being scrubbed in this phase. Data cleaning consists of various methods.

- **Missing Values**

This is a common problem arrives whenever we are going to start a new data science project. In real-time data, most of the data sets consist of missing values. Dealing with missing values is very important because if we left them as it is, the results will be affected and may not be accurate. What we can do is we can fill these missing values with some valid entities or we can simply delete them. Pandas is a framework in Python that provides us various methods to clean our data. For missing values, it has a method called ‘fillna’ by which we can fill these missing values with some entities like mean, median, mode or zero. It also has a method ‘drop’ for deleting these values.

- Anomalies

Anomaly or outliers are nothing but the entities of data that differs from other entities. For example: we have a set {140, 190, 17, 130, 110, 180}. The set consists the values between 100 and 200 except 17 that is obviously different from other values. We can use a box plot or a scatter plot for dealing with these outliers. This process is also known as anomaly detection.

- Duplicate Values

In real world, data sets can also contain duplicate or identical values that can also affect our results or predictions. Pandas provides us a method named as ‘drop_duplicates()’ by which we can delete duplicate entities from the data set.

- Reformatting

Another challenge is reformatting of data. It is the process in which we define the singularity of the column by converting all the values under a defined standard format. For example: we have a data set contains ‘date’. Like: [10/02/1999, 2008/04/19, 10-05-1998, 22.01.1994, 15-Jan-2020]. We will convert this in like [10-02-1999, 19-04-1008, 10-05-1998, 22-01-1994, 15-01-2020] (Fig. 2).

C. Explore

Exploratory data analytics (EDA) will be performed in this phase. We will create histograms, scatterplots, graphs and charts to see the relationships between the

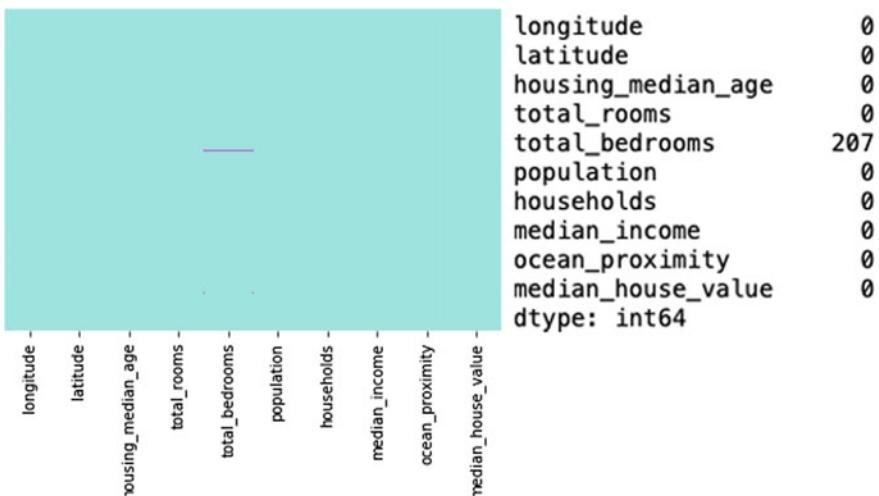


Fig. 2 Graph is called a heat map that is representing NaN values in the data and it shows us all the columns that contains NaN values

features of the data set. It will also show the important insights and characteristics of data. In this process, we will check for outliers, test hypothesis with the help of pictorial representation and graphs.

Correlation Correlation defines the measurement of association or link between two or more than two variables. It is of two types.

- Negative correlation: when the value of variable ‘ x ’ grows, the other one ‘ y ’ reduces.
- Positive correlation: when there is an increase in ‘ x ’ y also grows.

Formula of correlation

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2} \sqrt{\sum (y - \bar{y})^2}}$$

where

r = coefficient of correlation,

\bar{x} = the average of the calculated observations of x ,

\bar{y} = the average of the calculated observations of y .

D. *Modelling*

This is the phase in which we create our actual machine learning model. A machine learning model is nothing but a mathematical computational representation of real-time process. Modelling phase consists of four steps. This phase consists of mainly four steps (Fig. 3).

- Feature Engineering

Feature engineering is the process of constructing new attributes from the existing attributes. For example, there are two columns present in our data set [length, width] and we need area for some reason and we can construct a new feature named ‘area’ by multiplying length and width together. We can train our model more efficiently using feature engineering.

- Dimension Reduction

Dimensionality reduction is the process of removing unwanted features from our Data. For example: we want to predict the salary of an employee of an organisation, the salary is dependent on his experience, education, projects not on father name, mother name and phone number, so we will remove those features so that the accuracy of our model will be better. Techniques of dimension reduction: backward elimination, forward elimination, principal component analysis (PCA).

| | longitude | latitude | housing_median_age | total_rooms | total_bedrooms | population | households | median_income | median_house_value |
|--------------------|-----------|----------|--------------------|-------------|----------------|------------|------------|---------------|--------------------|
| longitude | 1 | -0.92 | -0.11 | 0.045 | 0.068 | 0.1 | 0.055 | -0.015 | -0.046 |
| latitude | -0.92 | 1 | 0.011 | -0.036 | -0.065 | -0.11 | -0.071 | -0.08 | -0.14 |
| housing_median_age | -0.11 | 0.011 | 1 | -0.36 | -0.32 | -0.3 | -0.3 | -0.12 | 0.11 |
| total_rooms | 0.045 | -0.036 | -0.36 | 1 | 0.92 | 0.86 | 0.92 | 0.2 | 0.13 |
| total_bedrooms | 0.068 | -0.065 | -0.32 | 0.92 | 1 | 0.87 | 0.97 | -0.0073 | 0.049 |
| population | 0.1 | -0.11 | -0.3 | 0.86 | 0.87 | 1 | 0.91 | 0.0048 | -0.025 |
| households | 0.055 | -0.071 | -0.3 | 0.92 | 0.97 | 0.91 | 1 | 0.013 | 0.066 |
| median_income | -0.015 | -0.08 | -0.12 | 0.2 | -0.0073 | 0.0048 | 0.013 | 1 | 0.69 |
| median_house_value | -0.046 | -0.14 | 0.11 | 0.13 | 0.049 | -0.025 | 0.066 | 0.69 | 1 |

Fig. 3 Coefficients of correlations correlation can be calculated as coefficient of correlation that is the value between -1.0 and $+1.0$. If the value of correlation is exactly 1 that means it is a perfect positive correlation

- Splitting of Data

The partition of data into training set and testing set will be done in this phase. Training set is use to teach the patterns in the data to the algorithm and the testing set is use to evaluate the results. We can split the data using `train_test_split()` method in Python using `sklearn.preprocessing` module. There is an attribute named ‘test size’ in which we can assign the size of our split. For example: we want 80% of our data set is for training set and 20% is for test set then we will pass `test_size` as 0.2 in the parameter.

- Model Creation

We will apply various kinds of machine learning algorithms and see the fluctuation in the results. We will determine which algorithm is better for forecasting. Our data is contiguous in nature so we will use regression algorithms, such as—linear regression, polynomial regression, decision tree regression and random forest regression. We can use any linear model form the `linear_model` module in Python. For categorical data, we will use classification techniques that include logistic regression, decision tree classifier, etc.

3 Features

Features are very important part of forecasting. After forecasting, we can help people in finding the right area for their living on behalf of the views of the people who already lives in that particular area. The views of these people are called ‘features’ in predictive analytics and forecasting (Fig. 4).

```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'ocean_proximity', 'median_house_value'],
      dtype='object')
```

Fig. 4 There is a list of the features of our data set in the diagram

3.1 Longitude and Latitude

Longitude and latitude are two measured numbers by which we can locate position of any place on the surface of earth. These are the parts of geographical coordinate system. There are two features in our data set, respectively, ‘longitude’ and ‘latitude’ that are representing the exact location of the house.

3.2 Ocean_Proximity

This is a categorical feature in the data which tells us about the location of the house with respect to the ocean. It answers the question, from the ocean, how far the house is? There are five categories in this column: near bay, near ocean, island and inland. Sometimes, people prefer a house near the ocean at that time this classification helps us in determining the location and prices of houses.

3.3 Population

This feature tells us about the total inhabitants in the area. It answers the question ‘How much people lives in a particular area?’ By taking the views of the people, we can predict the surroundings of a particular area that help people finding a right society for their living. Somebody who wants to own a shop as well near his house then we will recommend a house in highly populated area.

3.4 Total_Rooms, Total_Bedrooms and Median_House_Value

Median_house_value is the price of the house for purchase. Median_house_value is also the output variable or label. We will compare the prices of the houses on different locations. These are contiguous values that is why we are applying linear modelling.

Total_rooms and bedrooms feature contains the number of rooms and number of bedrooms, respectively. We can the addition of both of them in order to find out the area of the house for predicting the house value.

4 Results

This paper is to predict the house value of houses in California's state. It includes the information and raw data about the houses like—longitude, latitude, number of rooms, etc. This model can predict housing prices in California's different districts. We explored the data using advanced visualisation techniques. We applied decision tree and random forest techniques for the predictions and had an accuracy of 80% in our model. It is a ‘regression’ problem because the label ‘median_house_value’ is continuous. We will apply all types of regression and predict the median_house_value. We will compare which type of regression is best for this prediction.

5 Conclusion

Our main goal was the understanding of data mining and predictive analytics using real estate data in which the label ‘median_house_value’ was contiguous in nature that is why we have used supervised machine learning algorithms for the training of our model by which we are predicting the output. This paper was to predict the value of houses in California's state. It includes the information and raw data about the houses like—longitude, latitude, number of rooms, etc. Constructing new features is a very important step in data science known as feature engineering that can be done manually by the data analyst. Like in our paper, we can sum up the values of two columns, respectively, ‘total_rooms’ and ‘total_bedrooms’ for finding the area of the house because area can help us in comparing the value of the houses. Area can also be the major feature that helps us in understanding the actual quantity of land that is acquired by our house. In data preprocessing, we have a step known as label encoding in which we have to encode the categorical features into integer values so that it will not disturb us when we will perform statistical operations on data. Sklearn provides us a predefined package preprocessing that have the methods like ‘labelencoder() & one-hotencoder()’. From these predictive analytics, we can forecast the houses values and recommend the houses to the customers on behalf of their requirements.

References

1. M. Cain, C. Janssen, Real estate price prediction under asymmetric loss. *Ann. Inst. Stat. Math.* **47**, 401–414 (1995)
2. D. Sun, Y. Du, W. Xu, M. Zuo, C. Zhang, J. Zhou, Combining online news articles and web search to predict the fluctuation of real estate market in big data context. *Pac. Asia J. Assoc. Inf. Syst.* **6**(4), 19–37 (2014)
3. A. Páez, Recent research in spatial real estate hedonic analysis. *J. Geogr. Syst.* **11**(311) (2009)
4. C. Jichun, W. Fengfang, Application of BP neural network in the prediction of real estate price's chronological sequence. *Stat. Decis.* **14**, 42–43 (2008)
5. O. Jiantao, The application for real estate investment price by nonlinear gray forecast model. *Ind. Technol. Econ.* **24**(5), 78–80
6. H. Xiaolong, Z. Ming, Applied research on real estate price prediction by the neural network. *IEEE Xplore*, 9 Sept 2010
7. B. Park, J.K. Bae, Using machine learning algorithms for housing price prediction: the case of fairfax county, virginia housing data. *Expert Syst. Appl.* **42**(19), 6806 (2015)
8. R. Duplin, Predicting house prices using multiple listings data. *J. Real Estate Financ. Econ.* **17**(1), 35–59 (1998)
9. G.-Z. Fan, S.E. Ong, H.C. Koh, Determinants of house price: a decision tree approach. *Urban Stud.* **43**(12), 2301–2315 (2006)
10. P.K. Asabere, F.E. Huffman, Price concessions, time on the market, and the actual sale price of homes. *J. Real Estate Financ. Econ.* **6**, 167–174 (1993)
11. T. Luo, K. Kramer, D.B. Goldgof, L.O. Hall, S. Samson, A. Remsen, T. Hopkins, Active learning to recognize multiple types of plankton. *JMLR* **6**, 589–613 (2005)
12. P.D. Turney, Types of cost in inductive concept learning, in *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California (2000), pp. 15–21

Bitcoin Cost Prediction Using Neural Networks



Nitin Sharma, Yojna Arora, and Priyanka Makkar

Abstract Bitcoin is the digital currency and the worth of it is dependent on the mining network, if the mining network is complex the cost would be high and vice versa. In this paper, the Long Short term Memory version of recurrent neural network is used, to forecast the value of bitcoin. For the purpose to enhance a better understanding of pricing and overall idea of this excellent process, the paper first give a brief overview of Bitcoin and its economy then description of the data which contains information from the stock market indices, blockchain, coin marketcap and sentiment. The paper provides a prediction model to predict the cost of bitcoin and the use of LSTM for structures with a defined time series. To conclude, the analysis has set forth the results of the Bitcoin price forecast for the next 30 and 60 days.

Keywords Bitcoin · Deep learning · Blockchain · Prediction

1 Introduction

By viewing the aspects Bitcoin 10 years ago when it is undergoing revolt. Bitcoin dispatches itself as the framework that settled the dual invest problems, persistent matter for innate digital hard cash network. However, the affect in forthcoming years was remarkable [1]. Scattered record Technologies, Crypto currencies etc. All have come from the thought of “Bitcoin.” This is allocated, to the special dissolution hybrid accompanied by inborn motivation. On the supplement of this gamut, with the facts existing are considered as an oil in today’s world, together with the massive

N. Sharma (✉) · Y. Arora · P. Makkar

Department of Computer Science & Engineering, Amity University, Gurugram, India
e-mail: Wye.sharma@gmail.com

Y. Arora
e-mail: Yarora@ggn.amity.edu

P. Makkar
e-mail: Pmakkar@ggn.amity.edu

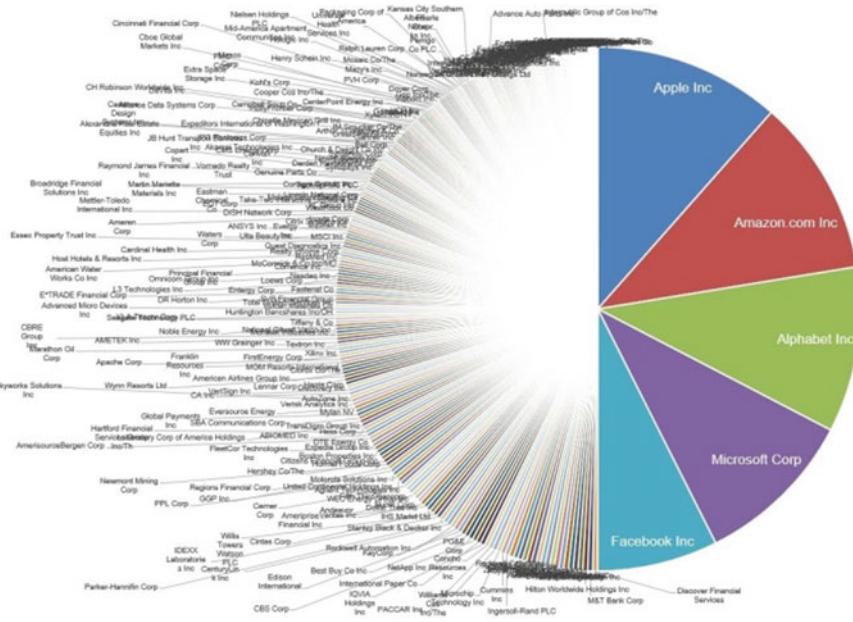


Fig. 1 Big tech giant companies shares worth more than the other companies in term of market capitalization [4]

growth in instruments accurately, robotics is highly used. As an outcome, we are ready to forecast the cost of bitcoin, notwithstanding the portable nature present not just in Bitcoin exchanges (Fig. 1), but in monetary markets as a rule [2].

1.1 Bitcoin Price

A lot of authenticate Bitcoin, while other are pessimist. Regardless the cost of Bitcoin is an issue considered from an money making treasure, software engineering, money related, and mental view meanwhile In the hour of composing this article, the figure seems, by all accounts, to be getting steady. Somehow or another, this may be standard, given that numerous Stockholders are on hold to view connections [3]. There is no improvement found in the scaling issue, actually the underlying push around 2017 is presently on and it is normal for a market to get consistent for certain hours. Be that as it may, given are the characteristics of bitcoin like:

- Duty exempted
- Authentication
- Bondless and unblock against space
- Decentralized Storage
- Trust and secure

- Minor agreement fee
- Unforgeable.

It is reasonable clarification for nations to set up economies and financial frameworks to fortify the monetary structure, while seeking to ingress another component to trust Bitcoin would be contemplated is the quick track improvement in innovation (Fig. 1) which esteems bitcoin by seeing it has characters as a program with the decentralized storage that cannot be hit by banking systems.

1.2 Declination of Bitcoin

The Bitcoin's stockpile is fixed by structure and spoke to by this geometric movement arrangement:

$$S_n = \frac{a(1 - r^n)}{1 - r} = 210,000 \times \frac{50(1 - 0.5)}{1 - 0.5} \approx 21 \times 10^6 \quad (1)$$

This is naturally referenced in Bitcoin's information the constriction of supply, favors the speed at which items like silver, gold are evacuated [5]. It causes numerous to consider Bitcoin as breakdown yet this money is endlessly flimsy alone on the grounds that $1 \text{ Bitcoin} = 10^8 \text{ satoshis}$ [6] (Fig. 2).

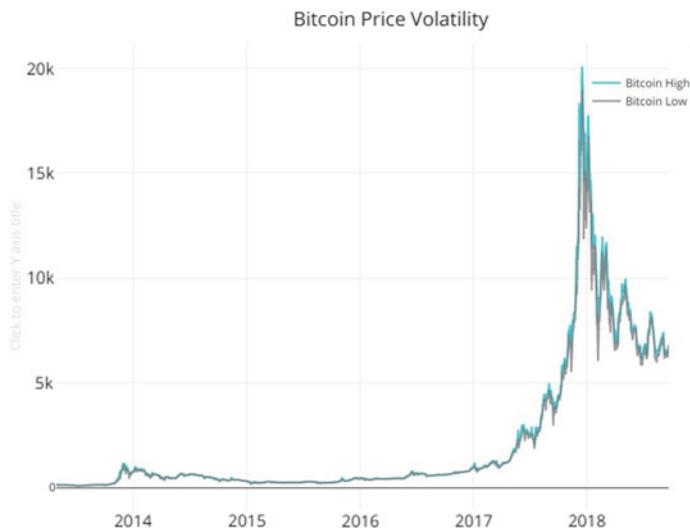


Fig. 2 Peak price shifts of the bitcoin [7]

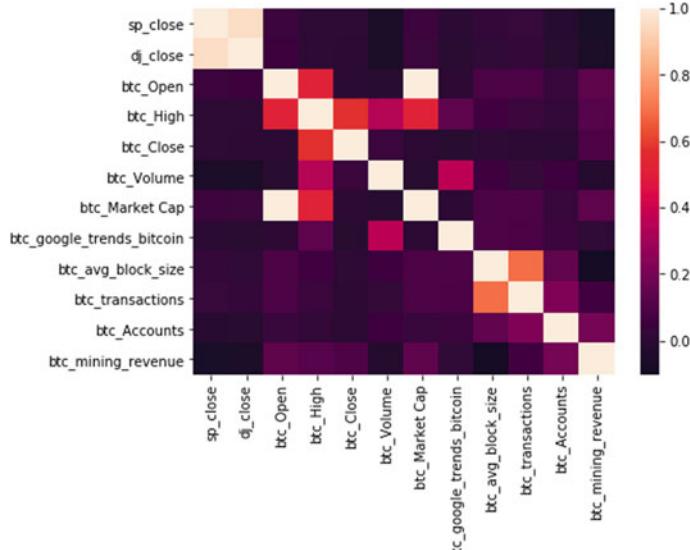


Fig. 3 Depiction of the highest correlation

2 Data Pretreatment

2.1 Data Assembly

Periodic data of all four channels are reviewed since the year 2013. Initially the Bitcoin value history that is pulled back from Coin showcase stopper via its open application programming interface (API) [8]. Secondly, the data from Block chain has been collected, in particular. At that point we consider to pick the medium size of the block, the quantity of client locations, exchanges, and the income of the miners [9]. We consider it token inborn to have Block chain information, gave the steady degree scaling issue, elseway the quantity of the records [10] (Fig. 3).

2.2 Data Normalization

Depending upon the way to normalize the time series data, specifically financial is very difficult. As a principle in the neural networks that first should stack the data which takes the higher values which is heterogeneous [11]. Doing so can activate huge incline update which would prevent network against coincide. To make the learning simpler and quick for the system, data should must follow these qualities:

- Short values consideration—usually, greater part of the values should lie in the range from 0 to 1.
- Homogeneity—values of the features must have approximately same value.

The maximum methods considered for the normalization in the data transformation includes are:

Max-Min Scaling: The data inputs are summarized between 0 and 1 number:

$$Y^J = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (2)$$

Mean Normalization: that form data to have a value lies between -1 and 1 and with a mean of 0 :

$$Y^J = \frac{x - \text{mean}(X)}{\max(X) - \min(X)} \quad (3)$$

Z-Score (Regularity): Point the features are reshuffle with their mean is 0 and where S.D. is 1 :

$$x' = \frac{x - \text{mean}(x)}{\rho} \quad (4)$$

In this scenario, we apply Min-Max Scaling also with adjusting includes on scale between 0 and 1 mentioned that most of the time series data have a peak, hence out of all other we consider the case of Min-Max Scaling because it does a good job.

3 Methodology

In this part this has been described that how to create time series data approval to manage ML problems. The cost prediction is evaluated as fixed fairly than classification and also mentioned the method to use LSTM in these cases. Then we apply hyper parameters.

3.1 Software Used

For the concept of Deep Learning the backend system that we adopt is: Tensorflow and Keras in the frontend layer of constructing neural network. For the tasks linked to the data pandas is used, for the matrix/vector operations NumPy is used and to keep training and test data sets Scikit learn is applied for performing the Min-Max standardization. At last, to display the charts we used Plotly.

3.2 Time Series Set

Generally, time series data is an arrangement of numbers along with the time. Long short term memory is a series to forecast the act of a supervised learning unlike its auto encoder form. As like the comprehensive dataset shall get breakdown into two parts, i.e., input and output. Furthermore, the LSTM is significant in contrast including predefined statistics linear models as it can easily be handled by numerous input estimating problems. In this approach, the LSTM will handle past data to foresee 30 days ahead of final price. Firstly, to determine from what number of prior days one specific forecasting will have admittance to.

3.3 Division of Data into Training and Testing Sets

This part is one of the most critical, solely for this situation of Bitcoin. From the start we wish to foresee the year next, however this infers the data from first Jan 2018 up to Sept 2018 will be considered for testing, the imperfection of this is of course the unexpected contribute 2017, which makes this neural system handle this plan as a last input and anticipating of the year 2020 would not considered be so significant.

3.4 Shifting of the Data into Tensors

LSTM assumes that the data ought to be given as 3D vector in floating values. An essential component of tensors is in their casing that in Python is a line of whole number qualities which fill in as the components of it on 3 pivot. For instance, in the testing dataset of bitcoin the arrangement of the training data set contribution as (1611, 35, 12) presently we are having 1611 examples, time venture of 35 value and 12 parts. In extensive the idea is basic in which we confine the data into blocks of 35 and put these blocks of the data into the NumPy library. In LSTM the input layers are by plan and indicated by the input contention, these 3D input shapes are:

- Sample or Pattern
- Windows Size or Block Size
- Number of Components.

3.5 Long Short Term Memory Meantime

A main component of feed forward Networks, is that they don't hang on the memory in this manner each info is handle severally, with no state being spared between inputs as long as we have a tendency to square measure coping with statistic wherever data

from previous Bitcoin value square measure required, we must always maintain some data to predict the longer term associate degree design contributing this is often the recurrent neural network (RNN) that beside the output features a self-coordinating loop that the window we offer as input gets processed in an exceedingly sequence instead of in an exceedingly single step. However, once the time step (size of window) is massive (which is usually the case) the gradient gets too small/large, that ends up in the development referred to as vanishing/exploding gradient generally [12]. This issue happens while the enhancer back propagates and will make the algorithm run while the loads almost do not change by any means [13, 14]. RNN varieties moderate the issue, in particular LSTM and GRU (Fig. 2) [15].

- Forget gate: $f_t = \sigma(W_f S_t - 1 + W_f S_t)$
- Input gate: $i_t = \sigma(W_i S_t - 1 + W_i S_t)$
- Output gate: $o_t = \sigma(W_o S_t - 1 + W_o S_t)$
- Intermediate Cell State: $\tilde{C} = \tanh(W_c S_t - 1 + W_c X_t)$
- Current Cell status (memory for the next input): $c_t = (i_t * \tilde{C}_t) + (f_t * c_{t-1})$
- Calculating new state: $h_t = o_t * \tanh(c_t)$ (Fig. 4).

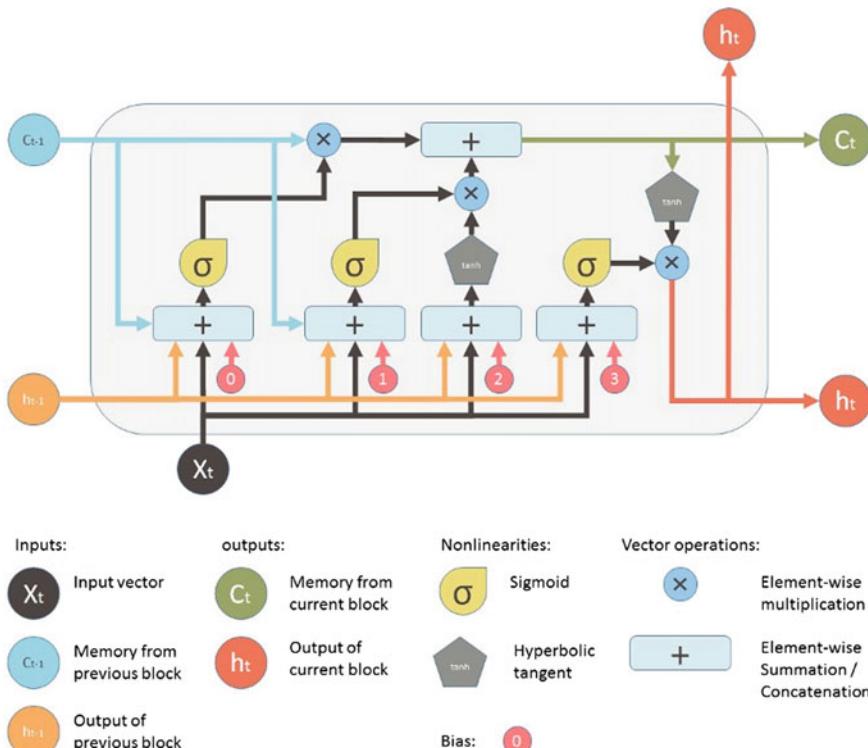


Fig. 4 Flow in the LSTM network

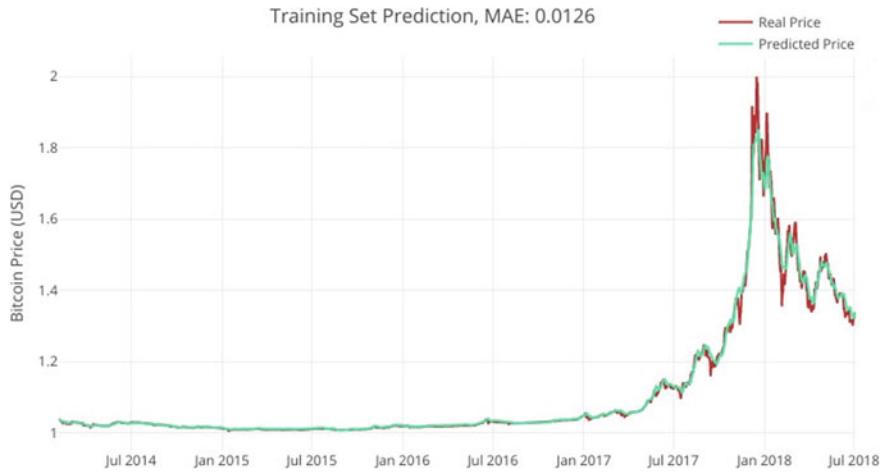


Fig. 5 Training set for prediction

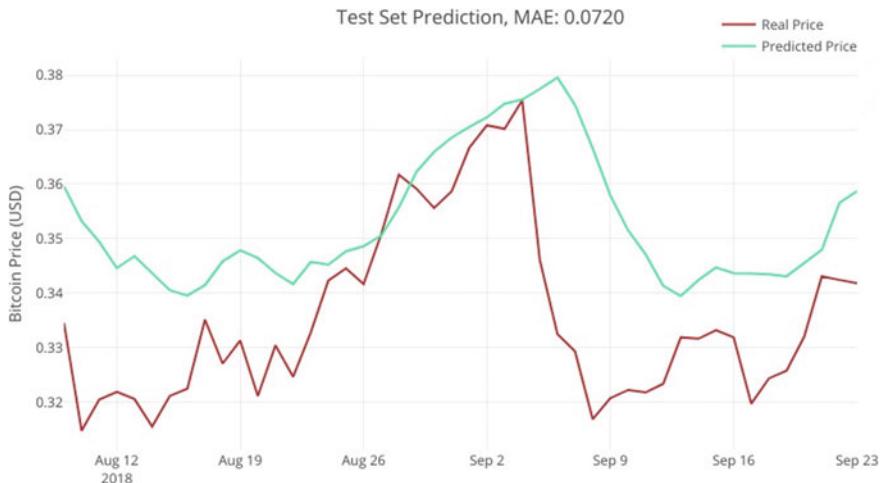


Fig. 6 Prediction on test set and the size of the batch is 100

As it is clearly understand from the equations that every particular gate has distinct set of loads. In the previous mathematical statement, the input gate and transitional cell status are combined with past cell status and the forget gate. Output of the action is then further compiled to count new state. Hence, already stated advanced cell with 4 combining layers rather than exact one tanh layer in RNN prepare LSTM ideal for progression forecasting.

3.6 Hyper Parameters

Optimizer

While Stochastic things are implemented in most of the Neural Network issues, it gets the issue of coincide to a limited minimal. Some other amend are deviations in algorithms, like RMS Prop, Adam and Adagrad. Adam was found to work efficiently better than the rest of the algorithms and hence we proceed with that.

Activation function

The choice for vital role was easy. The most suitable are tanh, ReLu and sigmoid. Sigmoid go through from fading grade, thus almost no significant move from neuron to its corresponding weight, in addition it is not focus everywhere nil, as a outcome the gradient power would be either too high or too low for a number. By comparison, tanh generates the output zero and the exercise is roughly considered to sigmoid. Rectified Linear Unit (ReLu) is highly adopted and as it was created later, it will be excel. To predict the Bitcoin value we chose tanh for better results.

Free Spirit Scale

Regularization is a technique to require the weights in the network. Where as in normal neural networks, l1 and l2 regularization technique is used, in multilayer network, dropping out regularization is used. It randomly set some taking in units to 0 in system to avoid over fitting. Therefore, its rate shows the percentage of confined neurons in the previous layer and lies between 0 and 1. We did experiment with 0.25 and 0.3 and finally we took 0.3.

Total Neurons in hidden layers

We considered for the 10 neurons in hidden layers, it actually figures lot to have extra neurons, as in training process will conclude. Also, when trying with a bigger number did not produce better results.

Span

Rather swiftly, we took for 100 years, after taking different values like 50 or 20. As the total of hidden layer neurons, more ages, the more time it holds for training to finish, so one epoch is the full of iteration under the training data. Further, it cab overfit the model.

Cluster Size

We determined to feed the network, along a cluster of 120 data.

Planning of Network

We consider the Sequential API of Keras, than the sole functional. The architecture is described as:

LSTM Layer: LSTM layer is the essential layer, including all the gates which are mentioned at beginning are already done by Keras. The LSTM criteria are the number of neurons, and the input are discussed above.

Dropout Layer: This is used ahead of the dense layer. As in Keras, a dropout may be added after any hidden layers, in this case it is used after the LSTM layer.

Dense Layer: This is the legitimate perfectly connected layer.

Activation Layer: Here we are resolving fixing issues; the last layer gives the definite union of activations of the previous layers with the weighted vectors. So, the activation is a linear side by side it keeps passed as a parameter to the preceded dense layer.

4 Result and Analysis

In this paper, the consequences of our LSTM model have been introduced. It has been noted during training of the data that the more of the clump size (200) (Figs. 7 and 8) the most exceedingly awful would be the prediction on the testing set. Without a doubt there is no big surprise, as more the training of the data, the more danger of overfitting the model becomes. While it is difficult to foresee the estimation of Bitcoin, it has been seen that attributes are suitable to that calculation; future work incorporates the Gated periodic Unit (GPU) form on RNN, additionally for tuning,

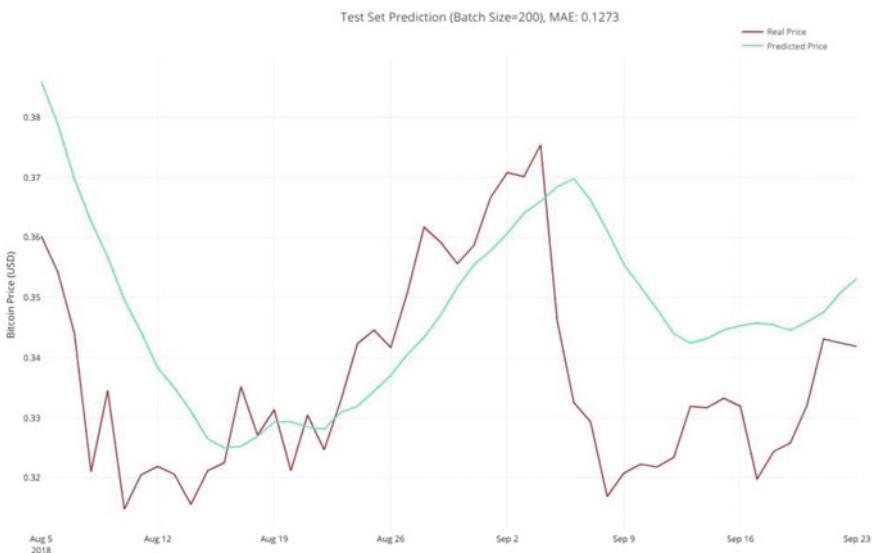


Fig. 7 Prediction on test set, for the 60 days

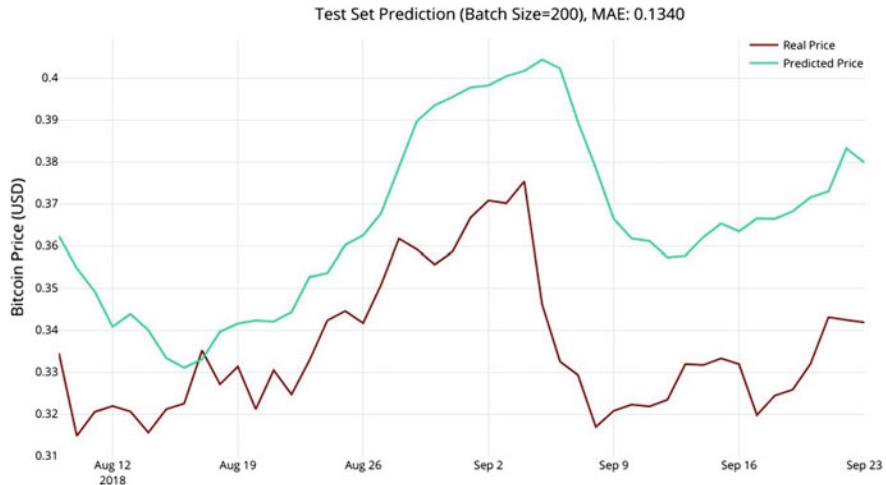


Fig. 8 Prediction on the testing data, having the batch size is of 200 and loss is more in this batch

on prior hyper parameters. The misfortune from the Mean Absolute Error work, when model is utilized for training and testing reason in prediction (Figs. 5, 6 and 9).

5 Conclusions

All in all, evaluating a cost related inconstancy is extreme given the collecting of solidarity affecting available situation. Further, actual cost is to a great extent expanded and determined on future investigation instead of on past data. However, by applying deep neural network has brought us with a clear considerate significance of Bitcoin, and Long short term memory architecture. The development in work includes implementing hyper parameter set, to obtain more accurate network planning. In like manner, more features can be considered notwithstanding from the analysis with Bitcoin, more features are not generally give the better outcomes. Micro monetary elements ought to be remembered for the model for the improved predictions. Anyway, may be data we congregate of Bitcoin, it is stored for years, might have turn into interesting, manufacturing significant explanation in the previous years. An evolution in shared exchanges is open-ended and changing the mode of amount utility. It seems all doubts have not been sorted out, with time it will be clear.

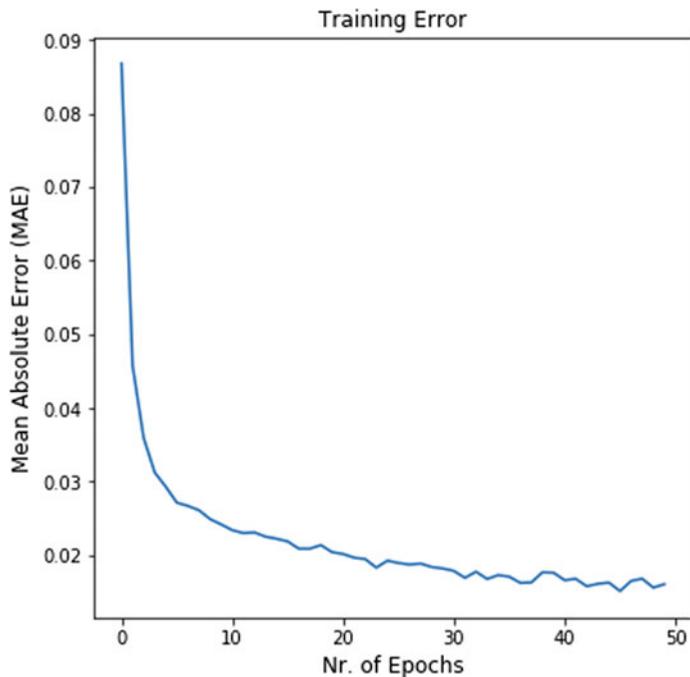


Fig. 9 Loss in training data

References

1. H. Jang, J. Lee, An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *IEEE Access* **6**, 5427–5437 (2018). J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2 (Clarendon, Oxford, 1892), pp. 68–73
2. M. Nakano, A. Takahashi, S. Takahashi, Bitcoin technical trading with artificial neural network (2018)
3. S. Velankar, S. Valecha S. Maji, Bitcoin price prediction using machine learning, in *2018 20th International Conference on Advanced Communication Technology (ICACT)* (IEEE, 2018), pp. 144–147
4. A. Radityo, Q. Munajat I. Budi, Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods, in *2017 International Conference on Advanced Computer Science and Information Systems (ICACIS)* (IEEE, 2017), pp. 433–438
5. S. McNally, J. Roche, S. Caton, Predicting the price of Bitcoin using machine learning, in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (IEEE, 2018), pp. 339–343
6. E. Sin, L. Wang, Bitcoin price prediction using ensembles of neural networks, in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (IEEE, 2017), pp. 666–671
7. Y.B. Kim, J.G. Kim, W. Kim, J.H. Im, T.H. Kim, S.J. Kang, C.H. Kim, Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLoS ONE* **11**(8), e0161197 (2016)

8. R.C. Phillips, D. Gorse, Predicting cryptocurrency price bubbles using social media data and epidemic modelling, in *Computational Intelligence (SSCI)* (IEEE, 2017)
9. S. Gullapalli, Learning to predict cryptocurrency price using artificial neural network models of time series (2018)
10. N.A. Bakar, S. Rosbi, Autoregressive integrated moving average (ARIMA) model for forecasting cryptocurrency exchange rate in high volatility environment: a new insight of bitcoin transaction. *Int. J. Adv. Eng. Res. Sci.* **4**(11), 237311 (2017)
11. L. Lei, Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. *Appl. Soft Comput.* **62**, 923–932 (2018)
12. I. Georgoula, D. Pournarakis, C. Bilanakos, D. Sotiropoulos, G.M. Giaglis, Using time-series and sentiment analysis to detect the determinants of bitcoin prices (2015)
13. A. Greaves, B. Au, Using the bitcoin transaction graph to predict the price of bitcoin (2015)
14. Y. Yoon, G. Swales, Predicting stock price performance: a neural network approach, in *System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on*, vol. 4 (IEEE, 1991), pp. 156–162
15. T. Gao, Y. Chai, Y. Liu, Applying long short term memory neural networks for predicting stock closing price, in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (IEEE, 2017), pp. 575–578

A Comprehensive Analysis for Validation of AVISAR Object-Oriented Testing Tool



Prashant Vats and Manju Mandot

Abstract In this paper, we have attempted to present the procedural implementation and validation of the three-level testing framework of “AVISAR” for effectual and exhaustive testing of the object-oriented softwares for providing an the effective and collective set of test cases that will be used to make ensure maximum accuracy without including any kind of redundancy for a object-oriented software which is being putted under test with ensuring the maximum code coverage.

Keywords Object-oriented testing · Object-oriented software (OOS) · Line of codes (LOC) · Jenetics · Java Parser

1 Introduction

The object-oriented testing of a software code is carried out to discover bugs in it. It is a widely known fact that the comprehensive software testing cannot be performed, i.e., 100% code coverage cannot be attained [1] while testing an OOS as it obliges that each instruction in the code provided and each possible route will be taken at least once to makes sure that each “line of code” (LOC) is being executed through the code coverage procedure, at least once. We devise the test cases with the obligatory pre-conditions, conditions of publication, the inputs and outputs needed to discern the outcome of these test cases sets. If there is any difference in between the prescribed input and the anticipated output, the outcome is observed, to show the presence of bug and for calling the error resolution mechanism. Throughout the test, the reason is to locate the regions with maximum odds of discovering a fault. Because it is difficult to find these regions, our reason should be to develop policies and strategies

P. Vats (✉)

AIMACT, Banasthali University, Vanasthali, Rajasthan, India

e-mail: prashantvats12345@gmail.com

M. Mandot

J.R.N. Rajasthan Vidyapith, Udaipur, Rajasthan, India

e-mail: Manju.mandot@gmail.com

for choosing software testing techniques that are effective. Also, often terms such as fault, mistake, error, and bugs, the terms which are used to describe the testing procedure. The error can be described as any discrepancy that is causing a variance between the expected output and the prescribed input. Also, if there are errors in a software code, software failures will occur. When you run this presence of errors also known as failures, it is called errors, which can cause software failures. In contrast, working with the methodology of object-oriented paradigm [2] has become a difficult job to apply usual test approaches to identify the occurrence of errors in an OOS. OOS offers elements such as polymorphism and encapsulation that supports software reuse. Object-oriented software employs classes and their instances that present a structured and classified work system. Class objects can interact by passing messages and invocations. The usual test method cannot be applied directly to object-oriented software. In the object-oriented methodology, attributes such as encapsulation group data in a single encapsulated unit, which does not only bounds the state of the class objects but it also limits their intermediate test case analysis of the test results to a certain extent. In addition, inheritance in object-oriented software to test a subclass, its inheritance configuration must be divided into a single class. When this is completed, superclass test efforts are not used and, therefore, results in duplicate tests. Also, it becomes fairly tiresome to get a snapshot of a class in OOS, with no creation of additional procedures that show the status of the class. In addition, each new subclass context needs exhaustive regression tests since a procedure can be implemented with polymorphism in a different manner. In this paper, we present the procedural implementation and validation of the three-level testing framework of “AVISAR” for the effectual OO testing of the object-oriented softwares for providing an effective set of test cases sets that will make ensure achieving of maximum accuracy without any extra redundancy for a OOS under test with ensuring the maximum code coverage.

2 The Conceptual Design Schema of the AVISAR Framework

The proposed framework AVISAR is implemented as follows: In the proposed three-tiers testing framework, AVISAR is constructed with Java class libraries using the genetic algorithm. There are three levels in its conceptual schema: 1. Providing input as an object code, 2. Test case generation, 3. Test case prioritization and effort estimation. Figure 1 illustrates the graphical user interface of the AVISAR framework for the examining of object-oriented software. In the AVISAR framework for testing, at its requirement level, a conceptual model generator is fed with the requirements which utilize the object code for the production of the test cases with their objects to widen these test cases more. They are then afterwards used for generation of the test cases for the system under test, as an instantiator of objects by using the Jenetics class of Java. In object-oriented software, the classes communicate with the others by means of the object instantiations and using objects, message passing

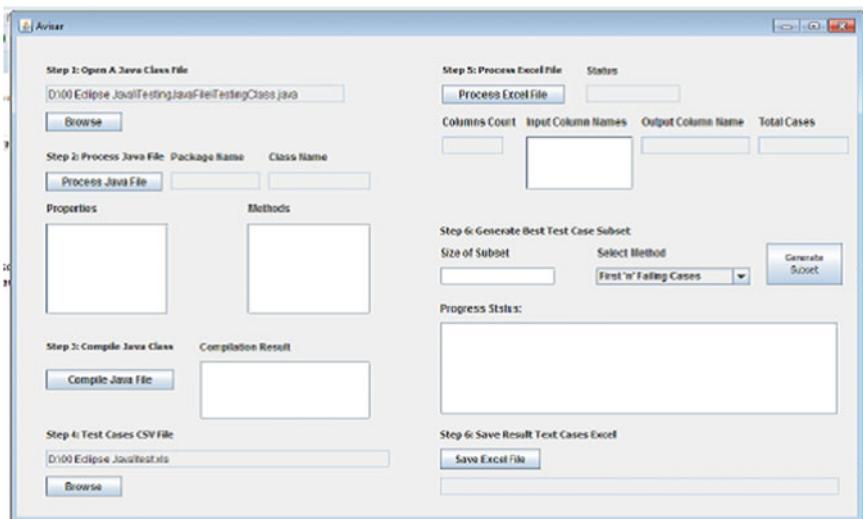


Fig. 1 To show the graphical user interface of the AVISAR framework

with one another. These objects in the object class then are used to capture the events during the execution of a Java code and will then structure the basis of the process of selection of the test cases sets for the classes that are present in object-oriented software. In AVISAR, the test case scenarios are created with the Java Parser classes at the "Test Case Generation and Selection Level." It presents the development of reports of test cases to structure the base of the essential and useful test plans for preparing the required test reports. In addition, to determine its fitness function using the genetic algorithm, this test case sequence can be chosen as a genetic sequence and can be utilized to carry out genetic operations through operators such as crossover, replication, and mutation to discover the effective test case sequence, which guarantees highest coverage of code during execution. On top of it at the "effort estimation level," the genetic algorithm-based selected prioritized sequence of test cases will be provided with a DD path flow-based test technique to find the time complexity for software under test. The decision and lines of codes (LOC) nodes are used to find the estimate of effort to perform the software under test. The use of genetic algorithm for testing path at the effort estimation level makes ensure that the highest code coverage will be attained by means of the effective GA-based prioritized test cases. The AVISAR framework's three levels were implemented using the Java language.

3 The Java Component Libraries Used in the AVISAR Framework

In this section, we are going to explain the Java component classes that we have used to implement the said framework AVISAR.

3.1 Jenetics Library

Jenetics is a well-developed class library based on **evolutionary genetic programming**, written in modern day Java. It is being designed for the various algorithm concepts with a clear separation like gene, g.genotype, fitness function, chromosome, phenotypes, and chromosomal population. Jenetics permits you to minimize or maximize without changing a particular physical function. The Java class library, unlike other implementations for genetic algorithm, to perform the evolution steps uses the evolution *stream* concept (Evolution Stream). Because the Evolution Stream implements the Java Stream interface, with the rest of the Java Stream API, it works smoothly. Following are the classes of Jenetics library that are being used in Avisar:

3.1.1 Factory

As the name suggests, the factory class is used as a generator of new chromosomes based on a particular genotype.

```
{
final Factory<Genotype<IntegerGene>> gtf = Genotype.of(
    IntegerChromosome.of( minIndex, maxIndex, sizeOfSubset )
);
```

In the above code, gtf (Factory Object) is a Java object that acts as a chromosome generator. This particular factory defines the genotype of integer gene using Java generics as shown in Fig. 2.

3.1.2 Genotype

The genotype is the central important class with the genetic algorithm which is working with as shown in Fig. 3. It is the individual gene's structural representation. With one-to-many chromosomes, this class is the encoded problem solution.

```

36@ public List<Integer> startGA() {
37@     int i; // For traversing the genes in the result
38@     IntegerChromosome ich; // To store chromosome of result
39@     IntegerGene ig; // To store gene of chromosome of result
40@     Integer caselnumber; // To store individual case numbers in the gene of chromosome of result
41@ 
42@     // 1. Define the genotype (factory) suitable for the problem
43@     // A factory is just a chromosome generator
44@     // We are here just letting the factory know how to generate the chromosome
45@     //final Factory<Genotype<BitGene>> gtf = Genotype.of( BitChromosome.of(10,0.5) );
46@     final Factory<Genotype<IntegerGene>> gtf = Genotype.of(
47@         //IntegerChromosome.of( minIndex, maxIndex ),
48@         IntegerChromosome.of( minIndex, maxIndex, sizeOfSubset )
49@         /* ,sizeOfSubset */ );
50@ 
51@     // 3. Create the execution environment
52@     //final Engine<BitGene, Integer> engine = Engine.builder(HelloWorld::gval,gtf).build();
53@     final Engine<IntegerGene, Integer> engine = Engine.builder(GAMaxCodeCoverageCases::evol,gtf).build();
54@ 
55@     // 4. Start the execution (evolution) and collect the result
56@     //final Genotype<BitGene> result = engine.stream().limit(100).collect(EvolutionResult.toBestGenotype());
57@     final Genotype<IntegerGene> result = engine.stream().limit(10).collect(EvolutionResult.toBestGenotype());
58@ 
59@     // This may stay the same
60@     System.out.println("Result : \n\t" + result);
61@ 
62@ }
63@ 
```

Fig. 2 To show the use of factory class in Java in AVISAR

```

78@ private static int eval( final Genotype<IntegerGene> gt ) {
79@     //***** Here The Fitness Function Has To Be Implemented *****/
80@     int i; // For Traversing Chromosome
81@     int caselnumber=0; // To save case index
82@     IntegerChromosome ich;
83@     IntegerGene ig;
84@     int failedCases=0;
85@ 
86@     ich = gt.getChromosome().as(IntegerChromosome.class);
87@     for( i=0; i<(ich.length()-1); i++ ) {
88@         ig = ich.getGene(i);
89@         caselNumber = ig.intValue();
90@         if( !checkCase(caselNumber) ) {
91@             failedCases++;
92@         }
93@     }
94@ 
95@     //System.out.println("Cases: " + ich);
96@     //System.out.println("Failed Cases: " + String.valueOf(failedCases));
97@ 
98@     return failedCases;
99@ 
100@     //return gt.getChromosome().as(IntegerChromosome.class).getMax();
101@ }
102@ 
103@ }
104@ 
105@ /**
106@  * private static int eval( final Genotype<BitGene> gt ) {
107@  *     return gt.getChromosome().as(BitChromosome.class).bitCount();
108@  * }
109@ */
110@ private static boolean checkCase( int caselnumber ) {
111@ 
```

Fig. 3 To show the use of genotype class in Java in AVISAR

“Class Genotype<G extends Gene<?,G>>
 java.lang.Object
 org.genetics.Genotype<G>
 All Implemented Interfaces:
 Serializable, Iterable<Chromosome<G>>, Factory<Genotype<G>>, Verifiable
 public final class Genotype<G extends Gene<?,G>>
 extends Object
 implements Factory<Genotype<G>>, Iterable<Chromosome<G>>, Verifiable,
 Serializable

```

42 // 1. Define the genotype (factory) suitable for the problem
43 // A factory is just a chromosome generator
44 // We are here just letting the factory know how to generate the chromosome
45 //final Factory<Genotype<BitGene>> gtf = Genotype.of( BitChromosome.of(10,0.5) );
46 final Factory<Genotype<IntegerGene>> gtf = Genotype.of(
47     IntegerChromosome.of( minIndex, maxIndex ),
48     IntegerChromosome.of( minIndex, maxIndex, sizeOfSubset )
49     /* ,sizeOfSubset */
50 );
51
52 // 3. Create the execution environment
53 //final Engine<BitGene, Integer> engine = Engine.builder(HelloWorld::eval, gtf).build();
54 final Engine<IntegerGene, Integer> engine = Engine.builder(GAMaxCodeCoverageCases::eval, gtf).build();
55
56 // 4. Start the execution (evolution) and collect the result
57 //final Genotype<BitGene> result = engine.stream().limit(100).collect(EvolutionResult.toBestGenotype());
58 final Genotype<IntegerGene> result = engine.stream().limit(10).collect(EvolutionResult.toBestGenotype());
59
60
61 // This may stay the same
62 System.out.println("Result : " + result);
63 System.out.println("Count of failed cases in Result are: " + String.valueOf(eval(result)));
64
65 // generating list of test cases by case number
66 List<Integer> subsetCases = new ArrayList<Integer>();
67 ich = result.getChromosome().as(IntegerChromosome.class);
68 for( int i=0; i<ich.length()-1; i++ ) {
69     ig = ich.getGene(i);
70     caseNumber = ig.intValue();
71     subsetCases.add(caseNumber);
72 }
73
74
75 return subsetCases;
76

```

Fig. 4 To show the use of engine class in Java in AVISAR

3.1.3 Engine

The genetic engine algorithm as shown in Fig. 4, which is the main class, allows decoupling the execution engine configuration. The engine is configured through the Engine Builder class and after creation, cannot be changed. Evolution Stream performs the actual evolution that is created by the engine.

Syntax:

```

io.jenetics.engine
Class Engine<G extends Gene<?,G>,C extends Comparable<? super C>>
java.lang.Object
io.jenetics.engine.Engine<G,C>
All Implemented Interfaces:
EvolutionStreamable<G,C>, Function<EvolutionStart<G,C>,EvolutionResult<G,C>>
EvolutionStreamable<G,C>, Function<EvolutionStart<G,C>,EvolutionResult<G,C>>
public final class Engine<G extends Gene<?,G>,C extends Comparable<? super C>>
extends Object implements Function<EvolutionStart<G,C>,EvolutionResult<G,C>>,
EvolutionStreamable<G,C>

```

3.1.4 Evolution Result

After an evolution step, it represents a state of the genetic algorithm as shown in Fig. 5. It also represents the final evolution of the genetic mutation process state and can be performed and created with using an appropriate gene chromosome collector.

G denotes the genomes type

C denotes the fitness function type.

```

46     //final Factory<Genotype<BitGene>> gtf = Genotype.of( BitChromosome.of(10,0.5) );
47     final Factory<Genotype<IntegerGene>> gtf = Genotype.of(
48         //IntegerChromosome.of( minIndex, maxIndex ),
49         IntegerChromosome.of( minIndex, maxIndex, sizeOfSubset )
50         /* ,sizeOfSubset */ );
51
52
53     // 3. Create the execution environment
54     //final Engine<BitGene, Integer> engine = Engine.builder(HelloWorld::eval,gtf).build();
55     final Engine<IntegerGene, Integer> engine = Engine.builder(GAMaxCodeCoverageCases::eval,gtf).build();
56
57     // 4. Start the execution (evolution) and collect the result
58     //final Genotype<BitGene> result = engine.stream().limit(100).collect(EvolutionResult.toBestGenotype());
59     final Genotype<IntegerGene> result = engine.stream().limit(10).collect(EvolutionResult.toBestGenotype());
60
61
62     // This may stay the same
63     System.out.println("Result : \n" + result);
64     System.out.println( "Count of failed cases in Result are: " + String.valueOf(eval(result)) );
65
66     // generating list of test cases by case number
67     List<Integer> subsetCases = new ArrayList<Integer>();
68     ich = result.getChromosome().as(IntegerChromosome.class);
69     for( int i=0; i<(ich.length()-1); i++ ) {
70         ig = ich.getGene(i);
71         caseNumber = ig.intValue();
72         subsetCases.add(caseNumber);
73     }
74

```

Fig. 5 To show the use of evolution result class in Java in AVISAR

All implemented interfaces:

```

Serializable, Comparable<EvolutionResult<G,C>>
public final class EvolutionResult<G extends Gene<?,G>,C extends Comparable<? super C>>
    extends Object
    implements Comparable<EvolutionResult<G,C>>, Serializable

```

3.1.5 Integer Gene

This is a class based on values, shown in Fig. 6; the use of identity-based sensitive

```

36@ Public List<Integer> startGA() {
37
38     int i; // For traversing the genes in the result
39     IntegerChromosome ich; // To store chromosome of result
40     IntegerGene ig; // To store gene of chromosome of result
41     Integer casenumber; // To store individual case numbers in the gene of chromosome of result
42
43     // 1. Define the genotype (factory) suitable for the problem
44     // A factory is just a chromosome generator
45     // We are here just letting the factory know how to generate the chromosome
46     //final Factory<Genotype<BitGene>> gtf = Genotype.of( BitChromosome.of(10,0.5) );
47     final Factory<Genotype<IntegerGene>> gtf = Genotype.of(
48         //IntegerChromosome.of( minIndex, maxIndex ),
49         IntegerChromosome.of( minIndex, maxIndex, sizeOfSubset )
50         /* ,sizeOfSubset */ );
51
52
53     // 3. Create the execution environment
54     //final Engine<BitGene, Integer> engine = Engine.builder(HelloWorld::eval,gtf).build();
55     final Engine<IntegerGene, Integer> engine = Engine.builder(GAMaxCodeCoverageCases::eval,gtf).build();
56
57     // 4. Start the execution (evolution) and collect the result
58     //final Genotype<BitGene> result = engine.stream().limit(100).collect(EvolutionResult.toBestGenotype());
59     final Genotype<IntegerGene> result = engine.stream().limit(10).collect(EvolutionResult.toBestGenotype());
60
61
62     // This may stay the same
63     System.out.println("Result : \n" + result);
64     System.out.println( "Count of failed cases in Result are: " + String.valueOf(eval(result)) );
65

```

Fig. 6 To show the use of class integer gene

operations kinds for on Integer Gene instances including their hash code identity for their synchronization or reference equality.

```
Class IntegerGene
java.lang.Object
org.jenetics.IntegerGene
All Implemented Interfaces:
Serializable, Comparable<IntegerGene>, BoundedGene<Integer, IntegerGene>,
Gene<Integer, IntegerGene>, NumericGene<Integer, IntegerGene>,
Factory<IntegerGene>, Mean<IntegerGene>, Verifiable
public final class IntegerGene
extends Object
implements NumericGene<Integer, IntegerGene>, Mean<IntegerGene>,
Comparable<IntegerGene>, Serializable
NumericGene implementation which holds a 32 bit integer number.
```

3.2 Integer Chromosome

Integer chromosome is shown in Fig. 7. The values for integer which are included in the range up to codonMax() from codonMin(), will give rise to chromosomes. The GeneticAlg.int implements evolutionary algorithms. The genomeMin() and genomeMax() allow granular range control for individual genetic codons. Firstly, creating an initial genetic chromosomal population using randomly generated chromosomes and cue suggestions. The fitness cost of each chromosome is estimated using the evalFunc() costing function. The algorithm terminates in case termination Cost() is reached by one of the chromosomes; otherwise, to create a new generation, evolutionary operators are applied to the population, including selection, crossover, and mutation. This gene iteration continues until the number of the fitness gene generations exceeds the alterations or the required cost is attained.

```
79@ private static int eval( final Genotype<IntegerGene> gt ) {
80@     //***** Here The Fitness Function Has To Be Implemented *****/
81@     int i; // For Traversing Chromosome
82@     int caseNumber=0; // To save case index
83@     IntegerChromosome ich;
84@     IntegerGene ig;
85@     int failedCases=0;
86@ 
87@     ich = gt.getChromosome().as(IntegerChromosome.class);
88@     for( i=0; i<(ich.length()-1); i++ ) {
89@         ig = ich.getGene(i);
90@         caseNumber = ig.intValue();
91@         if( !checkCase(caseNumber) ) {
92@             failedCases++;
93@         }
94@     }
95@ 
96@     //System.out.println("Cases: " + i);
97@     //System.out.println("Failed Cases: " + String.valueOf(failedCases));
98@ 
99@     return failedCases;
100@ 
101@     //return gt.getChromosome().as(IntegerChromosome.class).getMax();
102@ }
103@ }
104@ }
```

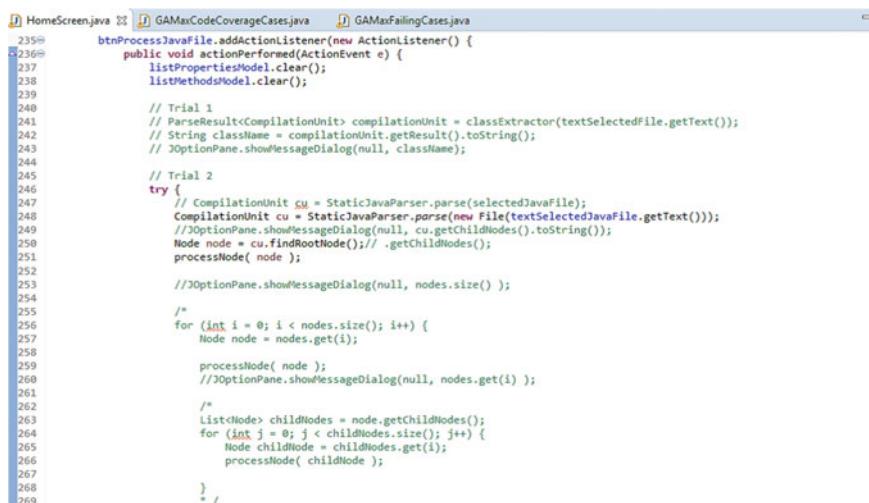
Fig. 7 To show the use of integer chromosomse

The genetic supplied monitorFunc() is recalled at each generation, with a list like GeneticAlg.int, with including its argument being the returning value. The evalFunc must return a numeric value after receiving integer sequences. Finding a chromosome which minimizes this value would be the goal of optimization. To parallelize the evaluation of the cost function, set the plapply() function in a parallel lapply() function, also includes function calls such as mclapply() of the parallel package. Non-data dependency handling functions, like as parLapply() function in Java, the functions, and variables that are being required for evalFunc() to run correctly must be implanted to workable genetic nodes before calling GeneticAlg.int() function.

```
"GeneticAlg.int(genomeLen, codonMin, codonMax,
genomeMin = rep.int(codonMin, genomeLen),
genomeMax = rep.int(codonMax, genomeLen),
suggestions = NULL, popSize = 50,
iterations = 100, terminationCost = NA,
mutationChance = 1/(genomeLen+1), elitism = floor(popSize/10),
geneCrossoverPoints = NULL,
monitorFunc = NULL, evalFunc, allowrepeat = TRUE,
showSettings = FALSE, verbose = FALSE, plapply = lapply)"
```

3.3 Java Parser

Interaction with source code written in Java in a JRE environment as a Java class object representation is allowed by the Java Parser library, shown in Fig. 8. Formally, moreover, we called this Java class object representation as an “Abstract Syntax



```
235@  btnProcessJavafile.addActionListener(new ActionListener() {
236@     public void actionPerformed(ActionEvent e) {
237@         listPropertiesModel.clear();
238@         listMethodsModel.clear();
239@     }
240@     // Trial 1
241@     // ParseResult<CompilationUnit> compilationUnit = classExtractor(textSelectedJavaFile.getText());
242@     // String className = compilationUnit.getResult().toString();
243@     // JOptionPane.showMessageDialog(null, className);
244@     try {
245@         // CompilationUnit cu = StaticJavaParser.parse(selectedJavaFile);
246@         CompilationUnit cu = StaticJavaParser.parse(new File(textSelectedJavaFile.getText()));
247@         // JOptionPane.showMessageDialog(null, cu.getChildNodes().toString());
248@         Node node = cu.findRootNode()// .getChildNodes();
249@         processNode( node );
250@         //JOptionPane.showMessageDialog(null, nodes.size());
251@         /*
252@          for (int i = 0; i < nodes.size(); i++) {
253@              Node node = nodes.get(i);
254@              processNode( node );
255@              //JOptionPane.showMessageDialog(null, nodes.get(i));
256@              /*
257@               List<Node> childNodes = node.getChildNodes();
258@               for (int j = 0; j < childNodes.size(); j++) {
259@                   Node childNode = childNodes.get(j);
260@                   processNode( childNode );
261@               }
262@           */
263@         }
264@     }
265@     */
266@   }
267@ }
```

Fig. 8 To show the use of Java Parser in the AVISAR framework

Tree” (AST). Additionally, it provides a convenience mechanism to navigate the tree with what we have termed as Visitor Support. This gives developers the ability to focus in their source on identifying interesting patterns, rather than the laborious writing of tree traversal code. The final principal feature of the library is the ability to manipulate the source code underlying structure. This can then be written to a file, providing developers with the facility to build their own code generating software.

3.4 Jacoco Code Coverage Tool

A software metric called code coverage is used during automated tests to measure the number of lines of our code executed that runs the test cases using JUnit Java class library which will initiate the JaCoCo agent to act, so they will generate code coverage report written in a binary format in the targeted destination directory—target/jacoco.exec. In our pom.xml file, we need to declare this maven plugin to get up and running with JaCoCo:

```
<<plugin> <groupId>org.jacoco</groupId>
<artifactId>jacoco-maven-plugin</artifactId>
<version>0.7.7.201606060606</version>
<executions> <execution> <goals> <goal>prepare-agent</goal>
</goals> </execution> <execution> <id>report</id>
<phase>prepare-package</phase> <goals> <goal>report</goal> </goals>
</execution> </executions> </plugin>
```

The following Java class link as provided below will guide you to the latest version of the Java plugin classes in the central repository maven Java class libraries.

```
public boolean isPalindrome(String inputString) {
    if (inputString.length() == 0) return true;
    Else
        { char firstChar = inputString.charAt(0);
        char lastChar = inputString.charAt(inputString.length() - 1);
        String mid = inputString.substring(1, inputString.length() - 1);
        return (firstChar == lastChar) && isPalindrome(mid); } }
All we need now is a simple JUnit test:
{ public void whenEmptyString_thenAccept()
    Palindrome palindromeTester = new Palindrome();
    assertTrue(palindromeTester.isPalindrome("")); }
```

4 Validation of the Proposed Tool AVISAR

In this section of the paper, we have compared the proposed tool AVISAR with some existing tools of testing of object-oriented programs and the comparison has been given in Table 1.

Table 1 Comparison of AVISAR OOT tool with existing OOT tools

| Criteria | AVISAR | ASTOOT | ERASER | TOTEM | ARTOO | EVACON |
|------------------------------|---|---|---|---|--|--|
| Proposed by authors | Gossain et al. [3] | Doong and Frank [4] | Savage et al. [5] | Lionel and Yvan [6] | Ilinca et al. [7] | Inkumsah and Xie [8] |
| Year of release | 2020 | 1991 | 1997 | 2002 | 2008 | 2007 |
| Code used for implementation | Jenetics java library, JCOCO, Java Parser | C++ | Java | JSysDG (Java) In C# | EiffelBase library <i>(iCUTE)</i> | Java concolic Unit Testing Engine |
| OS support | Windows, Linux | Windows | Windows XP | All Windows | Windows, Mac, Linux | Windows, Linux |
| Platform | Cross platform GUI based | Cross platform code | GUI | GUI | Cross platform code based | Cross platform GUI based |
| Purpose | Providing input as object code, “Test case Generation,” and then “Test Case Prioritization” & “Effort estimation” | To generate test cases that capture both class state values and state transitions | For dynamically detecting data races in lock-based multithreaded programs | To support the derivation of functional system test requirements, which will be transformed into test cases, test oracles, and test drivers | For evaluating the efficiency of detection of real-time faults in real-time software | Achievement of high structural coverage like branch coverage of object-oriented programs |
| Testing level | Integrated system level | Class level | Class level | System level | Object level | Class level |
| Type of testing approach | Structural testing | Aspect-based orientation testing | DD flow path-based testing | UML-based testing technique | Adaptive random testing | Structural testing |

5 Conclusion

In this paper, we have given the details about a proposed tool AVISAR for the testing of object-oriented Programs. Further, we have discussed about the technical details of our proposed OOT tool AVISAR and have validated the tool with the existing tools. After the comparison, we have found that AVSIAR provides better optimized results for the test case selection and prioritization in comparison to the other tools.

References

1. K.K. Aggarwal, Y. Singh, *Software Engineering*, 2nd ed. (New Age International (P) Ltd, India, 2006)
2. I. Jacobson, M. Christerson, in *Object Oriented Software Engineering—A Use Case Driven Approach* (Person Education, Upper Saddle River, NJ, 2009)
3. A. Gossain et al., AVISAR—a three tier architectural framework for the testing of object oriented programs, in *IEEE Second International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH)* (2016)
4. R.K. Doong, P.G. Frankl, The ASTOOT approach to testing object oriented programs. *ACM Trans. Softw. Eng. Methodol.* **3**, 101–130 (1994)
5. S. Savage, M. Burrows, G. Nelson, P. Sobalvarro, T. Anderson, Eraser a dynamic data race detector for multithreaded programs. *ACM Trans Comput. Syst.* **15** (1997)
6. B. Lionel, L. Yvan, A UML-based approach to system testing, in *Proceedings of Carleton University TR SCE-01-01-Version 4* (2002)
7. C. Ilina, L. Andreas, O. Manuel, M. Bertrand, ARTOO: Adaptive Random Testing for Object-oriented Software, in *ICSE'08*, Leipzig, Germany, 10–18 May 2008
8. K. Inkumsah, T. Xie, Evacon a framework for integrating evolutionary and concolic testing for object-oriented programs, in *ASE'07—2007 ACM/IEEE International Conference on Automated Software Engineering* (2007), pp. 425–428

Author Index

A

- Aakaash, N., 299
Abdul Rajak, A. R., 209
Abin, Deepa, 95
Agarwal, Khyati, 179
Akshaya Bala, K., 299
Alla, Kalavathi, 503
Anitha, H. M., 473
Antonijevic, Milos, 163
Arora, Yojna, 589, 599
Arya, Meenakshi S., 299

B

- Bacanin, Nebojsa, 163
Baig, S. Q., 143
Bane, Anushree, 541
Bezdan, Timea, 163
Bhat, Meghana R., 77
Bhattacharjee, Krishnanjan, 491
Bhattacharya, Debayan, 335
Bhowal, Suvam, 229
Bidwai, Sandeep, 289
Biswas, Adeep, 335
Busygyn, Volodymyr, 33

C

- Chandrashekhar, Pantina, 269
Chatterjee, Sujoy, 385
Chaudhary, Mahima, 363
Chaurasia, Manish Kumar, 43
Choubey, Suyash, 353

D

- Deivalakshmi, S., 237
Desai, Pranav, 579
Deshpande, Sachin, 571
Devi, Gayatri, 143
Dharkar, Piyush, 491
Dhas, Neha, 95
Dhuru, Saloni, 561
Dutta, Abhishek, 1
Dutt, Varun, 179

E

- Erranki, Kiran L. N., 245

G

- Ganatra, Nilay, 515
Ganesan, Karthik, 321
Giri, Nupur, 541
Gopalakrishnan, Anilkumar Kothalil, 107
Gosar, Ankur P., 65
Gowri, S., 229
Gupta, Aman, 551
Gupta, Deepak, 443
Gupta, Hardik, 589
Gutjahr, Georg, 269, 353
Gwal, Ashok Kumar, 217

H

- Haridas, Mithun, 269
Hulina, Iryna, 33

J

- Jabez, J., 229
 Jadhav, Pramod P., 277
 Jain, Anjana, 363
 Jain, Isha, 43
 Jain, Neeraj, 1
 Jibukumar, M. G., 119
 Jindal, Rajni, 43
 Joshi, Shashank D., 277

K

- Kadam, Rishikesh, 541
 Kamble, Soham, 433
 Karwande, Atharva, 433
 Katre, Paritosh, 491
 Kawathekar, Seema S., 375
 Kesav, Nivea, 119
 Kolhe, Tejas, 433
 Kothawade, Rushikesh, 491
 Krishna, Sai, 179
 Kumar, Ajai, 491
 Kumar, Biswajit, 321
 Kumari, Kamlesh, 405
 Kumari, Neha, 395
 Kumar, Rajeev, 395
 Kurtadikar, Vidya Sujit, 153

L

- Le, Chi Quynh, 191
 Li, Danyang, 51
 Li, Xinlai, 51

M

- Magar, Rushikesh, 433
 Maheshwari, Laksh, 433
 Makkar, Priyanka, 589, 599
 Malusare, Aarati, 133
 Mandot, Manju, 613
 Martynenko, Andrii, 33
 Mayannavar, Shilpa, 289
 Mehta, Raj Paresh, 23
 Mehta, Swati, 491
 Mishra, Dillip K., 143
 Mishra, Kirti, 87
 Mishra, Kriti, 87
 Modi, Maulik A., 203
 Moroz, Boris, 33
 Moroz, Dmytro, 33
 Mukhopadhyay, Debabjyoti, 87, 133

N

- Nagar, Aishwariya Rao, 77
 Nagwani, Naresh Kumar, 1
 Nain, Himanshu, 551
 Nair, Gowri M., 269
 Nandanwar, Anagha, 133
 Nandhana, K., 237
 Nandhini, S., 237
 Nanty, Simon, 385
 Nedungadi, Prema, 269, 353
 Nguyen, Le Cuong, 191
 Nikam, Kalpesh, 95

P

- Paliwal, Shaiyya, 173
 Panda, Rama Ranjan, 1
 Pande, Himangi Milind, 153
 Pandey, Alpana, 217
 Panjwani, Anmol A., 483
 Pasquier, Nicolas, 385
 Patel, Atul, 515, 529
 Patel, Sanskruti, 529
 Patel, Tushar M., 203
 Patil, Akhilesh P., 321
 Patil, Aseem, 11
 Patil, Nilesh M., 483
 Pawar, Reeta, 217
 Phalnikar, Rashmi, 257
 Pillai, Neil, 491
 Pooja, Gopu, 1
 Poreddy, Yaswanth Reddy, 245
 Pranathi, Veerabrahmam, 299
 Preetha, S., 451, 473
 Pronika, 423

Q

- Qadri, Syed Aarif Suhaib, 229

R

- Raghunath, Aparna, 209
 Rahate, Ganesh, 343
 Rai, Anurag S. D., 217
 Rajeshwari, C. S., 217
 Rajeshwari, K., 77, 473
 Rajkumar, S., 335
 Ramachandran, Vedantham, 503
 Rana, Sanjeev, 405
 Rane, Tanmay P., 483
 Rani, Peddireddy Janaki, 503
 Rao, Raghvendra, 353
 Rath, Arabinda, 143

S

- Sakshi, Shriya, 133
Sanghvi, Meet Ashok, 23
Sanivarapu, Prasanth Vaidya, 313
Sasane, Sanskruti, 95
Saxena, Isha, 43
Segu, SaiTeja, 245
Selvanambi, Ramani, 335
Sethi, Kartik, 321
Shah, Aneri, 463
Shah, Bela, 463
Shah, Darshin Kalpesh, 23
Shah, Prasham Sanjay, 23
Shah, Purvesh, 463
Shah, Smeet, 463
Shanthana Lakshmi, S., 443
Sharma, Nitin, 589, 599
Sharma, Shekhar, 363
Sharma, Simran, 571
Sharma, Vikas, 589
Sheeba, 65
Sheela, S. V., 451
Shidaganti, Ganeshayya, 65
ShivaKarthik, S., 491
Shrivastav, Avinash, 561
Shvachych, Gennady, 33
Singh, Artika, 23
Sneha Priya, K., 77
Sovitkar, Sarika Ashok, 375
Suganya, R., 237
Sunagar, Pramod, 321
Suratwala, Darshit, 343

T

- Talele, Pratvina, 257

- Thangavel, Senthil Kumar, 237
Tiwari, Laxmi, 87
Tran, Vu Kien, 191
Tyagi, S. S., 423

U

- Uniyal, Prakhar, 179

V

- Valecha, Bhavika, 541
Valecha, Pulkit, 173
Varsha, S., 237
Vats, Prashant, 613
Verma, Devika, 491
Vidhani, Vishakha, 541
Virendrasingh, Suryavanshi, 179

W

- Wali, Uday V., 289
Wani, Tejas, 95
Wyawahare, Medha, 433

Y

- Yadav, R. K., 173

Z

- Zivkovic, Miodrag, 163
Zivkovic, Tamara, 163
Zuluaga, Maria A., 385