

Algorithms for Intelligent Systems

Series Editors: Jagdish Chand Bansal · Kusum Deep · Atulya K. Nagar

Hari Vasudevan
Antonis Michalas
Narendra Shekokar
Meera Narvekar *Editors*

Advanced Computing Technologies and Applications

Proceedings of 2nd International
Conference on Advanced Computing
Technologies and Applications—
ICACTA 2020



Springer

Algorithms for Intelligent Systems

Series Editors

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

Kusum Deep, Department of Mathematics, Indian Institute of Technology Roorkee,
Roorkee, Uttarakhand, India

Atulya K. Nagar, Department of Mathematics and Computer Science,
Liverpool Hope University, Liverpool, UK

This book series publishes research on the analysis and development of algorithms for intelligent systems with their applications to various real world problems. It covers research related to autonomous agents, multi-agent systems, behavioral modeling, reinforcement learning, game theory, mechanism design, machine learning, meta-heuristic search, optimization, planning and scheduling, artificial neural networks, evolutionary computation, swarm intelligence and other algorithms for intelligent systems.

The book series includes recent advancements, modification and applications of the artificial neural networks, evolutionary computation, swarm intelligence, artificial immune systems, fuzzy system, autonomous and multi agent systems, machine learning and other intelligent systems related areas. The material will be beneficial for the graduate students, post-graduate students as well as the researchers who want a broader view of advances in algorithms for intelligent systems. The contents will also be useful to the researchers from other fields who have no knowledge of the power of intelligent systems, e.g. the researchers in the field of bioinformatics, biochemists, mechanical and chemical engineers, economists, musicians and medical practitioners.

The series publishes monographs, edited volumes, advanced textbooks and selected proceedings.

More information about this series at <http://www.springer.com/series/16171>

Hari Vasudevan · Antonis Michalas ·
Narendra Shekokar · Meera Narvekar
Editors

Advanced Computing Technologies and Applications

Proceedings of 2nd International Conference
on Advanced Computing Technologies
and Applications—ICACTA 2020



Springer

Editors

Hari Vasudevan
Dwarkadas Jivanlal Sanghvi
College of Engineering
Mumbai, Maharashtra, India

Narendra Shekokar
Department of Computer Engineering
Dwarkadas Jivanlal Sanghvi
College of Engineering
Mumbai, Maharashtra, India

Antonis Michalas
Department of Computing Sciences
Tampere University of Technology
Tampere, Finland

Meera Narvekar
Department of Computer Engineering
Dwarkadas Jivanlal Sanghvi
College of Engineering
Mumbai, Maharashtra, India

ISSN 2524-7565

ISSN 2524-7573 (electronic)

Algorithms for Intelligent Systems

ISBN 978-981-15-3241-2

ISBN 978-981-15-3242-9 (eBook)

<https://doi.org/10.1007/978-981-15-3242-9>

© Springer Nature Singapore Pte Ltd. 2020

This work is subject to copyright. All rights are reserved 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, express 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

Preface

It is our pleasure to welcome you to the 2nd International Conference on Advanced Computing Technologies and Applications 2020 (DJ ICACTA 2020) in Mumbai, India. The purpose of the International Conference on Advanced Computing Technologies and Applications has been to create a meeting point of researchers, engineers and practitioners, who can address new challenges in the area of intelligent computing and to exchange and share their experiences and research results regarding various aspects of linguistic computing, data computing, statistical computing and ambient applications.

DJ ICACTA 2020 promises to be both stimulating and informative with the involvement of a broad range of keynote and invited speakers from industry as well as academia. The programme consists of invited sessions and discussions with eminent speakers, covering a wide range of topics in the domain of artificial intelligence.

DJ ICACTA 2020 received more than 160 paper submissions from all over the world, out of which 42% were selected. This was done keeping in mind the intention of preserving a high-quality standard for the next editions of this conference.

We would like to express our thanks to all participants. First of all, we thank the authors, whose quality work has become the essence and foundation of this conference. We also thank all the members of the programme committee and the reviewers for their expert reviews and contributions. We must deeply thank the invited speakers for their excellent contribution in sharing their valuable knowledge. Finally, special thanks to all the members of the DJ ICACTA 2020 team, whose collaboration has been the fundamental for the success of this conference.

We hope you will have a technically rewarding experience during the conference, and you will use this occasion to meet old friends and make many new ones. We wish you all the very best and hope you have an unforgettable stay at Mumbai, India.

Mumbai, India

Dr. Hari Vasudevan
Dr. Narendra Shekokar
Dr. Meera Narvekar

About DJSCE

Shri Vile Parle Kelavani Mandal's (SVKM) Dwarkadas Jivanlal Sanghvi College of Engineering (DJSCE) was established in the year 1994. In a span of 25 years, DJSCE has come a long way and has made its impact felt not only in the country, but also abroad. **DJSCE is an Autonomous Institution, affiliated to the University of Mumbai. The college has been granted autonomy by the University Grants Commission (UGC), New Delhi, for a period of 10 years, starting from the AY 2019–2020 till 2028–2029.** The college offers eight undergraduate programmes, three postgraduate programmes and three Ph.D. courses, affiliated to the University of Mumbai. All the undergraduate programmes are accredited by the National Board of Accreditation. Students of the college have been performing exceedingly well in national and globally competent multinational companies and also in the universities in India and abroad as they pursue their higher education. The favourable location of the institute in the heart of Mumbai, along with state-of-the-art facilities and a distinguished faculty, has been a nurturing ground for students of high academic capabilities. Continued efforts of the parent trust, the faculty and the students have always propelled the college into the top echelons of quality engineering institution, as the college is rated amongst the best in the country.

About IACTA

ICACTA 2020 is the 2nd International Conference on Advanced Computing Technologies and Applications, organized by SVKM's Dwarkadas Jivanlal Sanghvi College of Engineering (DJSCE). It was conducted on the 28th and 29th of February 2020. The theme of the conference was “Intelligent Computing”. The primary objective of ICACTA 2020 was to provide a platform for researchers, academicians and industry professionals from all over the world to present their research and development work in the area of intelligent computing. The conference was organized to help the IT industry as well as to derive the advances for

next-generation computing. The conference aimed to provide participants with an opportunity to exchange new ideas in their research work, so as to establish business or research relations and to find global partners for future collaboration. ICACTA 2020 focused on recent advancements in the domain of intelligent computing, such as linguistic computing, statistical computing, data computing and ambient applications.

ICACTA 2020

Patrons

Shri Amrish R. Patel, Chief Patron, President, SVKM

Shri Bhupesh R. Patel, Joint President, SVKM

Shri Bharat Sanghvi, Vice-President, Trustee, SVKM & I/C, DJSCE

Shri Chintan A. Patel, Vice-President, SVKM

Shri Sunandan R. Divatia, Hon. Secretary, SVKM

Shri Harshad H. Shah, Hon. Treasurer, SVKM

Shri Jayant P. Gandhi, Hon. Joint Secretary, SVKM

Shri Shalin S. Divatia, Hon. Joint Secretary, SVKM

Shri Harit H. Chitalia, Hon. Joint Treasurer, SVKM

Shri Jagdish Parikh, Hon. Joint Treasurer, SVKM

International Advisory Committee

Dr. Antonis Michalas, Assistant Professor, Tampere University of Technology, Finland

Dr. William Grosky, Professor, Department of Computer & Information Science, Western Michigan University, Michigan

Dr. Yuri Borissov, Associate Professor, Institute of Mathematics & Informatics, Bulgarian Academy of Science, Bulgaria

Mr. Manoj Bubna, COO, Nvizion Solutions Inc, Mumbai, India | Chicago, US
Co-Founder, Nitrogen Limited, Swindon, UK

Dr. Marina Gavrilova, Professor & Head of Department of Computer Science, University of Calgary, Canada

Dr. Ching-Hsien Hsu, Professor, Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan

Dr. Peter Mueller, Research Staff Member, IBM Research Laboratory, Switzerland
Dr. Luminita Moraru, Professor, Department of Chemistry, Physics & Environment at Dunarea de Jos, University of Galati, Romania

Dr. Ngoc Thanh Nguyen, Professor & Head of Department of Information, System, Wroclaw University of Science & Technology, Poland

Dr. Neeli Prasad, Professor, ITU, California

Mr. Arnab Chakraborty, Global Managing Director, Accenture, California

Mr. Prashant Kothari, Project Manager, CERN, Switzerland

Dr. Rajendra Akerkar, Professor, Western Norway Research Institute, VestlandsforskningSogndal, Norway

Mr. Soujanya Bhumkar, Co-founder, CEO & Director, Cooliris Inc, San Francisco, California

Mr. Ankit Sharma, Head of Application Support, IT at Eastspring Investments, Singapore

Mr. Sandeep Shah, Founder Chairman & CEO, Skyscape, Massachusetts

Dr. Sanjiv Bhatia, Professor & Graduate Director, Computer Science, University of Missouri, St. Louis

Dr. Ajay Gupta, Professor, Department of Computer Science, Western Michigan University, Michigan

National Advisory Committee

Dr. Debajyoti Mukhopadhyay, Director, NHITM, Mumbai

Dr. Surya Durbha, Professor, Center of Studies in Resources Engineering, IIT Bombay

Dr. Supratim Biswas, Professor, CSE Department, IIT Bombay

Dr. S. P. Dattagupta, Professor, Electrical Department, IIT Bombay

Dr. D. Datta, Head, of Computational Radiation Physics, Section of Health Physics Division, Homi Bhabha, National Institute (BARC), Mumbai

Mr. Atul Gandre, Global Technology Head, Tata Consultancy Services, Mumbai

Mr. Anoop Kumar, Vice-President & Head, Business Excellence, Infosys, Bengaluru

Mr. Saurabh Srivastava, Deputy General Manager, HCL Technology, Bengaluru

Dr. M. Sasikumar, Director, CDAC, Mumbai

Dr. Suresh Ukarande, Associate Dean, Faculty of Science & Technology, University of Mumbai

Mr. Tumul Prasad Srivastava, Operations Head, Tech Mahindra, Pune

Mr. Lalit Sharma, Vice-President, Credit Suisse, Pune

Mr. Lalit Sharma, Vice-President, Credit Suisse, Pune

Dr. Sathyababu Korra, Assistant Professor, NIT Rourkela

Dr. Suraj Sharma, Assistant Professor, IIT Bhubaneswar

Dr. Rajesh Ingle, Dean, Professor & Head, Department of Computer Engineering, PICT, Pune

Dr. U. D. Kolekar, Principal, AP Shah Institute of Technology, Thane

Dr. J. W. Bakal, Principal, Shivajirao S. Jondhale College of Engineering, University of Mumbai

Organizing Committee

Dr. Hari Vasudevan, General Chair, Principal, DJSCE

Dr. A. C. Daptardar, General Co-Chair, Vice-Principal (Admin.), DJSCE

Dr. Manali J. Godse, General Co-Chair, Vice-Principal (Acad.), DJSCE

Dr. Narendra M. Shekokar, Conference Chair, Professor, Computer Engineering, DJSCE

Dr. Meera Narvekar, Conference Chair, Head, Computer Engg., DJSCE

Dr. Ramchandra Mangrulkar, Technical Chair

Dr. Vinaya Sawant, Organizing Chair, Head, Information Tech, DJSCE

Members

Prof. (Dr.) Abhijit Joshi

Prof. Aruna Gawade

Prof. Kiran Bhowmick

Prof. Kriti Srivastava

Prof. Purva Raut

Prof. Khushali Deulkar

Prof. Lakshmi Kurup

Prof. Neha Katre

Prof. Harshal Dalvi

Prof. Ashok Patade

Prof. Arjun Jaiswal

Prof. Chetashri Bhadane

Prof. Sindhu Nair

Prof. Ruhina Karani

Prof. Anusha Vege

Prof. Stevina Correia

Prof. Mitchell D'silva

Prof. Lynette D'mello

Prof. Pranit Bari

Prof. Deepika Dongre

Prof. Sudhir Bagul

Prof. Pratik Kanani

Prof. Pankaj Sonawane

Prof. Sonali Jadhav

Prof. Neha Kesho Ram

Prof. Prachi Tawde

Prof. Aniket Kore

Contents

1	Career Counselling Chatbot Using Cognitive Science and Artificial Intelligence	1
	Godson D'Silva, Megh Jani, Vipul Jadhav, Amit Bhoir and Prithvi Amin	
2	Voice Assistant for Ubuntu Implementation Using Deep Neural Network	11
	Shruti Singh, Harshita Arya and P. Arun Kumar	
3	Trust Attacks in Internet of Things: A New Data-Centric Cybercrime on Enterprise Use Case	21
	Parikshit N. Mahalle and Gitanjali R. Shinde	
4	Analysis of Light Pollution Prediction Using Mathematical Model and Machine Learning Techniques	31
	Aastha Sainger, Rishikesh Yadav, Pradnya Tipare, Samidha Waghralkar, Vimla Jethani and Amit Barve	
5	Design and Implementation of Library Shelf Management (LiBOT) Using Machine Learning	45
	Anish Pandita, Mit Parekh, Jitendra Sachwani, Romit Shah and Ramchandra Mangrulkar	
6	IndoorNet: Generating Indoor Layouts from a Single Panorama Image.....	57
	Yash Kotadia, Krisha Mehta, Mihir Manjrekar and Ruhina Karani	
7	Lightweight Random Number Generation for Elliptic Curve Cryptography for Use in IoT	67
	Aruna Gawade and Rushabh Vinchhi	

8 Improvement of Lightweight Integrity Verification Algorithm Using TDHA	75
Anushka Gangwal, Dhyey Mehta, Soham Khedekar and Aruna Gawde	
9 Catchment Area Detection and Optimization	85
Richard Joseph, Sanket Gokhale, Akash Hasamnis, Grishma Gurbani and Rishil Kirtikar	
10 Automation in Healthcare Using IoT and Cryptographic Encryption Against DOS and MIM Attacks	97
Prajakta Kamble and Aruna Gawade	
11 Intelligent System to Diagnose LBP Using Genetic Algorithm and Support Vector Machine	107
Mittal Bhatt and Vishal Dahiya	
12 Area Analysis for Dengue Prediction	117
Aniket Milind Banginwar, Shreyas Nanaware, Kalpita Bhagat and Deepshikha Chaturvedi	
13 Crowdsourcing for Urban Laborers and Time Optimization	127
Nihit Natu, Ayush Gupta, Viraj Mahadik and Amiya Kumar Tripathy	
14 Implementation of Residual Network (ResNet) for Devanagari Handwritten Character Recognition	137
Mandar Mhapsekar, Prathamesh Mhapsekar, Aniket Mhatre and Vinaya Sawant	
15 An Efficient E-Commerce Design by Implementing a Novel Data Mapper for Polyglot Persistence	149
Kishan Trivedi, Sambhav Shah and Kriti Srivastava	
16 Improving Extreme Learning Machine Algorithm Through Optimization Technique	157
Nilesh Rathod and Sunil B. Wankhade	
17 Intrusion Detection System Against Malign Packets—A Comparative Study Between Autoencoder and Ensemble Model.....	165
Adit Sadiwala, Kishan Rathore, Yash Shah, Harsh Shah and Kriti Srivastava	
18 Chest Pathology Detection Using Medical Imaging	177
Devansh Shah, Purav Nisar and Pankaj Sonawane	
19 Disorder Detection in Tomato Plant Using Deep Learning	187
Saiqa Khan and Meera Narvekar	

20 Improving Security of IoT Networks Using Machine Learning-Based Intrusion Detection System	199
Smita Sanjay Ambarkar and Narendra M. Shekokar	
21 Sapling Health Monitoring System	211
Shubham Mishra, Nishanth Shastry and Tarun Tiwari	
22 Proposed Infrastructure for Census Enumeration and Internet Voting Application in Digital India with Multichain Blockchain	223
Vivek Tirodkar and Sonali Patil	
23 A Cost-Efficient and Time Saving Exercise Posture Monitoring System	237
Sarvesh Virkud, Aditya Mehta, Necil Dabre and Jignesh Sisodia	
24 Sarcasm Detection on Twitter Data: Generative Versus Discriminative Model	247
Ashwini M. Joshi and Sameer S. Prabhune	
25 Network Intrusion Detection System Using Machine Learning Approach	255
Mrunal Teli, Riya Singh, Minal Kyada and Ramchandra Mangrulkar	
26 Ascent of Pre-trained State-of-the-Art Language Models	269
Keval Nagda, Anirudh Mukherjee, Milind Shah, Pratik Mulchandani and Lakshmi Kurup	
27 Syt-AJ: Treating Lazy Eye Using Virtual Reality	281
Tejas Ved, Jay Chauhan and Neha Katre	
28 Deep Learning Challenges in Medical Imaging	293
Vaibhav Saraf, Pallavi Chavan and Ashish Jadhav	
29 Home Security System Usings Face Recognition	303
Janhavi Baikerikar, Vaishali Kavathekar, Yash Agarwal, Sanika Bhat, Christine Polly and Saloni Juwatkar	
30 Semantic Web-Based Knowledge Extraction: Upper Ontology Guided Crime Knowledge Discovery	311
Kaneeka Vidanage, Noor Maizura Mohamad Noor, Rosmayathi Mohemad and Zuriana Abu Bakar	
31 Attribute Reduction for Medical Data Analysis Using Rough Set Theory	325
Prerna Bhavsar, Parth Jhunjhunwala and Lynette D'Mello	

32 Emotion Identification Using CNN-Based Transfer Learning	337
Aarti M. Karnade, Prachi Dalvi and D. R. Kalbande	
33 Secure and Decentralized Academic Transcript System Based on Blockchain Technology	345
Jalla Manikanta Swamy and Keyur Parmar	
34 A Brief Survey of Sentiment Analysis	353
Ashwini Save and Narendra Shekokar	
35 Comparison of Traditional Machine Learning and Deep Learning Approaches for Sentiment Analysis	365
Dhvani Kansara and Vinaya Sawant	
36 Stock Price Prediction Using Grammatical Evolution	379
Lynette D'Mello, Aditya Jeswani and Janice Johnson	
37 Smart Notifications Based on Total Relevancy Score	391
Bhaktij Patil, Hemal Mamtora, Kunal Mandalya, Niket Parekh and Pramod Bide	
38 Ensemble Method Combination: Bagging and Boosting	399
Jyoti Deshmukh, Mukul Jangid, Shreeshail Gupte, Siddhartha Ghosh and Shubham Ingle	
39 YOLO Based Recognition of Indian License Plates	411
Jimit Gandhi, Purvil Jain and Lakshmi Kurup	
40 Efficacy Analysis of Technology Approaches Toward Auto-assignment of Clinical Codes to the US Patient Medical Record	423
Milind Godbole and Anuja Agarwal	
41 Survey of Sentiment Analysis on Social Media	441
Suyash Chavan, Jai Puro, Sarthak Kawade and Pramod Bide	
42 Survey on Detection and Prediction Techniques of Drive-by Download Attack in OSN	453
Madhura Vyawahare and Madhumita Chatterjee	
43 System to Fight Counterfeit Drugs	465
Soham Tendulkar, Alban Rodrigues, Keval Patel and Harshal Dalvi	
44 Comparative Analysis of Hand Gesture Recognition Techniques: A Review	471
Parth Shah, Raj Shah, Maulik Shah and Kiran Bhowmick	
45 Human Activity Recognition	479
Chetashri Bhadane, M. Umair Siddiqui, Siddhant Soni and Vijay Pratap Singh	

46 Reference Model Storage Covert Channel for Secure Communications	489
Dhananjay M. Dakhane and Vaibhav E. Narawade	
47 Texture Synthesis and Style Transfer for Aesthetic Design Creation	497
Aditya Shah, Dhruvin Shah, Harsh Shah, Sneha Shahane and Khushali Deulkar	
48 Credit Card Fraud Detection Using Meta-classifiers Consisting of Semi-supervised and Supervised Algorithms	503
Rutuja Taware	
49 Correlation Between Number of Hidden Layers and Accuracy of Artificial Neural Network	513
Purva Raut and Apurva Dani	
50 Face Completion Using Generative Adversarial Network	523
Purva Raut, Moxa Doshi, Monil Diwan and Karan Doshi	
51 Novel Approach of Computing Optimal Placement of Solar Panel Using Augmented Reality	533
Krupalu Mehta, Avani Sakhapara, Dipti Pawade and Vivek Surve	
52 Automated Scoring System for Online Discussion Forum Using Machine Learning and Similarity Measure	543
Dipti Pawade, Avani Sakhapara, Rishi Ghai, Shruthi Sujith and Sneha Dama	
53 Intrusion Detection: A Machine Learning Approach	555
Vipul Borhade, Aparna Nayak and R. Dakshayani	
54 Automated Damage Detection in Operational Vehicles Using Mask R-CNN	563
Naeem Patel, Shantanu Shinde and Freddy Poly	
55 Audio Tagging for Emotion Recognition: A Review	573
Raj Shah, Harshi Thaker, Shaurya Shettigar, Mahima Thakar and Chetashri Bhadane	
56 Implementation of ROS in Drones for Animate and Inanimate Object Detection	579
Chinmay Sankhe, Bhavesh Ahuja, Austin Coutinho, Chandan Bhangale and Nupur Giri	
57 Modeling CNN for Best Parameter Investigation to Predict Viable Exoplanets	591
Gaurav Singh, Sarang Gawane, Amandeep Prasad and Kalpita Wagaskar	

58	Blockchain-Powered Real Estate System	609
	Aman Jain, Bhumi Chitroda, Aditya Dixit and Harshal Dalvi	
59	Optimizing Reverse Image Search by Generating and Assigning Suitable Captions to Images	621
	Dhvani Kansara, Aditya Shinde, Yashi Suba and Abhijit Joshi	
60	Genre-Based Indian Viewer Movie Reviews—A Sentiment Analysis Classification of Text and Emoticons with a Supervised Machine Learning Approach	633
	Ashish Modi and Elizabeth L. George	
61	Detecting Offensive Text on Facebook Using Natural Language Processing and Machine Learning	645
	Ameya Kasbekar, Rashmi Rana, Vidhi Shah and Abhijit R. Joshi	
62	A Proposed Approach for Financial Investment Recommendation and Decentralized Account Management	657
	Harshita Khandelwal, Harsh Jain, Shreeyaa Agrawal and Vinaya Sawant	
63	User-Based Personalized Text Summarizer	663
	Pratik Nalage, Jay Parekh, Arjav Metha and Abhijit R. Joshi	
64	Juxtaposing Deep Learning Architectures for Breast Cancer Classification	679
	Purva Raut, Viraj Mehta and Akshen Kadakia	
65	A Novel Design for Voice-Enabled Home Automation and Personalized Recommendation System	691
	Harsh Parmar, Narendra Shekokar and Pranjali Thakre	
	Author Index	699

About the Editors

Dr. Hari Vasudevan has his Master's degree in production engineering as well as a Postgraduate diploma in industrial engineering from VJTI (University of Mumbai) and Ph.D. from IIT Bombay. He has also done a 3-month full time certificate programme (ERP-BaaN) from S. P. Jain Institute of Management and Research, Mumbai under the University Synergy Programme of the BaaN Institute, Netherlands. His areas of interest include manufacturing engineering, manufacturing systems and strategy, market orientation of manufacturing firms and world-class manufacturing. He is an approved Ph.D. guide at the University of Mumbai and NMIMS (Deemed to be University) and has so far guided 7 Ph.D. students. He is the president of Indian Society of Manufacturing Engineers (ISME); Life member of ISTE, New Delhi; Fellow of the Institution of Engineers (India); Fellow of ISME and a senior member of IEDRC. He has over 26 years of experience in teaching and 2 years of experience in industry. Presently he is working as the Principal of Dwarkadas J. Sanghvi College of Engineering, Mumbai. He has published over 115 papers in International and National conferences as well as journals and has to his credit a couple of textbooks and few book chapters in various publications. He has also received 3 awards, of which 2 are from National professional bodies and 1 from an International NGO.

Dr. Antonis Michalas have received his Ph.D. in Network Security from Aalborg University, Denmark, and currently, he is working as an Assistant Professor at the Department of Computing Science at the Tampere University of Technology, Faculty of Computing and Electrical Engineering. Prior to that, he was working as an Assistant Professor in Cybersecurity at the University of Westminster, London. Earlier, he was working as a postdoctoral researcher at the Security Lab at the Swedish Institute of Computer Science in Stockholm, Sweden. As a postdoctoral researcher at the SCE Labs, he was actively involved in national and European research projects. Dr. Antonis has published significant number of papers in the field related journals and conferences and has also participated as a speaker in various conferences and workshops. His research interest includes private and secure e-voting system, reputation systems, privacy in decentralized environments,

cloud computing, trusted computing and privacy-preserving protocols in participatory sensing applications.

Dr. Narendra Shekokar has received his Ph.D. in Engineering (Network Security) from NMIMS University, Mumbai, and he is working as a Professor and Department of Computer Engineering at SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai (Autonomous College affiliated to University of Mumbai). He was a member of Board of Studies at the University of Mumbai for more than 5 years, and he has also been a member of various committees at the University of Mumbai. His total teaching experience is 22 years. He is Ph.D. guide for 6 research fellows and more than 25 students at postgraduate level. He has presented more than 50 papers at international & national conferences and has also published more than 20 research papers in renowned journals. He has received the Minor Research Grant twice from University of Mumbai for his research projects. He has delivered an expert talk and chaired a session at numerous events and conferences.

Dr. Meera Narvekar has obtained her Ph.D. in 2016 from SNDT University, Mumbai, in the area of Mobile Computing. Her thesis work was on optimization of data delivery in mobile networks. She is currently the Head of Department of Computer Engineering at SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai (Autonomous College affiliated to University of Mumbai). She is presently a member of Board of Studies at the University of Mumbai. She was nominated as a Senate member of the University of Mumbai in 2008. She has a total experience of 20 years in teaching. Dr. Meera Narvekar has published around 45 papers in various international and national journals and conferences. She currently is guiding projects with applications in agriculture, which has also received grant from University. She has delivered talks in various conferences and workshops. She is also in the reviewer list and has been session chair of many conferences.

Chapter 1

Career Counselling Chatbot Using Cognitive Science and Artificial Intelligence



Godson D'Silva, Megh Jani, Vipul Jadhav, Amit Bhoir and Prithvi Amin

1 Introduction

Choosing a career can be easily defined as the biggest decision that one takes at a very young age and cannot be decided through the help of any mathematical equation. Making career choice has become much more difficult these days for youth as well as domains, etc. Still today, career choices are unfortunately made by parents, due to the vast number of option available in every stream in the form of diversifications, specializations and niche domains. Still today, the career choices are made by youths based on the advice, suggestions, experiences, stories, recommendations, news reports, etc. This is where the mistake happens. It is very important for youth to understand that not one size fits all and parents to understand that their child is different than the rest. It is high time that youth and parents considered a scientific and proven way to judge what career option will fit [2].

Considering the current scenario where unemployment in India is projected to increase from 17.7 million last year to 17.8 million in 2017 and 18 million next year. In percentage terms, the unemployment rate will remain at 3.4% in 2017–18. One of the crucial reasons for this drastic situation is due to the lack of proper career guidance. The approaches used till date are a psychometric test, which aims to provide measurable objective data that can provide a better all-round view of a youth's suitability in a career. On the basis of results obtained from the psychometric tests, related career path is prompted, but there are no ropes shown to achieve success in that career.

To overcome this issue, the need of the hour is to have a personalized career counselling for youths to help them provide guidance for such an important decision

G. D'Silva (✉) · M. Jani · V. Jadhav · A. Bhoir · P. Amin

St. John College of Engineering and Management, Vevoor Village, Palghar, India
e-mail: dsilvagodson@gmail.com

in their life. The energy, skills and aspirations of young people are invaluable assets that no country can afford to waste. Helping them to develop and realize their potential to the fullest is a precondition for sustainable national growth and development [1].

2 Conceptual Framework

The proposed system consists of main components such as language translation API, Emotion service API, MongoDB, data-driven documents (D3.js), E-portfolio dashboard, and bot framework integrated with Microsoft Azure services like virtual machines, blob and table storages, and application insights as illustrated in Fig. 1. Initially, the user can communicate with the system through various chatbot platforms such as Skype, Slack, and Facebook Messenger. The user can communicate with the chatbot in their preferred language through these chatbot platforms. Thereafter, the language translation API is used to translate the user preferred language into English for the chatbot to understand. After this, the psychometric tests will be conducted of the user and based on these test results the user can identify the personality and jobs suitable for them. Once this is done, the chatbot will provide training and mentoring to the users to achieve the necessary skill set for their desired jobs. All these results will be stored in MongoDB to keep the records of all these users. The chatbot will also make use of emotion service API to capture the facial and textual emotions of the user to identify the emotional state of the user as it can affect the results of the psychometric tests. The emotional state API is also used to identify whether the user is satisfied with the entire counselling procedure or not. Accordingly, based on the test results and the training and mentoring of the user an E-portfolio will be generated of the user. This E-portfolio will consist of the psychometric test results,

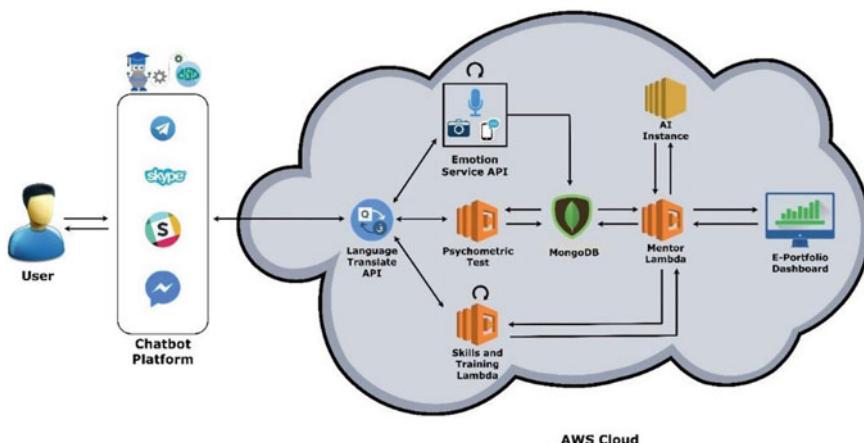


Fig. 1 System architecture

growth rate and impact of the training and mentoring phase, and also, an e-resume will be generated which can be used by the user to apply for various jobs [3].

3 Implementation

3.1 Conducting Psychometric Tests and Storing the Data in MongoDB

This operation is as illustrated in Fig. 2.

1. Initially, the user can communicate with the chatbot through various platforms like Skype, Slack, Facebook Messenger, etc.
2. The chatbot will ask the user to take the psychometric tests, and this request will reflect in the psychometric test Azure function.
3. While the user is appearing for the test, the chatbot will capture the facial and textual emotions of the user at regular intervals.
4. The results of the psychometric tests will be saved in MongoDB.
5. These results will also be displayed to the users, to let them know about their personality and job traits and the areas in which they are lacking and might need mentoring.

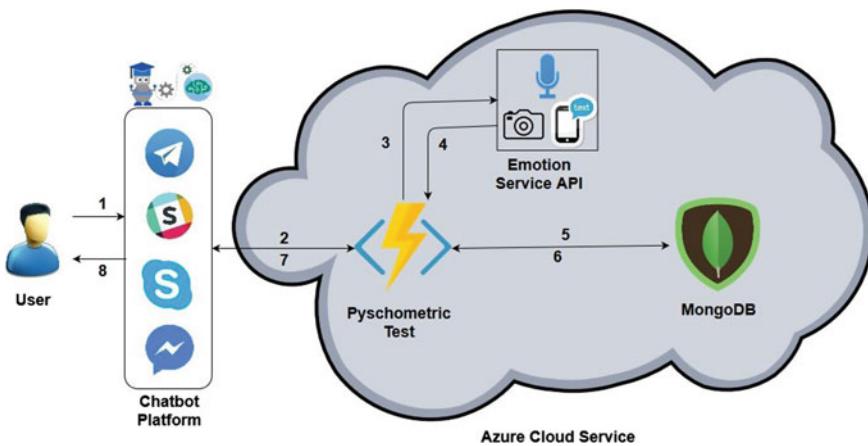


Fig. 2 Conducting psychometric tests and storing the data in MongoDB

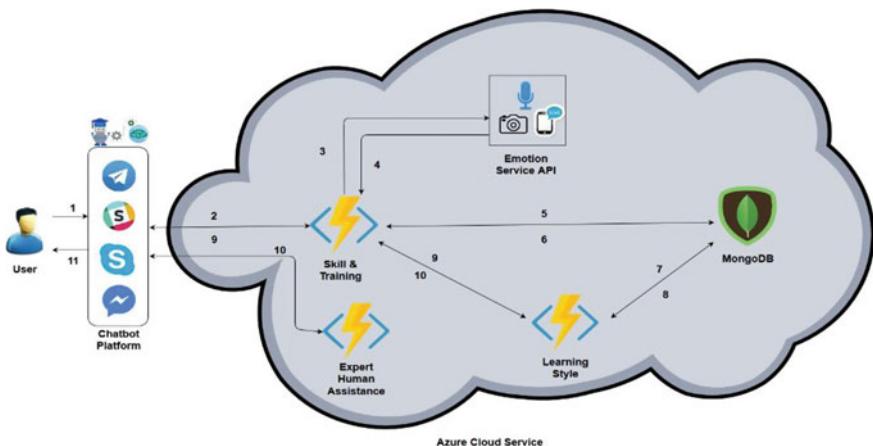


Fig. 3 Identifying the learning style of each individual and perform mentoring

3.2 Identifying the Learning Style of Each Individual and Perform Mentoring

This operation is as illustrated in Fig. 3. The process of identifying the learning style and mentoring procedure is given.

1. In this, the process will start in the skills and training phase where the user will have to select a course for training.
2. While this is done, the emotion analysis will be done to identify the user interests.
3. Once the user has chosen a content, it will be fetched from the MongoDB by the chatbot along with specific Q&A's.
4. The content will include different learning styles which will be displayed randomly to identify the users' learning style throughout the entire course.
5. The chatbot will further train the user based on the learning style.
6. Moreover, if the user is still unsatisfied, the chatbot will provide a feature of live assist where the user can speak to an actual mentor and get their doubts cleared.

3.3 Building an E-portfolio Based on Test Results

In this phase, the E-portfolio of the user is built as shown in Fig. 4.

1. The test results of the user are saved in the database.
2. Similarly, the data of learning styles is also stored in the database.
3. Based on these data, E-portfolio of the user is generated.
4. This E-portfolio contains all the results of the user. The chatbot will also generate a professional resume for the user.

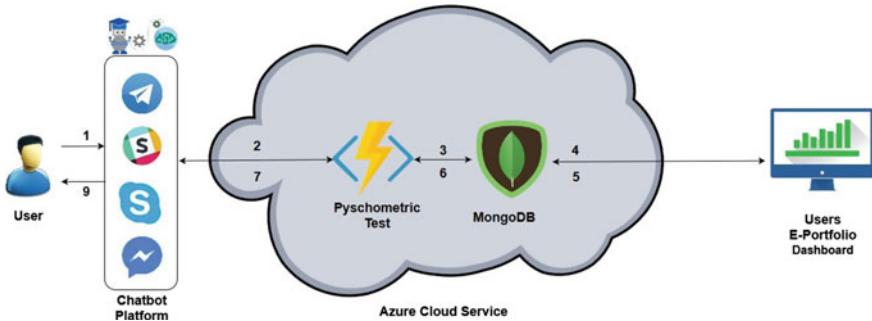


Fig. 4 Generated E-portfolio

4 Results

In this paper, a highly robust, multi-modal and scalable faster architecture is proposed which tries to solve the issues of traditional transport system by integration of MongoDB and d3.js and deploying it on Amazon public cloud.

4.1 Conducting Psychometric Tests to Identify Users' Personality and Suitable Jobs

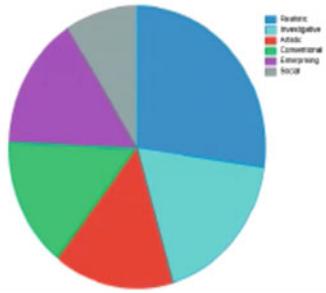
Initially, the chatbot starts by taking two types of psychometric tests of the users, namely ‘big 5 test’ and ‘holland code test’ to identify the personality of the user and the jobs suitable for them, respectively. The test results will depict the personality of the users, and also, a list of suitable jobs will be displayed. The user can also know whether his/her personality matches with the job recommended to them through the holland code test. With the help of these psychometric tests, the user will have various career options to choose from. The users can also visualize the big 5 and holland code test results by referring to the pie charts shown in Fig. 5 [1].

4.2 Training and Mentoring the User for Their Desired Job

Once the psychometric tests are done, the next step is to train and mentor the user to improve their skill sets for their desired jobs. Various methods of learning such as visual images, videos or articles will be given to the user to train them and identify their learning styles. Moreover, integrating natural language processing with chatbot can help the bot learn better and add more human touch to it [4]. If the user is still unsatisfied with the teaching methods of the chatbot, a live assist feature is provided by the chatbot where in a user can communicate with an actual mentor in their

Holland Test

Holland Test is used to derive a persons job



Big5 Test

Big 5 Test is used to derive a persons personality

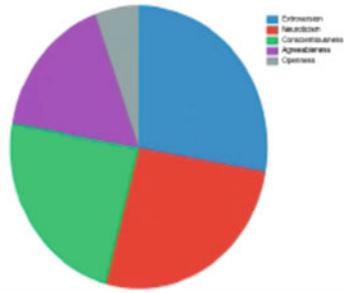
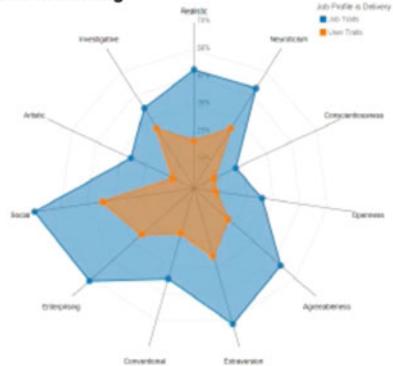


Fig. 5 Pie charts of Big 5 and Holland code tests

Before Mentoring



After Mentoring

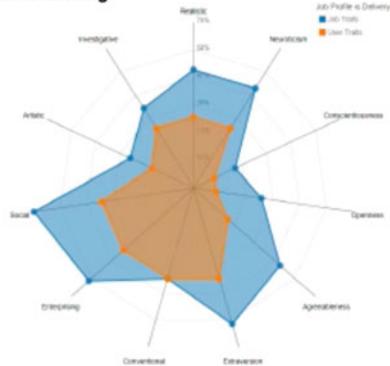


Fig. 6 Spider chart of progress report

preferred language. The users can also visualize their progress by referring to the spider charts shown in Fig. 6.

4.3 Generation of Professional Resume and List of Companies

After the mentoring process is done, the user now has the required skill set for a particular job. So, the chatbot will provide additional features to the user where a professional resume will be generated. All the user has to do is to upload his/her SSC and other qualification mark sheets, and the chatbot will be trained enough to extract all the data from those mark sheets and automatically build a professional resume.

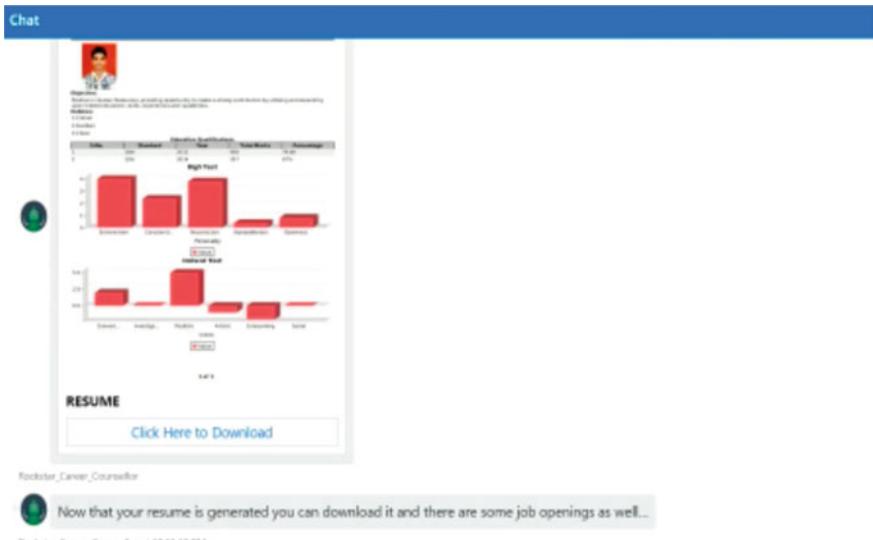


Fig. 7 Generated professional resume

Moreover, the chatbot will also have provisions where a list of companies with job openings will be displayed. This list of jobs will also consist of appropriate website links of these companies for the user to not only look into the companies' profile but also apply for those companies. The resume is shown in Fig. 7 [5].

4.4 *Data Visualization of the Big 5 and Holland Codes Generated*

The code generated from the big 5 test and holland code test can be visualized given in Fig. 8. The figure consists of a circle with many different layers. The innermost layer of the circle is divided into five partitions in which each and every partition depict the five personality traits of the big 5 test. Also, all the personality traits are assigned a specific colour. Similarly, the next layer consists of further partitioning of these colours into six sections which show the attributes of the holland code test. The aim of this figure is to show various jobs that are available for a user based on the fusion of their big 5 test and holland code test.

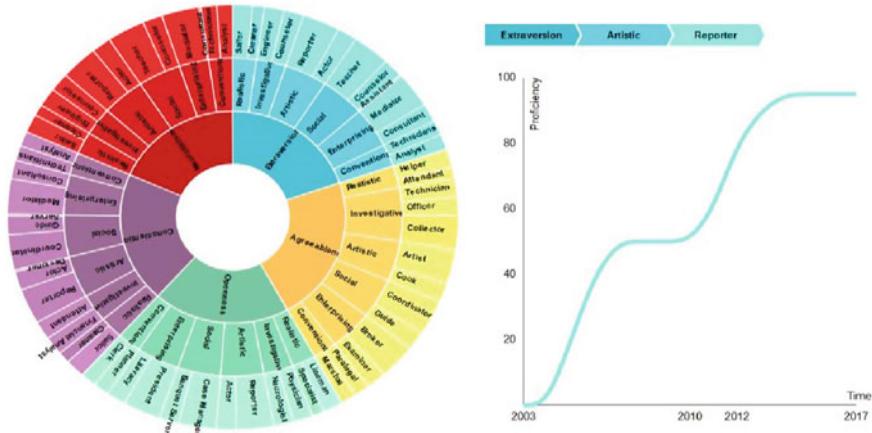


Fig. 8 Data visualization

5 Conclusion

Career counselling helps the students to know the pros and cons of the different streams, courses and educational options and the career path it offers thus, the students can make an informed choice, and get a career assessment that helps avoid the risk of change in career path later in life. The introduction of this career counselling chatbot will help an individual with all these aspects of choosing an appropriate career and help students grow in their respective career fields. The chatbot ticks all the boxes by not only providing mentoring but also generates professional resume and E-portfolios which can help users with suitable jobs. This system will just be one of its kinds which will felicitate appropriate career counselling for an individual. Moreover, the amalgamation of Microsoft Azure cloud services with open-source technologies like MongoDB and d3.js will help the system to be more robust, reliable and user-friendly.

References

1. Goldberg LR (1992) The development of markers for the Big-Five factor structure. *Psychol Assess* 4(1):26
2. Duijzer D (2017) Can we improve the user experience of chatbots with personalisation. In: Linebaugh T The effect of Holland's RIASEC interest inventory on the vocational identity development of Japanese high school students
3. Brown D (2006) Career information, career counseling, & career development, 9th edn
4. Chowdhury GG (2003) Natural language processing. *Ann Rev Inf Sci Technol* 37(1):51–89

1	Career Counselling Chatbot Using Cognitive Science ...	9
5.	D'silva GM et al (2017) Real world smart Chatbot for customer care using a software as a service (SaaS) architecture. In: 2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (ISMAC), IEEE	

Chapter 2

Voice Assistant for Ubuntu

Implementation Using Deep Neural Network



Shrutiika Singh, Harshita Arya and P. Arun Kumar

1 Introduction

Speech recognition is an area of artificial intelligence. It is a technique which identifies words spoken by human and converts them into machine understandable format [1, 2]. Speech recognition systems require “training” which includes contains audio waves and its text. There are also systems which do not require training, and such systems are called as speaker-independent systems [2]. Speech recognition has benefited from deep learning and big data. Many systems have introduced in the literature [2]. Such systems will need only a part of natural language processing, i.e., automatic speech recognition (ASR).

Acoustic modeling and language modeling are important parts of modern statistically based speech recognition algorithms [2]. Hidden Markov model (HMM) is widely used in many systems [3]. Earlier dynamic time warping was used but then was replaced by HMM. Then, neural networks emerged as an attractive acoustic modeling approach in ASR. Since then Neural Networks are used mostly. There are many applications of speech recognition like healthcare, car systems, military, telephony, people with disabilities, aerospace, hands-free computing and many more.

S. Singh (✉) · H. Arya · P. Arun Kumar
MIT World Peace University, Pune, India
e-mail: shrutika051220@gmail.com

H. Arya
e-mail: harshitaarya43@gmail.com

P. Arun Kumar
e-mail: pshivam.arunkumar@gmail.com

Kalinga Institute of Industrial Technology, Bhubaneswar, India

2 Literature Survey

Sirius is an open-source end-to-end standalone intelligent personal assistant (IPA) service. It receives queries in the form of speech or images and returns results in the form of natural language. Sirius implements the core functionalities of an IPA including speech recognition, image matching, natural language processing and a question-and-answer system. The advantage of Sirius is that it is open source and free of cost. Voice input is recorded and given to ASR, and it then converts voice to text and searches it in the database. Then the answer is displayed or action is performed. If the input is in the form of image, then image matching is done from image database and then searched in database to display appropriate output. Disadvantage of this system is that it needs Internet connection and also it can do only search on Internet and give answers and it cannot perform system operations [4] (Fig. 1).

Alexa is Amazon's cloud-based voice service available on tens of millions of devices from Amazon and third-party device manufacturers. With Alexa, we can build natural voice experiences that offer customers a more intuitive way to interact with the technology they use every day [5]. Alexa uses a two-step, scalable, and efficient neural shortlisting and re-ranking approach to find the most relevant skill for a given voice command. First, it is detected if the spoken word is the “wake word” by the wake word detection (WWD). Then, the Voice is recorded and sent to the Alexa cloud through AVS protocol. Voice is then converted to text, understood and that action command is sent back to the device. The device (media player) then outputs the received command. Alexa is user-friendly, and it also provides Home automation facility. But there is a disadvantage that it is a paid system and available only in English (Fig. 2).

Being one of the best intelligent voice assistants, Siri comes with many functions. You can ask Siri questions, get directions, send text messages and emails, get recommendations, make reservations and more. When you trigger Siri by saying “Hey Siri,” in the back-end, a powerful speech recognition system by Apple kicks off and converts voice into text—“Hey Siri.” Deep learning is used here. The servers at Apple run various natural language processing algorithms to understand the intent

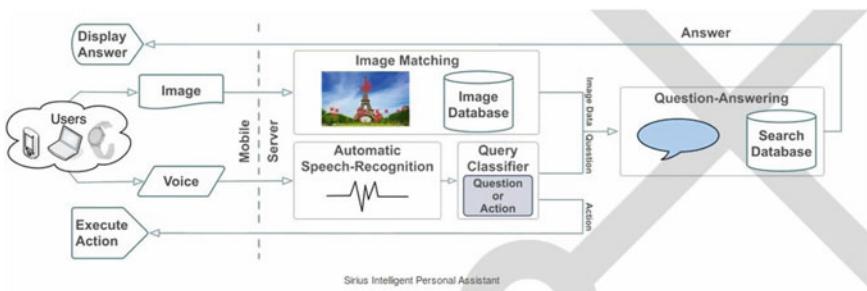
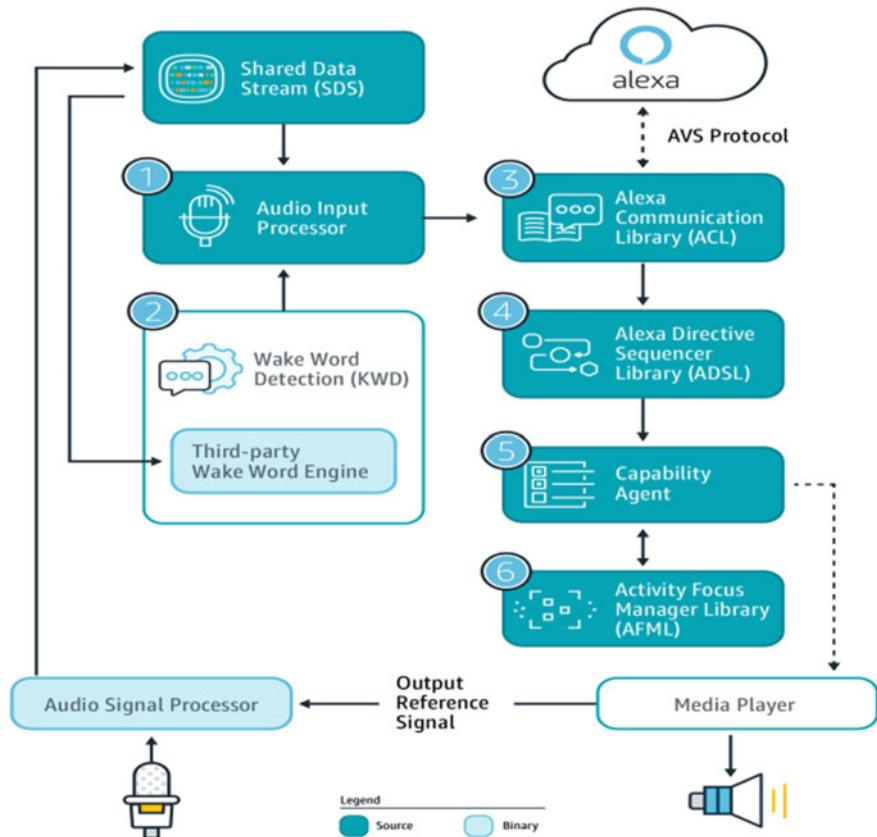
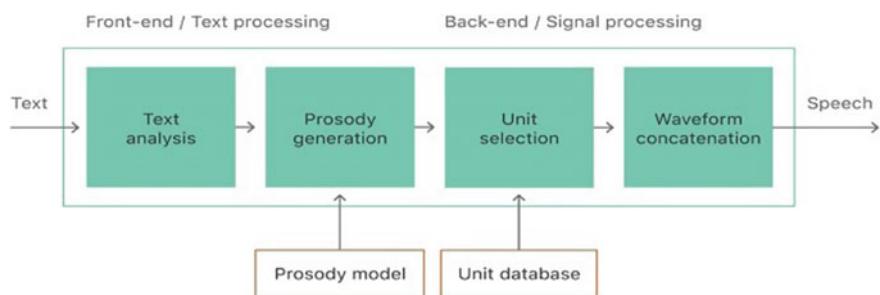


Fig. 1 Sirius architecture

**Fig. 2** Alexa architecture**Fig. 3** Block diagram of Siri

of what the user is trying to say. Siri is very simple to use, and it also comes with non-English options. But there are also some listening issues reported by users [6] (Fig. 3).

The Google Assistant is a virtual assistant powered by artificial intelligence and developed by Google that is primarily available on mobile smart home devices. Unlike Google now, the Google Assistant can engage in two-way conversations. Now it also does bank transactions [7]. Voice is recorded and sent to servers of Google to process as shown in Fig. 4. It first, breaks down the voice and then consults the database, and lastly, its operations are performed. It has many advantages like cast video on television, good sound quality, hands-free calling and can also identify different voices. Some disadvantages are that it is costly.

Table 1 shows the comparison study of different voice assistance systems.

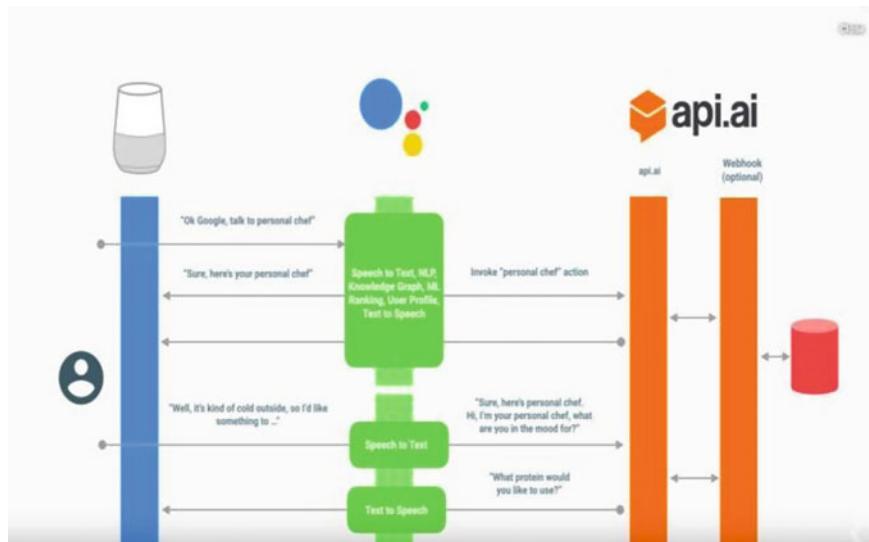


Fig. 4 Google Assistant

Table 1 Comparison of various voice assistants

Voice assistants	Technology used	Language	Algorithm	Where used
Sirius	NLP Image processing	Python	DNN-HMM DNN-GMM	Open-source system
Alexa	NLP	Python, Java	DNN	Home automation
Siri	NLP	Python	DNN	Apple phones
Google Assistant	NLP	Python	DNN	All mobile devices

3 Problem Definition

Interaction is a crucial task of communication between human and computer. The right technique of interaction will decrease communication gap between human and computer. So various user support methods are introduced. This paper proposes a voice assistant model for Ubuntu. Ubuntu system is open source; it is used by very few people. Most of people are not familiar to its functions and think as it is less user-friendly. Thus, in order to ease the functioning of Ubuntu system, there is need of a system that uses speech recognition technique. The U-Voice Assistant System that provides directions to build a system that takes input in the form of human voice commands, convert them into text form and perform particular action.

4 Working of the System

The system is first trained using deep neural network algorithm. For training purpose, we required audio file and its corresponding text. While training it maps features like pitch, amplitude, maximum frequency, minimum frequency, timbre and many more. We have used Libri Speech dataset for training. Finally, the model is built (Fig. 5).

U-Voice Assistant performs the following steps:

1. The system takes input in the form of human voice commands. The system requires a wake word as “Ubix.” If user does not speak Ubix at the beginning of the sentence, the system will not record anything.
2. The recorded audio is stored in the ‘.wav’ format. Then this audio is filtered to eliminate the background noise.
3. Deep neural network algorithm is used to convert this wav file into text form with the help of training data.

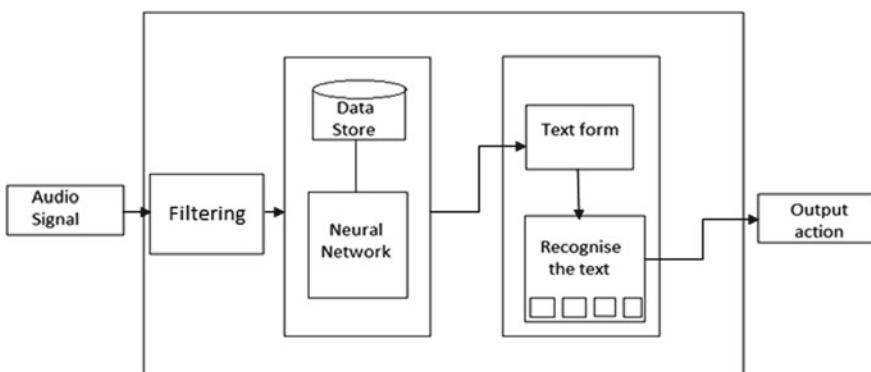


Fig. 5 System architecture

4. Then, only the required keywords are captured and corrected. The probability of each character is determined and character with highest probability is predicted.
5. Suppose the keywords identified are “ubix please open the browser,” then keywords like “ubix,” “open,” “browser” are identified and then using if-else conditions respective function is called.
6. The functions consist of code which executes the command to perform specific function. For open browser command—“xdg-open <http://www.google.com>” is used.
7. If the user speaks something irrelevant for which the system does not find any command, then a dialog box with warning message is popped up.

5 Algorithm

The filtered audio signal is given as the input to the DNN algorithm. We have used DNN for processing the audio signal, that is, to convert the waveform into the text. DNN has six hidden layers (Fig. 6):

1. Convolution Layer

Here, 1D convolutional network is used with 220 feature detectors. In this layer, features of wave are detected like amplitude, max frequency, min frequency, pitch and many others. It uses the function to eliminate negative values as follows:

$$f(x) = \max(0, x)$$

where x is the input. After this, max pooling is performed. It is used to avoid overfitting of features. It discards the data and replaces data with max value of each feature [8].

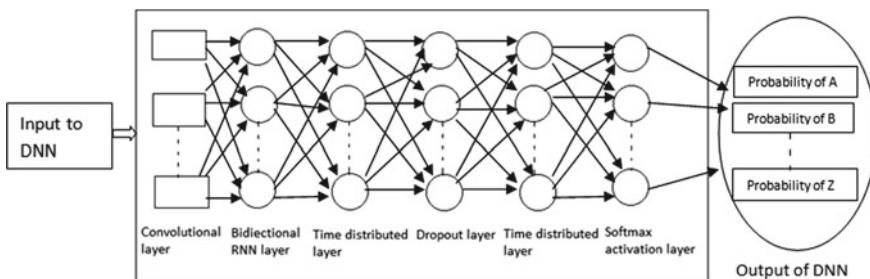
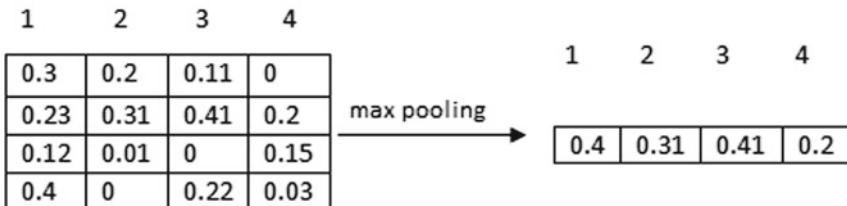


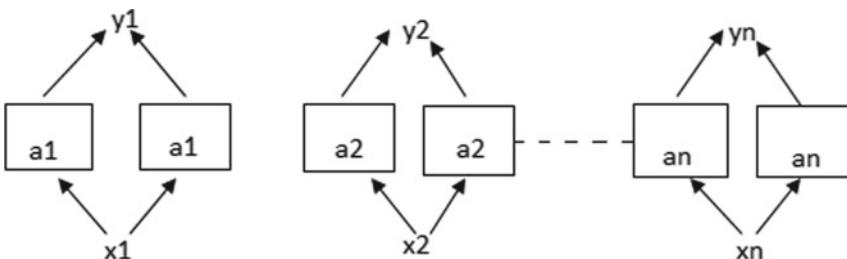
Fig. 6 Hidden layers of deep neural network

Features



2. Bidirectional RNN Layer

Output of the previous layer is input to this layer. This layer has memory that stores previously determined data. Bidirectional RNN is just combining two RNN's, one is for processing sequence from right to left and another from left to right [9].



where x_1, x_2, \dots, x_n are input from convolutional layer, and y_1, y_2, \dots, y_n are output of bidirectional layer.

3. Time Distributed Layer

Output of Bidirectional RNN is input to this layer. This layer is used for mapping input and output. Suppose if there are 13 time steps for 50 samples with 29 possible outputs, then we will get $13 * 50 * 29$ possible outputs. But we require only 29 possible outputs. By applying time distributed dense, $13 * 50 * 29$ possible outputs are mapped into 29 possible outputs.

4. Dropout Layer

Output of time distributed layer is input to this layer. In this layer, some neurons in the network are discarded to avoid overfitting. In our model, we have discarded 10% of neurons. For example, consider 20 neurons. If random numbers generated are 2 and 15, then 2nd and 15th neurons will get disconnected with other neurons.

5. Time Distributed Layer

After dropping neurons in the previous layer, those neurons are not considered for further processing. So, again this layer is used to keep one to one relation between input and output for remaining neurons.

6. Softmax Activation Layer

This is the final layer and it will calculate the probabilities for every 29 possible outputs. For example, if the probabilities of the classes are 0.8, 0.9, 0.7 and so on. The sum of probabilities of all the classes should be 1 which exceeds in this case. So, Softmax function is used so that the sum of probabilities of all the classes will become 1.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \text{ for } i = 1 \dots k$$

where k = number of classes.

Then, class with maximum probability is predicted for each neuron. The class predicted by majority of the neurons will be the final output. The letter with maximum probability will be the required character.

ALGORITHM

1. Prepare the data and split it into Train and Validation datasets.
2. Build a Sequential Bidirectional Neural Network with above-mentioned layers.
3. Add a final Softmax layer to get probabilities.
4. Train the network on the obtained Train datasets using Backpropagation algorithm to get optimal probabilities as results.
5. Evaluate results using ground truth labels in Validation datasets.

6 Experimental Setup

We tested the system in a Ubuntu 16.04 LTS Operating System with Intel Core i5 Processor and 8 GB RAM computer. We used TensorFlow v1.14 utilizing the internal Keras estimators to train and evaluate our model. We used Python 3.5 and Python 3.6 for testing purposes. Various python libraries such as Pandas for storing data, NumPy for matrix calculations and Scikit-Learn for evaluating our models are used. The results indicating the performance of our system are explained in the next section.

7 Results

With the help of proposed system, we recognize text from speech using speech recognition technique (Table 2).

1. We have used deep neural network for converting wav file to text.
2. Dictionary is used to capture the required keywords.
3. Commands are used to perform corresponding actions.

Table 2 Confusion matrix

	Predicted positive	Predicted negative
Actual positive	TP = 52	FN = 20
Actual negative	FP = 10	TN = 8

1. Accuracy

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

Accuracy of our system is 67%.

2. Precision

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

Precision of our system is 84%.

3. Recall

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

Recall of our system is 72%.

8 Conclusion and Future Scope

The U-Voice Assistant System has been developed and tested with proper data. The system results in the required output. The system is able to perform correct actions based on the voice commands. It reduces user's overhead and time required for typing. The interaction between user and computer is made easy, and the user is able to perform operations with ease and accuracy. Inconvenience in performing actions is minimized to a possible extent. This system can be implemented in different regional languages in the future. Also, implementing this system at a large scale can eliminate the use of keyboard and mouse to some extent. The accuracy of the system can be increased by providing more training data. More user-friendly interface can be created.

References

1. Xu Y, Du J, Dai LR, Lee CH (2014) An experimental study on speech enhancement based on deep neural networks. Proc IEEE Signal Process Lett 21(1):65–68
2. Jaity N, Nguyen P, Senior A, Vanhoucke V (2012) Application of pretrained deep neural networks to large vocabulary speech recognition. In: Proceeding of interspeech, pp 2–5
3. Yu Y Research on speech recognition technology and its application. In: Proceeding of IEEE, 23 Apr 2012
4. Chung H, Iorga M, Voas J, Lee S (2017) Alexa, can i trust you? Proc IEEE Comput Soc 50(09):100–104
5. Hebah HO, Nasreddin AA, Omari R (2017) Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation. In: Proceeding of IEEE computing conference
6. Xie Y, Le L, Zhou Y, Raghavan VV (2018) Deep learning for natural language processing. In: Handbook of statistics. Elsevier, Amsterdam, The Netherlands
7. Morgan N (2012) Deep and wide: multiple layers in automatic speech recognition. Proc IEEE Trans Audio Speech Lang Process 20(1):7–13
8. Li L et al (2013) Hybrid deep neural network—hidden Markov model (DNN-HMM) based speech emotion recognition. In: Proceeding of association conference on affective computing and intelligent interaction, pp 312–317
9. Deng L (2013) Design and learning of output representations for speech recognition. In: Proceeding of NIPS workshop learn. Output representations

Chapter 3

Trust Attacks in Internet of Things: A New Data-Centric Cybercrime on Enterprise Use Case



Parikshit N. Mahalle and Gitanjali R. Shinde

1 Introduction

The Internet of Things (IoT) is declared as prominent and potential transformation which will lower the business cost and in turn employees will become more productive. The use of IoT devices in enterprise is increasing at faster rate to improve business efficiency, customer service and delivery of valuable insights. Due to advancement in computing and communication technologies, the number of devices connected to the Internet is increasing exponentially and is expected to cross 50 billion by 2020 [1]. These devices include cameras, sensors, IoT cockpits, meters, thermostats, radio-frequency identification (RFID) tags and routers. All these Internet-connected IoT devices are continuously generating large amount of data and due to this malicious activity can go unnoticed. In recent years, there has been lot of attention on cyberattacks on IoT [2, 3] where attackers exploit small vulnerabilities to compromise the whole IoT system. There are many use cases in IoT like smart home and smart office where applications of IoT are at end user's side and attacks are underestimated. This is possible because in such use cases, all attack paths are not supervised. The most important and valuable asset of an individual is personal data. The data explosion is increasing at faster rate due to several reasons like availability of Internet at cheaper rate; number of devices connected to the Internet and more devices are being manufactured with the capabilities of sensing, computing

P. N. Mahalle (✉)

Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune, India
e-mail: aalborg.pnm@gmail.com

Center for Communication, Media and Information Technologies (CMI), Aalborg University,
Copenhagen, Denmark

G. R. Shinde

Smt. Kashibai Navale College of Engineering, Pune, India
e-mail: gr83gita@gmail.com

and communication. In fact, the data was small when IoT was not known and has become big when the IoT is known. Today a petabyte of data, i.e., 1,024 terabytes can meet the definition of big data. However, forward 10 years, even petabytes of data will not be qualified as big data [4].

Cyberattacks are upcoming threats to IoT ecosystem. A cyberattack is an event in which attacker makes malicious attempt to breach the information system of an enterprise or individual. The main intention of cyberattacker is to gain some benefits by hitting the business [5]. In the literature, there has been lot of attention paid on security attacks in cyber-physical system. However, IoT-enabled security attacks have been given little attention. Essentially in the context of IoT, devices are installed at backend, i.e., at device layer with reference to the identity management framework presented by Mahalle and Railkar [6]. This makes IoT devices more prone to the attacks that can affect critical systems and services. Common types of cyber-attacks like malware, phishing, man-in-middle attack, denial-of-service attack and Structured Query Language (SQL) injection attack have been given more attention [3]. Malware is malicious software entity which enters into IoT network through vulnerability. It happens when user clicks unknown links and either ransomware, i.e., blocking access to main functional components of the network or spyware, i.e., secretly obtaining information during data transfer from external device. Phishing is one of the main cyber-threats and its main goal is to steal important data like user credentials. It occurs from fraud communication received from attacker. However, this information appears to come from reputable sender. Man-in-middle attack is eavesdropping attack where attacker enters into two communicating entities and after interrupting the traffic, data is filter or stolen. In the context of IoT, unsecure Wi-Fi and malware breached into the IoT device can be the two common entry points for an attacker. Denial-of-service attack is an attack on availability where attacker tries to flood device or server which is intended to fulfill legitimate requests. Flooding exhausts resources and bandwidth resulting into the system down. In SQL injection attack, the server using SQL is penetrated with malicious code by attacker to reveal or change information. Web Site search box is an appropriate place to submit this malicious code. However, there is a new cyber-threat, i.e., trust attacks which are getting adapted intelligently in the surrounding. The main objective of this paper is to discuss trust attacks and its impact in IoT.

In this paper, we present detailed survey of cyberattacks on various IoT use cases in enterprise domain. Cyberattacks include many common attacks which exploit vulnerabilities in IoT use cases and the components. We focus on upcoming data-centric trust attack, i.e., those whose actual goal is to affect the business of an enterprise by modifying the critical data. We focus mainly on the attacks which have been verified and published by potential researchers and the mitigation strategies for trust attacks. To the best of our knowledge, this is the first systematic approach to review and assess data-centric trust attacks in IoT domain which is a new cybercrime in enterprise use case.

The rest of the paper is structured as follows. In Sect. 2, we review some potential work presented in the literature on cyberattacks on IoT and their proposed methodologies. In Sect. 3, we define, present and discuss new trust attacks in IoT with respect

to various use cases in enterprise. In this section, we also discuss the occurrence and impact of trust attacks and how it affects the business of an enterprise. Section 4 presents mitigation strategies and possible roadmap for trust attacks. Finally, Sect. 5 presents research and implementation gaps and concludes the paper.

2 Related Work

Recently, various cyberattacks and solutions with different objectives have been proposed to address cybercrime in IoT. Kamhouna proposed game-theoretic approach for deception of cyberattacks in IoT [7]. Author claims that the first phase of cyberattack is reconnaissance phase where attacker breaches the network to find vulnerabilities. The goal of the proposed work is to make this reconnaissance phase very difficult for attackers. Author presents cyber-deception as a complex game and proposed solution based on the proactive generation of honeypots. Anton et al. presented threat analysis of cyberattacks in IoT and used Internet-wide scanning for detecting smart IoT devices across the globe via TELNET protocol. Statistical analysis proves that routers, IP cameras and DVRs are key IoT devices responsible for launching cyberattacks as these devices are accessible via TELNET protocol [8]. This study concludes that around 15% of the host does not require authentication through TELNET.

Deep learning-based approach is presented by Zhou et al. to prevent irreversible damage caused by cyberattack in IoT [9]. Authors presented extensive literature survey of application of machine learning to address cyberattacks in IoT. Authors proposed deep feature embedding learning architecture, and experimentation results show that proposed technique achieves high accuracy time efficiency. However, the proposed work does not deal with the dimensionalities reduction of data. An interesting approach of detecting cyberattack with the help of deviations in the code running on their processors from known firmware is presented by Riley et al. [10]. Two IoT processors, the Atmega328 P and the PIC24, are selected for performing experimentation. In the proposed work, classifiers are implemented to identify code running on the devices. This classification helps to detect and identify register values based on signatures. Conventional attack detection techniques are based on monitoring and analysis of network logs; however, these logs and network statistics can be forged.

Li et al. have proposed novel method to detect cyberattacks based on the energy auditing and analytics [11]. Authors have proposed dual deep learning model which adaptively learns the system behaviors. The proposed model can detect both cyber and physical attacks. Other possibilities of cyberattack can happen by using port scanning tools [12], where the weak ports are identified in order to make network properties abnormal. Vulnerabilities, threats, intruders and cyberattacks in IoT are presented and discussed in [13]. Authors have discussed threats and intruder classification for cyberattacks.

Table 1 IoT devices accessible via TELNET

Device type	Number of devices
DVR	107,392
Router	98,438
IP camera	78,518
Switch	24,226
WAP	17,572
Web camera	16,183
Disk station	6675
Wireless router	3769
Firewall	2925
Satellite receiver	2190

IoT-based cyberattacks on smart grid and their mitigation are presented by Yilmaz et al. [14]. Minimally invasive attack mitigation via detection isolation and localization is proposed by authors where distributed DoS attacks are mitigated which are created by IoT-powered botnets. Hierarchical intrusion detection approach is used to mitigate distributed DoS attacks which are independent of any data type assumptions. Application of blockchain technology to address cyberattacks in IoT is discussed in [15]. The discussion is also supported by some examples of the use of blockchain-based solutions in various IoT environments. However, the proposed scheme and implementation details are not presented.

Internet-connected IoT devices are major threats of cyberattacks and using one device in an enterprise, attacker can get hold of entire IoT network. Interesting cyber-security examination has been carried out by Prokofiev et al. [8]. Network foot-printing tool is used to check the devices which are accessible by TELNET protocol. Table 1 gives statistics of the devices which are potential candidates for cyber-criminals being compromised by TELNET [8]. The figures presented in Table 1 are alarming and clearly concludes that these IoT devices are the key gateways into the broader enterprise network. Breaking it down even further, IoT hacks lead to the tampering of data, spying multimedia devices, extracting network details, obtaining user credentials, etc.

State of the art shows that significant attention has been paid to the cyberattacks in IoT. Most of the work presented in the literature focuses on distributed DoS attacks, malware and physical attacks and their defenses using machine learning and blockchain technologies. Mitigation strategies presented are not lightweight and missing with complete proof of concept. However, very little attention has been paid to the upcoming trust attacks which affect business of an enterprise to large extent. The following section presents detailed overview of trust attacks.

3 Trust Attacks

In enterprise, IoT devices are more vulnerable to attacks due to their ubiquity. These devices can be hacked in few minutes; however, it might take many days or weeks to mitigate it. If these devices become infected, then attacker can create an automated DoS attack using botnet. Jamming or spoofing techniques can be adapted by attackers to hack smart enterprise. There is a variation in the risk posed by various IoT devices. As per the report presented in [16] by world-class ethical hacker, IoT devices are categorized with respect to the risk they pose and these categories are listed below in the order of least to more severe.

1. Damaging—this device interferes in the enterprise network, extracts private credentials and includes devices like smart fridges and light bulbs.
2. Disruptive—these devices disturb enterprise operational processes and include connected printers, VOIP phones and smart video conferencing system.
3. Disastrous—these devices cause permanent damage by gaining access to enterprise information stored on the cloud or destroy critical equipment and include devices like IP-connected security systems.

Growth of any enterprise depends on the quality of services it delivers to the client, client satisfaction and increasing level of client's trust on the enterprise. Client trust cannot be built overnight. It requires persistent efforts by enterprise to meet the maximum expectations of the client. Enterprise includes various domains like banking, finance, agriculture, healthcare, etc. and all the use cases in these domains are IoT enabled which consists of device belongs to all the three categories mentioned above. Trust attacks are upcoming threats which destroy the trust relationship between client and enterprise and which in turn affect the business. In trust attacks, attackers are not interested in stealing the information; however, they are more interested in modifying the small critical information so that it will affect the quality of services being delivered to the client. This eventually results into the decrease in trust level of client toward enterprise. Attacker can hack one of these categories of devices, i.e., damaging, disruptive or disastrous and then these devices become a gateway to enter into broader IoT enterprise network. Trust attacks are special types of attacks which gains access to enterprise-critical information and affect the business.

Consider an enterprise which provides the services for IoT-enabled IP-connected infrastructure for climate control and energy meters. This enterprise is most trusted among customers for reliable and effective service delivery. This service uses heating, ventilating, and air conditioning (HVAC) equipment to perform automatic heating/cooling for residential, commercial and industrial infrastructures. These HVAC systems are major threats for attackers to gain control of IoT networks. The use of smart meter for monitoring of wireless energy is also creating additional threat. Attackers can compromise smart meters allowing them to alter the reported energy levels of company. This results into a serious issue of fraudulent accounting and metering and eventually clients will lose the trust on the enterprise. Attackers are also getting benefits from extensive use of wireless technologies by IP-connected

infrastructure. They exploit the vulnerabilities of these technologies and get access. HVAC systems are deployed on the existing internal backbone network where attackers have more opportunities to breach the network. They get access to data and carry out escalation of privileges affecting the system behavior. This creates serious inconvenience to the clients and results into the alleviating the happy index of clients. Once attacker gets hold of the IoT network; they can force critical rooms like server rooms to overheat and cause physical damage to the clients and thus causes trust attack.

Consider another example of popular IoT-enabled bank enterprise providing all financial and banking services to the customer. Bank has provision of mobile banking and electronic banking and also provides bank's application for performing various transactions and accessing the necessary details. Bank stored client database on the cloud provided by third-party cloud service provider. Client can use all the Internet-based services offered by bank in ubiquitous manner. There is a strong trust built between client and bank due to the quality and reliability of services provided by bank. In the sequel, bank is getting outstanding business. As stated earlier, SQL injection is one of the cyberattacks where the cloud server using SQL is penetrated with malicious code by attacker to reveal or change information. Unfortunately, attacker has full control over the client database. The main goal attacker is to update the records stored in database making data untrustworthy. Due to this data alteration, clients start getting wrong responses and unreliable services from the bank application. Due to these unexpected services and inconvenience, client loses trust on the bank resulting in alarming profit decrease. This is another example of trust attacks where attackers target not the resources, but they are more interested to affect client trust on an enterprise causing trust attack. The major actors in this trust attack scenario are enterprise (bank); client and cloud server and the sequence of actions in this attack are captured in Fig. 1.

4 Mitigation Strategies

Trust attacks are extremely difficult to detect as attacker make very small changes in the critical data. However, the main objective is to make the data untrustworthy and causes big cost to enterprise in long turn. The main goal of trust attack is not to steal data but to alter the data causing enterprise to lose the trust of their clients. In order to protect IoT networks from trust attack, an enterprise must ensure some recommendations. IoT device manufacturers should make a provision of associating some randomly generated passwords. Also there should be provision given to an enterprise to change the device credentials during commissioning phase. Enterprise should deploy machine learning-based proactive analytical tools to identify malicious activity. Enterprise should never underestimate attackers or rather they should always overestimate attackers in order to deploy more smart and secure solution in the network. There is a need of intelligent intrusion detection and prevention system which should emulate all Internet-connected IoT devices and collect the information about all events happening in the system proactively. Enterprise should block the

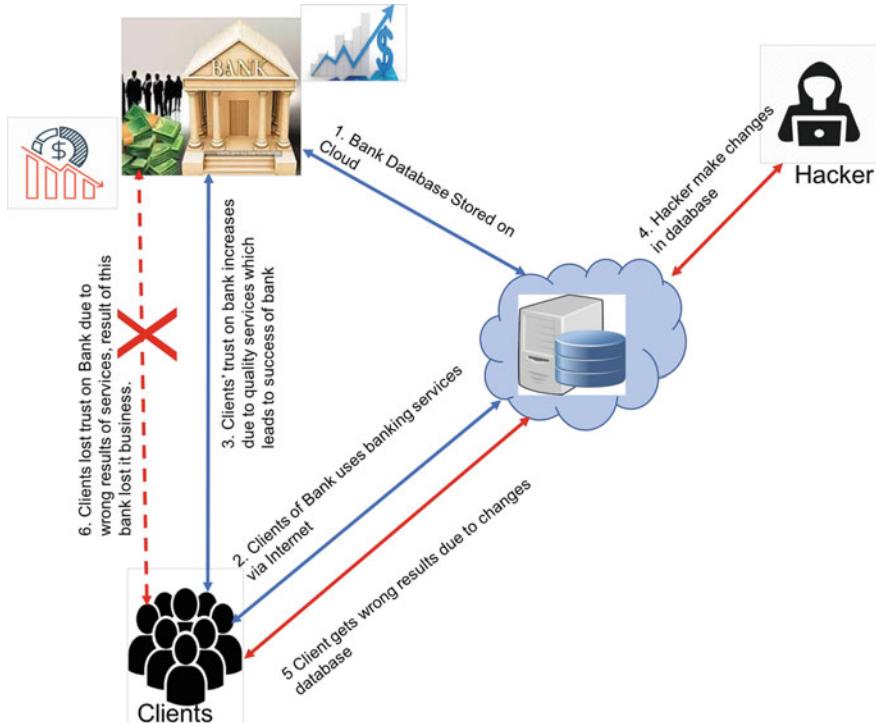


Fig. 1 Trust attack on bank use case

vulnerable network services like TELNET, Netcat, DuckDuckGo which can protect the network from malicious activities.

Enterprise should classify the devices into various categories like dumb, intelligent and smart devices [17] depending on their features and available resources. Based on this classification, each connected IoT device should adhere to stringent security protocol. As mentioned earlier, the most important and valuable asset of an enterprise is data which needs to be protected from trust attacks. The data explosion is increasing at faster rate due to several reasons like number of Internet-connected devices, on-demand compute power due to availability of cloud, etc. To protect this data from attackers, data generated by each device should be encrypted by lightweight encryption algorithm. The selection of encryption technique and algorithm needs a thorough analysis of the system. Last but not the least, there is a need of research for an automated, threat intelligence-based approach to mitigate trust attacks.

5 Conclusions and Future Outlook

The main contribution of our work is to introduce upcoming trust attack which is one of the cyberattacks on IoT. In this paper, we have discussed some of the major issues and challenges related to cyberattack in IoT. The main goal of cyberattack is to get unauthorized access to the IoT network. We have provided comprehensive analysis of recent potential work on cyberattacks on IoT from literature and it concludes that less attention has been paid to new upcoming trust attacks.

We have introduced and defined trust attack which is new threat which affects the trust of the clients toward enterprise in the next part of this paper. We have enlarged the bigger picture of trust attacks for an enterprise which provides the services for IoT-enabled IP-connected infrastructure and IoT-enabled bank enterprise providing all financial and banking services to the customer. Subsequently, this paper concludes with the mitigation strategies to address trust attack in perspective of manufacturer, researcher and enterprise.

The future outlook of this work is to design machine learning-based intelligent threat detection mechanism to prevent trust attacks. Evaluation of this mechanism against known attacks will be another interesting area.

References

1. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. *IEEE Commun Surv Tutor* 17(4):2347–2376, 4th Quarter
2. Cazorla L, Alcaraz C, Lopez J (2018) Cyber stealth attacks in critical information infrastructures. *IEEE Syst J* 12(2):1778–1792
3. Stellios I, Kotzanikolaou P, Psarakis M, Alcaraz C, Lopez J (2018) A survey of IoT-enabled cyberattacks: assessing attack paths to critical infrastructures and services. *IEEE Commun Surv Tutor* 20(4):3453–3495, 4th Quarter
4. Railkar PN, Mahalle PN, Shinde GR (2018) Access control schemes for machine to machine communication in inter of things: comparative analysis and discussion. In: 2018 IEEE global conference on wireless computing and networking (GCWCN), pp 59–63
5. Humayed A, Lin J, Li F, Luo B (2017) Cyber-physical systems security—a survey. *IEEE Internet Things J* 4(6):1802–1831
6. Mahalle PN, Railkar PN (2015) Identity management for internet of things. River Publishers, Denmark
7. Kamhoua CA (2018) Game theoretic modeling of cyber deception in the Internet of Battlefield things. In: 2018 56th annual Allerton conference on communication, control, and computing (Allerton). Monticello, IL, pp 862–862
8. Prokofiev AO, Smirnova YS, Silnov DS (2017) The internet of things cybersecurity examination. In: 2017 Siberian symposium on data science and engineering (SSDSE). Novosibirsk, pp 44–48
9. Zhou Y, Han M, Liu L, He JS, Wang Y (2018) Deep learning approach for cyberattack detection. In: IEEE INFOCOM 2018—IEEE conference on computer communications workshops (INFOCOM WKSHPS). Honolulu, HI, pp 262–267

10. Riley RA, Graham JT, Fuller RM, Baldwin RO, Fisher A (2019) A new way to detect cyberattacks: extracting changes in register values from radio-frequency side channels. *IEEE Signal Process Mag* 36(2):49–58
11. Li F, Shi Y, Shinde A, Ye J, Song WZ (2019) Enhanced cyber-physical security in internet of things through energy auditing. *IEEE Internet Things J*
12. McMaster University (2008) The five-layer TCP/IP model: description/attacks/defense—computing and software wiki
13. Abomhara M, Køien GM (2015) Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. *J Cyber Sec* 4(1):65–88
14. Yilmaz Y, Uludag S (2017) Mitigating IoT-based cyberattacks on the smart grid. In: 2017 16th IEEE international conference on machine learning and applications (ICMLA). Cancun, pp 517–522
15. Hassan QF (2018) Blockchain-based security solutions for IoT systems. In: Internet of things A to Z: technologies and applications. IEEE
16. Maler E, Catalano D, Machulak M, Hardjono T User-managed access (UMA) profile of OAuth 2.0; Kantara Initiative, Wakefield, MA (2016)
17. White Paper: how hackable is your smart enterprise? Know your IoT security risk by ForeScout (2016)

Chapter 4

Analysis of Light Pollution Prediction Using Mathematical Model and Machine Learning Techniques



**Aastha Sainger, Rishikesh Yadav, Pradnya Tipare, Samidha Waghralkar,
Vimla Jethani and Amit Barve**

1 Introduction

Light pollution is a wide concept and is linked to several problems which are originated by unnecessary and excessive use of artificial light. Research done over the years has gathered evidence that it causes a lot of health-related issues in men and women both [1]. It also leads to a change in the breeding track of migratory birds [1]. Several fields are now becoming aware of the consequences of light pollution ranging from human sciences to astronomical studies, etc. The main cause of light pollution which is the excessive and improper usage of artificial lighting has been discussed upon by various countries. Many other international organizations have also come up with guidelines for setting standards to eradicate light pollution. Currently, the instrument which uses high-dynamic-range (HDR) technology [2] to capture the data is Luminance Meter camera. Several images of bright surface are taken at different speeds which are then amalgamated to form one rich image. The maximum and

A. Sainger (✉) · R. Yadav · P. Tipare · S. Waghralkar · V. Jethani · A. Barve

Department of Computer Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, India
e-mail: aasthasainger97@gmail.com

R. Yadav

e-mail: rishisyadav5@gmail.com

P. Tipare

e-mail: pradnyatipare04@gmail.com

S. Waghralkar

e-mail: samidha.waghralkar1997@gmail.com

V. Jethani

e-mail: vimlajethani@gmail.com

A. Barve

e-mail: barve.amit@gmail.com

minimum luminance levels of the particular surface are analyzed with the help of a computer software [2].

The dataset for the implementation of this project has been obtained from International Dark-Sky Association (IDA) website [3]. Their pursuit is to safeguard and shelter the environment during night and our hereditament of sky during the night through standard and only required artificial lighting. The data has been captured through Defense Meteorological Satellite Program (DMSP) [3]. The following attributes have been considered in the dataset which play a major role in determining light pollution in an area: State, Location, Latitude, Longitude, SQM, Natural Brightness, Artificial Brightness, Bortle, Elevation and Population.

2 Literature Survey

2.1 Research Paper Survey

Various methods which help in detecting the light pollution are as mentioned below.

Prototype measurement method [2] determines the light pollution from a lighting source. Tools such as the LMK Luminance Meter and Chroma Meter were utilized. Chroma Meter was used for measuring illuminances of vertical plane and the Luminance Meter is used for measuring surface luminance. The CLA-200 m needs to be held adjacent to lamp in vertical position [2]. The Luminance Meter camera by making use of HDR technology takes several dynamic images of an illuminated surface at different speeds [2]. Portable spectrophotometer—The device helps in delivering the maps of the sky brightness during nighttime. The band belongs to any visible spectrum range. It takes the following conditions into account [4]:

- With finite requirements of time extravagant acclimatization.
- Mechanical plotting of the whole sky.
- Self-regulating reduction of data.
- Immense area of computation for rapid explication time.
- Complete measurement on field [4].

2.1.1 Night Sky Photometry with Sky Quality Meter (SQM, Unihedron)

The model which was completed with the objective of light pollution detection was first introduced by Walker in 1973. This model was used to deduce an empirical equation used to zero down good astronomical observation sites. Walker's law fundamentals are standardization of the cities as a contaminant focus and the application of the inverse-square law. The exponent of the distance observed is 2.5 and not 2 as for the inverse-square law. According to lighting designer Insulander [5], good design, safety and regulations are not difficult to be combined with sustainable lighting. But

lighting designer Mikaela Parsdotter Andersson describes that the aspects of safety or design are sometimes considered more important than saving nocturnal lives. The International Dark-Sky Association (IDA) along with the Illuminating Engineering Society joined together to establish common regulations to be used as guidelines for preventing light pollution and spreading awareness about it, the Model Lighting Ordinance [5]. The Korean government became the first one to make law in order to curb light pollution and its harmful effects.

2.2 *Similar Existing Project Comparison*

The present system has the following characteristics. The skywards light fluctuation established on DMSP satellite statistics is assessed. The maps represent the light pollution distribution in the airspace. This computation is done with the help of Garstang model [6]. The structure presumes Rayleigh dissipation by particles. It also presupposes Mie scattering by aerosols and considers disappearance across light paths. But this method based on linear regression model has the following limitations:

The technique of plotting unnatural sky flare has been in concern with:

1. Enumerating the luminescence at sea-extent.
2. Presuming the causes are at sea-extent.
3. Reckoning the skywards emission basis has the identical formation at all places.

In this instance, light pollution propagation function f is dependent on:

- Some common factors like direction of sky, atmospheric dispersal of molecules and their optical characteristics and appearance of emission function.
- Distance from origin.

It rejected calculations where there was existence of moon, ambiguous sky, big glow attachments in the adjacent area for which it inadequately displayed the position.

The main objective of the past work was to develop a mathematical model that shows the variability of the night conditions of light in places affected by the light pollution. The model played an important role in ecological light pollution research has helped to look out better astronomical observation locations for further enhanced studies of the nighttime.

But the prediction system being evolved in this research paper helps us in predicting whether which area is light polluted or not by taking into account various important parameters. After knowing about the appropriate and accurate predictions, simple measures can be imbibed in daily lifestyle so as to reduce the adverse impacts of light pollution.

2.2.1 Existing Models

Walker's Model: [7]

This model was based on the making of an equation that was empirical to zero down good astronomical locations. The basics of the Walker's law are:

- Standardization of cities as a contaminant focus.
- Use of inverse-square law.

The model:

$$I = C P d^{-2.5}$$

where: I = sky brightness, d = distance, P = population, C = depends on flux per inhabitant.

Treanor's Model: [7]

Treanor worked on Walker's model by using indicators related to scattering light by making use of suspension molecules. Treanor's model is not used for determining the contamination done by zenith's glare. The model:

$$P = L(r)L(N) = (Ar + B/r^2) \exp(-k/r)$$

where $L(r)$ = sky brightness, $L(N)$ natural sky brightness, P = the population, r = distance.

$$A = 1.8 \times 10^{-5} \text{.p}, B = 13.6 \times 10^{-5} \text{.p}, k = 0.026.$$

3 Proposed Work

We choose random forest algorithm and decision tree algorithm to predict whether there is light pollution in an area. A comparative study is done which includes analysis of both the algorithms to conclude which one is better based on their accuracies and cross-validation score. Another comparison of machine learning and mathematical model (Berry's model) with regard to predictions has been done.

3.1 Berry's Model: [7]

The drawback of Treanor's model of its inability in determining the adulteration done by the glare coming from the zenith by the glow of cities was rectified by

making some small changes in the equation. A function of distance for atmospheric circumstances was then obtained:

$$I = aP(bD^{-2} + cD^{-1})e^{-kD}, D = D^2 + H^2$$

where: I = sky brightness, P = population, D = distance, H = height of scattering layer, a = constant relating population, b and c = constants, k = aerosol extinction coefficient.

3.2 Decision Tree

Decision tree is a form of a tree structure which builds classification or regression models. An integrated decision tree is developed incrementally, while at the same time it breaks down a dataset into smaller subsets or trees. The tree which we get finally is a tree that has all the decision nodes and leaf nodes. The topmost node of the resulting decision tree is said to be the best predictor of all the nodes and that is why it is called as Root node. ID3 algorithm is one of the most prominent and efficient of all decision tree algorithms. To calculate the homogeneity of a record, ID3 algorithm uses entropy of the record.

3.3 Random Forest

Random forest algorithm creates the forest with a number of trees. It is a supervised classification algorithm. In the random forest classifier, the higher the number of trees in the forest gives high accuracy results. Random forest classifier (RFC) has the following advantages:

RFC can be used for classification.

1. Overfitting does not occur even in the case of excessive trees.
2. RFC takes care of the missing values on its own, i.e., implicitly.
3. RFC can be used for regression as well (same RFC used that for classification).
4. RFC can even be used for categorical values.

Random forest prediction pseudocode:

1. Using the test cases and generated trees in a random manner, RFC stores the result to predict the output.
2. Counts the odds in favor of each target that have been randomly obtained.
3. The one with the maximum count (votes) is then considered to be the final result for prediction.

4 Proposed Methodology

4.1 Using Mathematical Model

4.1.1 Berry's Model

Consider a place with the following attributes:

State: Kerala, Location: Kochi, Latitude: 9.5814, Longitude: 76.1531, Sqm: 19.55, Natural Brightness: 1.63, Artificial Brightness: 1460, Bortle: 5, Elevation: 7.

On applying Berry's formula, we get:

$$I = aP(bD^{-2} + cD^{-1})e^{-kD} \text{ and } D = D^2 + H^2$$

where: I = sky brightness, P = population, D = distance, H = height of scattering layer, a = constant relating population, b and c = constants, k = aerosol extinction coefficient [7], we get: $I = 0.001163240431$ nano-lamberts.

So here with this value we can predict that the following place Kochi is not light polluted (as artificial brightness is less than 200 cd/m²). Hence, in this way a mathematical model can also be considered for determining whether an area is light polluted or not by considering the threshold values.

4.2 Random Forest Algorithm's Methodology

To implement random forest algorithm for the available dataset, bands are used based on locations. This algorithm tries to create a number of CART models with large samples. For example, it will take 300 random observations and some initial observations to build a CART model. It will repeat the process and then make final prediction.

Band A: Goa, Band B: Maharashtra, Band C: Madhya Pradesh.

Cart 1: Natural Brightness in mcd/m², Cart 2: Artificial Brightness in mcd/m², Cart 3: Elevation in m.

Now consider an area with:

Natural Brightness = 2 mcd/m², Artificial Brightness = 1100 mcd/m² and Elevation = 454 m.

Now we need to predict whether which state among Madhya Pradesh, Goa and Maharashtra is more light polluted by considering Natural Brightness, Artificial Brightness and Elevation into consideration (the following values are based on assumptions for calculations).

The values in the tables indicate the number of places that fall under the specified category. For example, in the case of Table 1, there are six places in Madhya Pradesh that have natural brightness within the range 1–5 mcd/m². In the final results table

Table 1 Natural brightness (in mcd/m²)

Band	Goa	Maharashtra	Madhya Pradesh
Less than 1	2	8	15
1–5	6	9	11
6–9	18	16	2

Table 2 Artificial Brightness (in mcd/m²)

Band	Goa	Maharashtra	Madhya Pradesh
0–1000	2	16	8
1000–5000	8	12	7
5000–9000	21	12	5

Table 3 Elevation (in meters)

Band	Goa	Maharashtra	Madhya Pradesh
0–100	2	4	8
100–500	9	5	6
500–1500	12	7	2

Table 4 Final results

Cart	Band	Goa	Maharashtra	Madhya Pradesh
Nat. brightness	1–5	6	9	11
Art. brightness	1000–5000	8	12	7
Elevation	100–500	9	5	6
Mean of respective values		7.6	8.6	8

(Table 4), mean value is calculated for all three bands. Based on these values, it can be attributed that Band C (Maharashtra) has the largest value (8.6) and therefore it has the highest levels of light pollution when compared with that of Goa and Madhya Pradesh (Tables 2 and 3).

4.2.1 Cross-validation Score

This technique involves keeping a sample dataset on which training is not performed. Later, this model is tested on this sample.

Following is the process in cross-validation:

1. Keep aside a sample data set (testing set).
2. Train the model taking rest of the dataset also known as training set.

3. Use the reserved sample of the test (validation) set. This helps in determining the efficacy of the model's executional presentation.

If the model furnishes a constructive result on validation data, the current model can be adopted. Among the cross-validation methods available, k-fold cross-validation method has been selected. The working of it is as follows.

k-fold Cross-validation

1. Randomly split dataset into k number of folds.
2. For each k-fold, build the model on $k - 1$ folds of dataset.
3. Tests the model.
4. Record the errors.
5. Repeats this until each of the k-folds has served as the test set.
6. The average of k recorded errors is called the cross-validation error.

5 Experimental Setup/Implementation

5.1 Dataset

The dataset has been obtained from the website of US non-profit based organization, International Dark-Sky Association (IDA) [3]. Their pursuit is to safeguard and shelter the environment during night and our hereditament of sky during the night through standard and only required artificial lighting. IDA's primary viewpoint is to augment cognizance concerning the nighttime airspace. It also works incessantly to educate people regarding the refurbishment and indemnity of the nighttime airspace. IDA also dispenses guidance by provision of explication that can help annihilate light pollution. It has miscellaneous measures that focus on to safeguard the atmosphere for sustainable development [3].

The dataset values were captured by defense meteorological satellites also called as Defense Meteorological Satellites Program (DMSP). Meteorological, Oceanographic, and Solar-terrestrial physics for the United States Department of Defense is also monitored by these DMSP's [3]. The program is managed by the Air Force Space Command with on orbit operations provided by the National Oceanic and Atmospheric Administration.

The following attributes from the year 2011–2018 have been considered in the dataset which play a major role in determining light pollution in an area: State, Location, Latitude, Longitude, SQM, Natural Brightness, Artificial Brightness, Bortle, Elevation and Population. In a particular state, specific locations have been considered along with their coordinates and other light pollution determining factors such as SQM and Bortle. The units of various attributes are as follows (Table 5).

Table 5 Attributes considered

S. No.	Attribute name	Unit
1	SQM	mag./arc s ²
2	Natural brightness	mcd/m ²
3	Artificial brightness	μcd/m ²
4	Elevation	m

The attributes have been explained as follows:

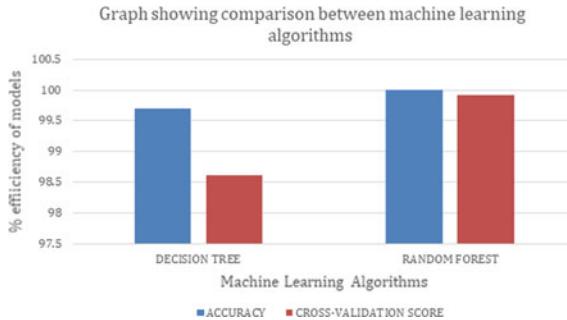
- Sqm: It measures the amount of light that hits the sensor. It converts that light into unit mag./arc-s. It has a range of 22–16 mag./arc-s. The light pollution increases as we move down the scale. The reading 21 indicates excellent dark sky while 16 indicates light-polluted sky.
- Natural Brightness: It gives the brightness of the sky which is measured in mcd/m². The computer screen is one million times greater than the natural sky brightness level of 0.25 mcd/m². More the impact of natural brightness in an area decreases with the increase in artificial lighting, more is the light pollution in an area.
- Bortle: It is a scale often used from 1 to 9. With scale 1 for an excellent dark sky up to scale 9 for the sky above an inner city [8].
- Artificial Brightness: Artificial light scattered in the atmosphere creates the most visible negative effect of light pollution—artificial skylight. In addition to hindering ground-based optical astronomical observations, it also hinders the natural path followed by migratory birds and turtles to return back to their ground where they lay eggs.
- Elevation: The elevation of a geographic location is its height above or below a fixed reference point, most commonly a reference geoid, a mathematical model of the Earth's sea level as an equipotential gravitational surface.

5.2 Machine Learning

Machine learning is implemented with these following steps:

- Step 1: Data Exploration: Data examination utilizes ocular inspection to apprehend the characteristics of the data which includes size or amount of data, completeness or correctness of the data, possible relationships among data

	A	B	C	D	E	F	G	H	I	J	K	L
1	Sr. No	State	Location	Latitude	Longitude	SQM	Natural_Brightne	Artificial_Brightne	Borte	Elevation	Population	Year
2	1	Kerala	Kochi	9.5814	76.1531	20.74255	1.63	1549.06	5	7	677381	2018
3	2	Kerala	Thiruvananthapuram	8.3013	76.5655	20.97597	1.33	1230.76	5	41	957730	2018
4	3	Kerala	Thrissur	10.3047	76.1237	21.71867	0.698	559.147	5	7	315975	2018
5	4	Kerala	Kozhikode	11.1527	75.473	22.14307	0.483	331.032	4	22	550440	2018
6	5	Kerala	Alappuzha	9.5931	76.2025	22.15368	0.479	326.768	4	10	174164	2018
7	6	Kerala	Kollam	8.5745	76.5222	22.99187	0.232	65.0393	4	106	397564	2018
8	7	Kerala	Telllicherry	11.4513	75.2914	22.7054	0.299	134.747	4	18	92664	2018
9	8	Kerala	Palakkad	10.4643	76.3934	21.86721	0.614	470.023	4	95	131726	2018

Fig. 1 Screenshot of dataset**Fig. 2** Comparison between machine learning algorithms

elements, etc. After the commencing discernment of the particulars is concluded, the data can be paired or rarefied by withdrawing the unserviceable segments of data that haphazardly inhabited space. Correcting improperly structured portions and expounding pertinent interconnections covering datasets is also an important step (Fig. 1).

- Step 2: Data Munging: Next important phase of machine learning project is called “data munging” also known as “data transformation”. It consumes up to 80% of the entire work as the imported data might be incompatible with the environment of machine learning algorithms. The amount of time needed for processing the dataset depends on how clean and complete the data is. Treating the missing values by filling them with mean, median, mean of the particular class, global constants, is a very effective and necessary step.
- Step 3: Building a Prediction Model: This process requires data mining and probability to forecast probable outcomes. Variables that are likely to influence future results are the predictors in every model. After the formulation of a statistical model, the model may employ a simple linear equation, or a complex neural network, mapped out by sophisticated software (Fig. 2).

6 Results and Analysis

On implementing the 2 algorithms we get, (Table 6).

Table 6 Accuracy of two machine learning algorithms

Algorithms	Accuracy	Cross-validation score
Decision tree	99.7	98.615
Random forest	100	99.917

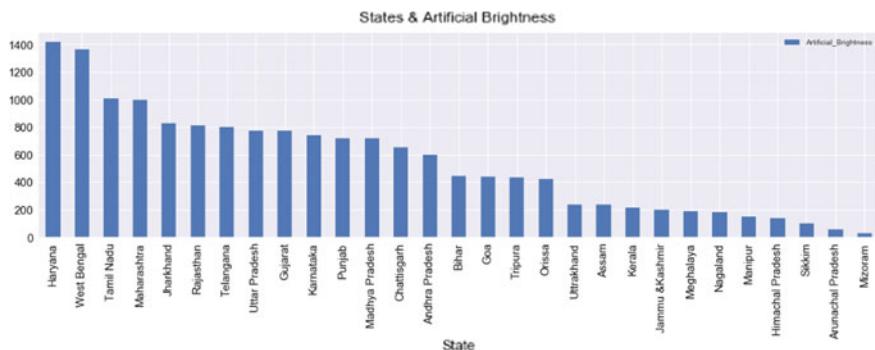
Although the two algorithms have same accuracy (approximately) but the cross-validation score of random forest (99.917%) is more than that of decision tree (98.615%).

Hence, with a greater cross-validation score of random forest we can conclude that Random forest algorithm is better than decision tree algorithm.

6.1 Outcome of Predictions

On plotting the graph of Artificial Brightness versus States, we get the following results (Fig. 3, Tables 7, 8 and 9).

From Fig. 4, we can see that the RMSE value of machine learning is 0.4533 and that of mathematical model is 0.7474. RMSE value is calculated using the following formula:

**Fig. 3** Graph showing artificial brightness in different states**Table 7** States having highest light pollution in decreasing order

S. No.	Name of the state
1	Haryana
2	West Bengal
3	Tamil Nadu
4	Maharashtra
5	Jharkhand

Table 8 States having least light pollution in increasing order

S. No.	Name of the state
1	Mizoram
2	Arunachal Pradesh
3	Sikkim
4	Himachal Pradesh
5	Manipur

Table 9 Results of prediction

No. of cities	Light polluted	Non-light polluted
Standard values	634	422
ML predicted values	518	538

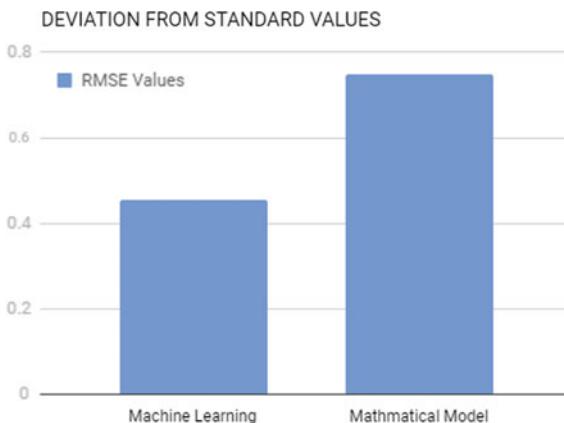


Fig. 4 Standard deviation graph of machine learning and mathematical model

$$\text{RMSE} = \sqrt{\left(\sum_{i=1}^n (P_i - O_i)^2 / n \right)}$$

here P_i is the predicted value and O_i is the observed value. RMSE value of machine learning involves the observed value of the dataset (O_i) and the value predicted by the machine learning model (P_i) using both the algorithms.

The lesser the RMSE value, the better the prediction model. Therefore, machine learning is better than mathematical model in case of prediction.

7 Conclusion

Light pollution has become a growing concern of modern times and should be addressed with immediate attention. Artificial light has caused a colossal damage to society in numerous ways like cancer, hampered ecosystem, etc. From the above inferences, it can be concluded that random forest algorithm is better than decision tree algorithm with regard to making predictions.

References

1. Rajkhowa DR (2014) Light pollution and impact of light pollution. *Int J Sci Res (IJSR)* 3(10):861–867
2. Lim HS, Ngarambe J, Kim JT, Kim G (2018) the reality of light pollution: a field survey for the determination of lighting environmental management zones in South Korea. *Sustain J* 10(2):374
3. www.darksky.org
4. Teikari P (2007) Light pollution: definition, legislation measurement, modeling and environmental effects. Universitat Politecnica De Catalunya, Catalunya, Barcelona
5. Insulander AM (2012) Light pollution—consequences and sustainable lighting design. Swedish University of Agricultural Sciences
6. Kang GK, Gao JZ, Chiao S, Lu S, Xie G (2018) Air quality prediction: big data and machine learning approaches. *Int J Environ Sci Develop* 9(1):8–16
7. Lamphar H, Páramo R (2010) Mathematical model for the measurement of light pollution. Polytechnic University of Catalonia, Department of light studies
8. Spoelstra H (2009) Dark Skies Awareness. IYA Cornerstone Project

Chapter 5

Design and Implementation of Library Shelf Management (LiBOT) Using Machine Learning



**Anish Pandita, Mit Parekh, Jitendra Sachwani, Romit Shah
and Ramchandra Mangrulkar**

1 Introduction

Internet of things (IoT) is the concept of interconnected and intelligent devices or object which has self-triggered operative switches. The interconnected devices have the inbuilt capacity of interactive with each other if they are placed in some confined area or outside the confined area through Internet [1, 2]. The IoT devices are intelligent, with the help of sensor, can take decisions wherever required. The majority of IoT applications are driverless cars, trucks tracking system, shopping mall inventory management system and so on. This is supported by technologies such as machine learning and artificial intelligence. Machine learning is a category of technology which makes the machine learn from given information. The basic usage of machine learning is to develop algorithms that aid in easy analysis of data [3, 4]. This IoT-based machine learning concept along with computer vision is adapted from human ability to learn from vision of objects and how they are presented. The combination is used to develop an effective algorithm to get an automatic visualized prospective. The concept of fusion of technologies such as machine learning and artificial intelligence with computer vision and IoT paves the way towards ground-breaking technolog-

A. Pandita · M. Parekh · J. Sachwani · R. Shah · R. Mangrulkar (✉)
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: ramchandra.mangrulkar@djsce.ac.in

A. Pandita
e-mail: anishpandita007@gmail.com

M. Parekh
e-mail: mitvparekh97@gmail.com

J. Sachwani
e-mail: sachwani.jitendra@gmail.com

R. Shah
e-mail: romits44@gmail.com

ical opportunities for the project. Looking towards the traditional implementation of libraries in and outside India, the majority of the work in library is manual. The main motivation behind this research work is to ease the burden of all the manual and laborious task that a librarian needs to perform. Library shelf management is the main work zone which brings authors attention and hence the efforts as well [5]. The initial drafting phases were full of multi-dimensional challenges ranging from the hardware sturdiness to software optimality and overall compatibility of software with hardware components. The rest of the paper is organized as Sect. 2 gives the background study of the proposed research. Section 3 gives basic idea of proposed model. Section 4 gives implementation of proposed model. Finally, Sect. 5 puts some results and implementation of proposed model. Conclusion and future direction are given in Sect. 6.

2 Background

2.1 Object Detection

Object detection using Haar feature-based cascade classifiers is a very effective object detection method. It is a technique where they make use of positive and negative images to train the classifier. The model requires a large volume of images of positive and negative classes in order to correctly determine the book [6, 7]. The negative images can be any random images without a book, and the positive image must have books present in them. The books must be of different sizes and orientation so that it becomes easy for the classifier to correctly determine a book if a test image already has such a kind of orientation. This classifier works on the principle of feature extraction. These are obtained by calculating the addition of different pixels which are found near the white rectangle and the ones which are found near the dark rectangle. This technique of feature extraction uses all the possible kernels to find out the features. This process involves enlarging the image to a just four pixels and extracting the feature, and this feature is tested for all the remaining images. There will be a lot of mistakes made by the model as the feature just works on white and black rectangles, but then, it is imperative to choose features that have minimal mistake rate that is the model selects the ones which can correctly classify the books as positive and negative. Now, this process is repeated till the required correctness and the error rate are gained or the minimum required features are fetched. The model is called a strong or a weak one if it has the ability to correctly classify the image as an image of book or not [8, 9]. When two or more models are merged and work in cohesion, they can form a strong classifier. At the last stage of the classifier, it has around 7000 features, so while testing the image instead of looking for all the 7000 features, it is better to check a frame with the help of which it can easily classify if the book is present in the image. If the frame is similar to that of a book region, it is tested further for the other features or else it is discarded [10, 11].

2.2 Capacitated Vehicle Routing Problem Solving Algorithm

The capacitated vehicle routing problem (CVRP) is a vehicle routing problem in which vehicles with limited carrying capacity need to pick up or deliver items at various locations. The bot in the proposed scenario will have certain maximum capacity of books that it would be able to load onto a particular container (vehicle) at any given instant of time. This algorithm decides what all books are needed to be picked without surpassing the maximum limit of a container to achieve the least travel time considering the entire table filled of books. The grid below shows the locations of the shelves to place the book with blue circles, and the containers location at the centre marked as 0. The number of books to be placed at a particular shelf is shown at the lower right of each shelf location. The overview is given in Figs. 1 and 2.

The distance matrix to each shelf from the container is then computed, using Haversine distance or Manhattan distance. The locations are specified in decimal degrees of latitude and longitudes. The containers in the system serve only their respective tables. A list of the entire path from the starting point to the corresponding shelf is returned to the container to initiate its movement towards the target shelf. An example of motion of the containers can be as shown below:

(i) Route for vehicle 0: Starts from location 0, moves to location 1 places 1 book, then moves to location 4 and places 4 books, then moves to location 3 and places

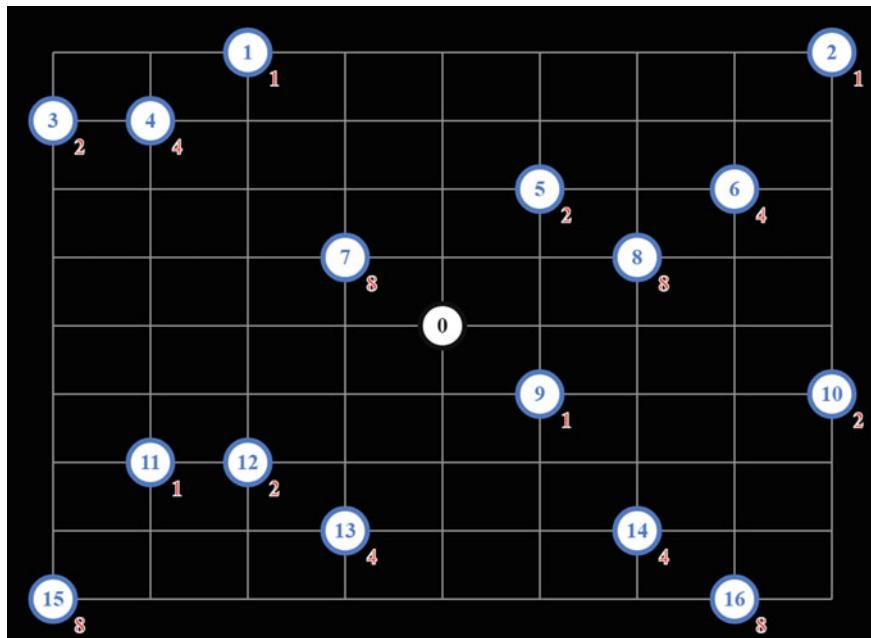


Fig. 1 CVRP capacitated vehicle routing problem: Initial Position

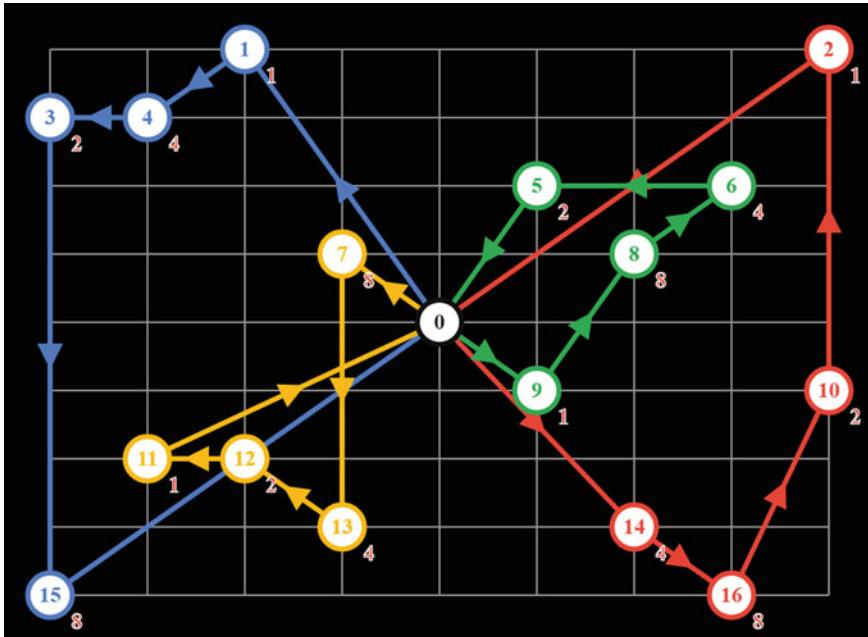


Fig. 2 CVRP capacitated vehicle routing problem: Intermediate Position

2 books, then moves to location 15 and places 8 books, finally end at the initial positions 0 with an empty container. Distance of the route: 2192 m. Max allowed load of the route: 15 (ii) Route for vehicle 1: Starts from location 0, moves to location 14 and places 4 books, then moves to location 16 and places 8 books, then moves to location 10 and places 2 books, then moves to location 2 and places 1 book, then moves to initial position 0 with an empty container. Distance of the route: 2192 m. Max allowed load of the route: 15 (iii) Route for vehicle 2: Starts from location 0, moves to location 7 and places 8 books, then moves to location 13 and places 4 books, then moves to location 12 and places 2 books, then moves to location 11 and places 1 book, then moves to initial position 0 with an empty container. Distance of the route: 1324 m. Max allowed load of the route: 15 (iv) Route for vehicle 3: Starts from location 0, moves to location 9 and places 1 book, then moves to location 8 and places 8 books, then moves to location 6 and places 4 books, then moves to location 5 and places 2 books, then moves to initial position 0 with an empty container. Distance of the route: 1164 m. Max allowed load of the route: 15.

2.3 Deterministic Binary Neurons

Binary neurons (BNs), as the name suggests, give binary values as their output. In our proposed model, we make use of deterministic binary neurons (DBNs). DBNs behave like hard thresholding function as the activation function.

2.4 Breadth-First Search Algorithm

Breadth-first search is the graph search algorithm that is used to effectively route the containers containing all the books from the table to all the shelves. The algorithm starts from the table being the starting, it then looks for the destination node at all its adjacent nodes, in all directions—up, down, left and right—then if it finds the destination shelf, it returns the path else recursively adds the adjacent node the list of possible paths and expands from there on ahead. All the existing table and shelf locations are treated as not passable terrain to avoid collision. Collisions with other containers are avoided by use of appropriate sensors data and implantation of collision avoidance in the algorithm which checks if the node on the path of the container would be travelled by any container during its movement. In this way, the algorithm visits all vertices in a graph that are away from the starting node which is the table and finds the shortest and most optimal path efficiently.

3 Proposed LiBOT (Library Bot)

The proposed Library Bot (LiBOT) would be designed keeping in mind the challenges encountered during related work and its detailed study. The system is going to be built on the client–server architecture. Thus, there exists a centralized server which will have the knowledge of the entire system environment. The bots are going to be clients. Using the HTTP requests, the clients are going to receive the communication to the server to fetch commands and return appropriate responses of required tasks. These tasks will be synchronized by the server for optimal resource usage. The LiBOT working is given stepwise in Algorithm 1.

4 Implementation and Analysis of LiBOT

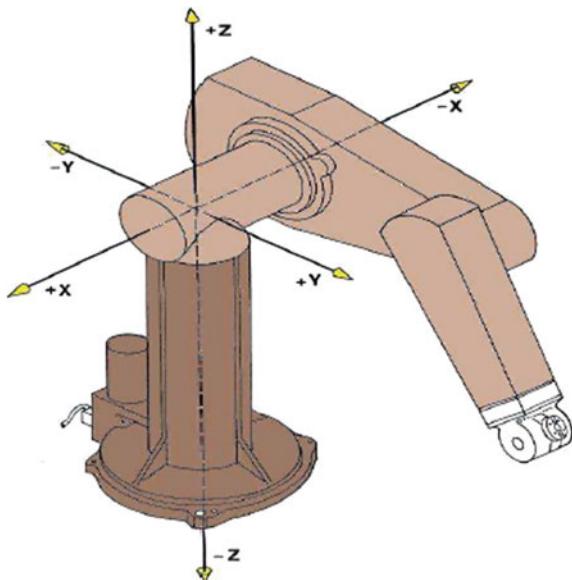
The system is built on the client–server architecture. Thus, there exists a centralized server which will have the knowledge of the entire system environment. The bots are going to be clients. Using the HTTP requests, the clients are going to communicate to the server to fetch commands and return appropriate responses of required tasks.

Algorithm 1 Working of LiBOT

1. The bot scans the table and recognizes the book using a colour code and matches the data with one in database.
2. The bot scans the table and recognizes the book using a colour code and matches the data with one in database.
3. The algorithm then calculates the optimal picking pattern for placement in container.
4. This is facilitated by the robotic arm which is main hardware component.
5. The claw is utilized to lift the book from the table.
6. The bot places the book in its container (heap).
7. Optimal path is calculated for traversal. This is done using travelling salesman problem algorithm.
8. After knowing path, the robot traverses the destination along the line (line follower algorithm).
9. On reaching the destination shelf, the bot picks up the book and places it in the respective shelf.
10. Book retrieval is also facilitated by this system.
11. The user asks the bot for a particular book, and the bot searches the database for the location of the book and if found fetches it for the user.

These tasks will be synchronized by the server for optimal resource usage. Puma robot arm is the architectural arm design that is referred for developing the structure of the robot. The arm is built on four axes which make it possible for the arm to move in the entire 3D space as shown in Fig. 3.

Fig. 3 Hardware architecture of the Puma Robot Arm



The reach of the arm is designed in such a way that it makes sure that it will serve the purpose of lifting and placing the books effectively. The camera is positioned over the table at the centre such that the F.O.V. of the camera covers the entire scene of the table. This will help in detection of the books and finding the associated unique 4×2 colour code on a particular book.

The proposed LiBOT performs activities given in following subsections.

4.1 Look

The process of LOOK involves computer vision which is done with the help of web camera fixed on the top of the table. A method based on image matching is used to detect the objects. The image of the table is captured, and then, the objects on the table are figured out. This is done with the help of object detection. Object detection works on the principle that it first recognizes the edges in the image. It then tries to find the corresponding objects by relating the edges to each other. This helps in finding the object. After this, the objects are tried to recognize. A pre-trained model for over 80 models is used to figure this out. This model uses the detected objects as input and gives the name of the object recognized as an output. The model generates output in less than a second successfully. The issues that this model can face are the blur images caused due to the moving of camera or the object that is being captured. The object that is being captured if has blurred edges then the object detection fails. To overcome this, a hidden Markov model (HMM) is used to figure out the actual image with less blur. This helps the algorithm to further detect the object. Once the object is detected, the latter algorithm can be used to recognize it. Hidden Markov model works on the concept of hidden states. It detects the current state of the scenario and tries to figure out the hidden states that it failed to capture. It then tries to link the starting states with the current states by creating a middle pathway. This way the hidden states are detected. The object detection in our proposed model is shown in Fig. 4.

4.2 Pick

One of the most challenging things in the field of robotics is the movement dimensions. It is difficult to find out the measurements that are required for the robot to move in the proper place. The measurements are needed to navigate the bot to the correct table. The measurements once found are fed to the hardware elements like motors and gears. Thus, it is important for the robot to find out these values. The robot manipulators are divided into two essential categories like kinematics and dynamics. The task is to find out the values using these categories. The goal of the process is getting a successful output value. The forward kinematics is used to get the output by using the measurement values. The inverse kinematics are used to get the measurements from the generated sensor values. The sensor values like the reading of



Fig. 4 Object (book) detection in LiBOT

ultrasonic sensors are used to get the distance. This helps in inverse kinematics. The inverse kinematics is calculated using the readings. The first step to compute forward kinematics (F.K) of robot is finding the motion parameters. This includes locate the robot arm, label joints and determine joint rotation or translation.

4.3 Travel

Here, the bot involves moving from shelf to shelf using following the non-deterministic approach because the bot does not know which book belongs to which shelf in the initial stages. The books are placed in the mobile containers after they are picked up by the unique robot arm. There is a possibility that books belonging to different stacks are placed in the container. The selection of the books is done using bin packing algorithm. The number of books in a container is selected on the basis

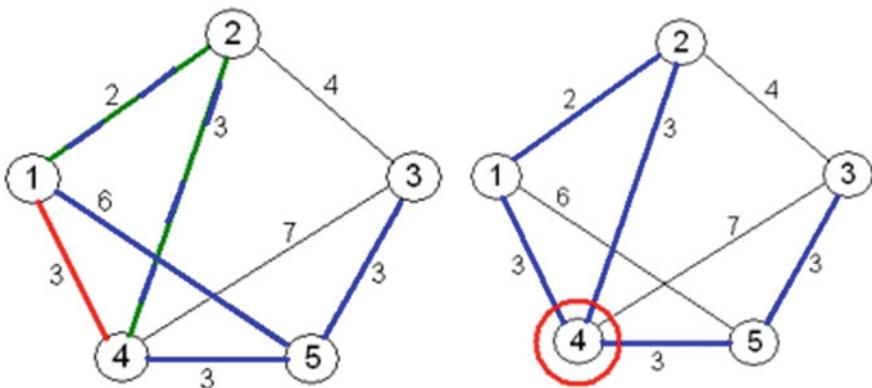


Fig. 5 Travel graph before and after updation

of the principle followed by the Beladys anomaly. It was found that when the container capacity was increased, the time to place them also increases exponentially. Similarly, if the container capacity is low, it will not fulfil our requirements, and the solution is not optimized. After the books are placed in the container, the main job is to find a shortest path in such a way that all the books are placed in their respective stacks in least amount of time. Thus, artificial intelligence comes into the picture. An algorithm similar to a travelling salesman is used to find the path. The library is considered as a graph with the stacks as the nodes, now if there are n nodes in a graph and one starting and ending node, a total of $(n - 1)!$ paths can be found. Thus, the time complexity of finding the shortest path is in the order of the $n!$. This is called for modification in the salesman algorithm. The algorithm used in the project has a different metric than the normal algorithm the metric takes into the consideration the weight of the book, obstacles in the path and finally the cost from going from one node to the other. Here, since the time complexity is the metric which has to be brought down, thus a non-deterministic approach is used. Figure 5 gives the travel graph example before and after update respectively.

4.4 Place

In this process, after the book reaches the corresponding stack, it is picked up by the robot arm again placed in the shelf. Now, there are different classification methods that are followed in each library. The classification themes are categorized into three kinds viz. based on language, based on synthesis and based on arrangement. There are various books written on the same topic in different languages. Now, the libraries have to maintain books from all genres in all languages, now if the book is classified according to the languages, it becomes easy for a person to fetch. Secondly, synthesis refers to a combination of code from different lists. This combination is

used to depict various attributes of a book. Library classification examples based on synthesis are colon classification, expansive classification and UDC (universal decimal). The arrangement classification theme is further divided into three categories namely [11–13]:

1. **Enumerative:** In this method, an alphabetical list of chapter heading is produced, and a unique identity is given to each title in alphabetical order. Generally, all libraries follow the enumerative pattern with hierarchical and faceted components.
2. **Hierarchical:** In this method, the topics are divided hierarchically. The more general topics are high in the hierarchy, and the specific topics are given lower levels.
3. **Faceted:** In this pattern, the topics are divided into independent facets.
4. **Specialized Classification:** This method is formed for some targeted areas, and some libraries make their own system that give importance to the targeted areas they specialize in.

There are some data warehousing standards and conventions that are being used widely for the resources and imaging. It is the job of the robot to place the book according to the standards and conventions followed by the library. This learning can be done with the help of machine learning techniques which enables the robot to find the existing pattern and also come up with some other patterns which can be further incorporated in the library management system in the near future. Placing of the book demands knowledge of the book already in the stack and then placing it is done with the help of communication between the server and the arm and then arm paces the book in its respective position.

5 Experimentation and Results

Following subsections gives the performance analysis of the implemented LiBOT in the library of DJSCE Mumbai. The working model is tested in the library, and the analysis is done using the few parameters observed during the working.

5.1 Performance

In the proposed LiBOT system, the functionality plays a very important role. It tells how the robot works in different environments. The criticality of these functions tells what the performance and the success of the system are. The model performs optimally even when the stack of books is full as the number of books left on the table is in large numbers. The system is designed in such a way that it works efficiently under various load conditions, book orientations and other parameters and still produces results with same accuracy and efficiency.



Fig. 6 Single book test and multiple-book test using LiBOT

5.2 Accuracy

The book which has been picked up by the LiBOT has to be accurately placed in its respective shelf where it belongs. Thus, it becomes necessary to involve the functionality to judge how accurately the bot is working and what should be done to increase the accuracy. LiBOT observed to work successfully with some approximation. Limitations in the working occur due to hardware efficiency and the trajectory path and its surface in the library.

5.3 Testing

The orientation of books is checked by the system which then decides how to invoke the pick process. This is made sure by the use of the unique JRAM code for proper detection of required book amongst the mess. Figure 6 gives single and multiple-book case successfully handled by LiBOT.

6 Conclusion

The major problem faced by the huge libraries is the issue created by the readers by not placing the books back into the shelf at the correct place. As a result of which it becomes difficult for the other readers to find the book. This issue is going to be solved by the bot developed. Multiple designs were referred for the design of the arm. The most satisfactory design was found to be the PUMA robot arm, which sufficed all the needs as per the tasks. The LiBOT will be built on the basis of the design of

the PUMA robot arm. The unique colour code developed by the group which has a grid of 4×2 cells can take four colours in each cell and will help in uniquely recognizing over 65 thousand books which is much more than what a library has. The entire system is thus designed and structured exactly based on the requirements and is expected to work effectively to give the best results. This research work can be extended to implementing the PUMA robot arm so as to make sure that the bot has a better reach to the books placed on the table.

References

1. Kim BK, Tomokuni N, Ohara K, Ohba K, Tanikawa T, Hirai S (2006) Ubiquitous function services based control of robots with ambient intelligence. In: Proceedings of IEEE international conference on industrial electronics, control, and instrumentation, pp 4546–4551
2. Dias B, Zlot R, Kalra N, Stentz A (2006) Market- based multirobot coordination: a survey and analysis. In: Proceedings of the IEEE, pp 1257–1270
3. Ando N, Suehiro T, Kitagaki K, Kotoku T, Yoon W-K (2005) RT-middleware: distributed component middleware for RT (robot Technology). In: Proceeding of IEEE/RSJ international conference on intelligent robots and systems, pp 3555–3560
4. Chae H, Lee J, Yu W (2005) A localization sensor suite for development of robotic location sensing network. In: Proceeding of 2nd international conference on ubiquitous robots and ambient intelligence, pp 188–191
5. Kim BK, Miyazaki M, Ohba K, Hirai S, Tanie K (2005) Web services based robot control platform for ubiquitous functions. In: Proceeding IEEE international conference on robotics and automation, pp 703–708
6. Browning B, Bruce J, Bowling M, Veloso M (2005) STP: skills, tactics and plays for multi-robot control in adversarial environments. *IEEE J Control Syst Eng* 219:33–55
7. Guo Y, Parker LE (2002) A distributed and optimal motion planning approach for multiple mobile robots. In: Proceedings of IEEE international conference on robotics and automation, vol 3, pp 2612–2619
8. Safaris R, Jezernik K, Calkin DW, Parkin RM (1999) Telerobot control via Internet. In: Proceedings of the IEEE international symposium on industrial electronics, vol 1, pp 298–303
9. Botelho S, Alami R (1999) M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In: Proceedings of the IEEE international conference on robotics and automation, pp 1234–1239
10. Finin T, Labrou Y, Mayfield J (1995) KQML as an agent communication language. In: Bradshaw J (ed) Software agents. MIT Press, Cambridge
11. Hansson R (1995) Industrial robot lends a hand in a Swedish library. *ABB Rev* 3:16–18
12. NCBI Homepage. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4212968/>. Last accessed 15 Nov 2019
13. NCBI Homepage. <http://micsymposium.org>. Last accessed 15 Nov 2019

Chapter 6

IndoorNet: Generating Indoor Layouts from a Single Panorama Image



Yash Kotadia, Krisha Mehta, Mihir Manjrekar and Ruhina Karani

1 Introduction

The increased interest in the field of 3D layout reconstruction can be attributed to its innumerable applications in field of robotics, indoor navigation [1, 2], augmented reality [3] and gaming [4]. In this paper, we aim to represent the details of a room in the 3D space as accurately as possible as shown in Fig. 1. For this, we use a panorama that helps capture maximum information of the room due to its large field-of-view (FoV). Capturing a single panorama also simplifies the process for the user using the model. Our model works in the following way as shown in Fig. 2. We use a panorama image of the room as input. This image is then used to identify the vanishing points and generate Manhattan lines based on the “Manhattan world” assumption [5]. The input image along with Manhattan lines is fed to a CNN with an encoder–decoder arrangement that predicts the corner map and boundary map of the room. We find that accurate prediction of corners and boundaries is extremely crucial for an accurate layout generation for a room. We also find that using a single encoder–decoder arrangement prevents the model from learning redundant features of the image. The resultant parameters are then optimized using the gradient descent optimizer to minimize loss. In the following sections, we will describe our algorithm in greater detail as well demonstrate a comparative study we have drawn by developing different variants of our model.

Y. Kotadia · K. Mehta (✉) · M. Manjrekar · R. Karani
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: krisha.mehta@djsce.edu.in

Y. Kotadia
e-mail: yash.kotadia@djsce.edu.in

M. Manjrekar
e-mail: mihirmvm5@gmail.com

R. Karani
e-mail: ruhina.karani@djsce.ac.in



Fig. 1 Examples of indoor layouts generated

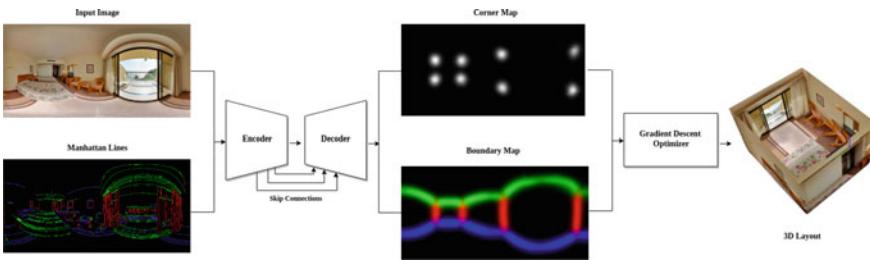


Fig. 2 IndoorNet Architecture: IndoorNet uses an encoder–decoder arrangement. A panorama image is provided as input. This input is used to generate an image of Manhattan lines. Both these images are fed to the encoder–decoder which produce the corner map and boundary map. These maps, after optimization, are used to generate the 3D layout

2 Related Work

Layout generation of indoor scenes can be performed in several ways. Main ways to differentiate between these approaches include the number of images layout generation, use of perspective or panoramic images, the shape of the room, and use of geometry. With the recent development of deep learning methods, there has been a corresponding development in the techniques used for semantic segmentation [6–9]. Mallya et al. [10] trained a fully convolutional neural network (FCNN) to predict informative edge maps for layout prediction as opposed to traditional methods that make use of geometric context and orientation maps. On the other hand, Dasgupta et al. [11] used FCNN to label layout surfaces like walls, floors, and ceiling. But these approaches fail to exploit the end-to-end learning ability provided by deep convolutional neural networks (CNN). RoomNet [12] is one of the first ones to exploit the functionality by using perspective images to generate the room layout and the

corresponding segmentation, completely specified given the locations of an ordered set of key points. PanoContext [13] uses a single panorama image to capture the whole room context. PanoContext improves upon the framework used for perspective images by generating a set of room layouts and composing them into many whole room hypotheses. Search for the best whole room hypotheses for output is done using a scoring system. LayoutNet [14], as well as our work, is more direct. Layouts from panoramic images with geometry and deep learning [15] and PanoRoom [16] use a combination of deep learning and geometry to predict room layouts. Meanwhile, Zhao et al. [17] develop physics- inspired optimization as a new inference scheme, which is inspired by mechanics concepts to estimate room layouts. Our work is very similar to LayoutNet that uses a single panorama image along with Manhattan lines to generate the corner map and the edge map. However, our approach is faster and smaller in size, without compromising on the accuracy. We use a single encoder-decoder pair to predict both the maps simultaneously. Our model does not make use of a 3D layout parameter regressor either as no significant improvement is observed while using it. With respect to optimizers, while LayoutNet uses Manhattan layout optimizer, we observed our method provides better results.

3 Approach

See Fig. 3.



Fig. 3 Qualitative results for layout prediction based on the PanoContext dataset [13]: We compare our model IndoorNet with two previous approaches PanoContext [13] and LayoutNet [14]. Each image consists predicted layout from given method (orange lines) and ground truth layout (green lines). Best viewed in color

3.1 *Capturing the Input Image*

The aim of this paper is to build a system that presents the maximum details of the target room. Keeping this in mind, we impose constraints on the input image to be captured such that it encompasses as many details of the target room as possible. To this extent, as proposed by [13, 14], we use 360 HFOV panoramic images under the equirectangular projection that ensures complete coverage of the room. Such images can now be easily obtained using applications such as Google Street View. The extended advantage of such an input would be the rich contextual information captured. Contextual information aid in mapping the walls and floor of the target room by making use of information such as the position of chairs would be on the floor and that of a fan would be on the ceiling.

3.2 *Calculating the Manhattan Lines*

Manhattan lines [5] are a representation of the lines in the target room image that defines the axis x , y , or z of the 3D layout to which the line belongs. As proposed by [14], we first divide the panoramic image into multiple overlapping perspective images. Next, select long lines and warp the line segments back to the panorama. We then calculate all possible vanishing directions using Hough transform and finally each line segment votes to select three mutually perpendicular vanishing directions. This representation of lines along with its 3D direction serves as a useful feature.

3.3 *Network Architecture*

We propose a fully convolutional network that predicts the corner probability map and the boundary probability map. Similar to [14], we use an encoder-decoder scheme, however, with a more productive variation.

FCN Encoder The input to the encoder is the 3-channel image concatenated with its 3-channel Manhattan line map. The resolution of the input is 512×1024 , and the Manhattan lines are calculated using the method described in Sect. 3.2. The encoder shares specifications as proposed by [14]. It contains seven convolutional layers. Each convolution layer consists of the following sequence of operations, kernel of size 3×3 , ReLU activation, and finally max pooling with down factor of 2. The initial convolution layer outputs a feature map with 32 channels, while the deepest layer outputs a feature map with 2048 channels. We have also explored a lighter model with 16 channels in initial layer and 1024 channels in the deepest layer. The performance of both methods is evaluated in Sect. 5.

FCN Decoder The decoder is trained to learn filters that can output the corner probability map (P_c) and the boundary probability map (P_b). The filters required to

generate the boundary map and the corner map from the high-dimensional representation of the room are naturally going to be similar. As opposed to [14] which uses two separate decoders, one for the boundary map and other for corner map, we propose to use a single decoder for both. Thus, our approach circumvents the inefficiencies of training redundant filters and is both lighter and faster. The 1024 or 2048 high-dimensional feature vector is upsampled seven times, each time followed by 3×3 kernel and ReLU. The final layer outputs the 3 channel boundary probability map and 1 channel corner probability map. Inspired by U-net [8], we add skip connections from the encoder to the decoder. This allows passing feature information to lower layers so that it can infer upon tiny details for more accurate predictions.

Loss Function The loss function is calculated as shown in Eq. 1:

$$L(P_b, P_c) = - \sum_{\bar{y}_b \in P_b} [y_b \cdot \log \sigma(\bar{y}_b) + (1 - y_b) \cdot \log(1 - \sigma(\bar{y}_b))] \\ - \sum_{\bar{y}_c \in P_c} [y_c \cdot \log \sigma(\bar{y}_c) + (1 - y_c) \cdot \log(1 - \sigma(\bar{y}_c))] \quad (1)$$

The network minimizes the binary cross-entropy loss of the predicted boundary map and corner map. In Eq. 1, \bar{y}_b is the predicted probability of a pixel in the boundary map and y_b is the corresponding ground truth. Similarly, \bar{y}_c is the predicted probability of a pixel in the corner map and y_c is the corresponding ground truth. We use binary cross-entropy loss instead of L_2 loss for corner prediction owing to the improved results as demonstrated by [14].

3.4 Gradient Descent Optimizer

The room is assumed to have a cuboid layout. We extract the cuboid parameters from the predicted boundary map and corner map. The 3D parameters are (l, w, h, dx, dz, y) where (l, w, h) are the dimensions of the cuboid, (dx, dz) is the translation of the floor in x -axis and z -axis, and y is the rotation of the floor with respect to the y -axis. Points are sampled along the boundary of the predicted cuboid which is then projected to the initial equirectangular image. The error is calculated as the L_2 distance between the projected points and the original points. Finally, to optimize the results, we apply gradient descent on the calculated error with respect to the cuboid parameters (Fig. 4).



Fig. 4 Qualitative results (randomly sampled) on the Stanford 2D-3D annotation dataset. The Stanford dataset is more complex due to a more narrow field of view. Here, orange lines show IndoorNet’s layout prediction while green lines show the ground truth. Best viewed in color

4 Experiments

4.1 Datasets

We evaluate our approach on two benchmark datasets.

PanoContext DataSet Zhang et al. provided with the PanoContext dataset [13] that consists of panorama images with annotated corners and boundaries.

Stanford 2D-3D Annotation Dataset The Stanford 2D-3D dataset [18] is more complex due to a more narrow field of view. It also consists of more clutter in the room as compared to other datasets. Hence, we use a total of 1000 images and split them as 75–10–15% for training, validation, and testing, respectively. To ensure that the model receives enough number of images as well as views images clicked in different conditions, we employ multiple augmentation techniques (Table 1).

Table 1 Comparative study of LayoutNet and IndoorNet

Test Dataset	Method	3D IoU (%)	CE (%)	PE (%)
PanoContext	LayoutNet	75.12	1.02	3.18
PanoContext	Ours	76.26	0.98	3.07
Stanford 2D-3D	LayoutNet	77.51	0.92	2.42
Stanford 2D-3D	Ours	79.43	0.9	2.73

Bold indicates best results

4.2 Results

We evaluate our model on three metrics: (1) Corner Error (CE): It gives the accuracy of the predicted corners. (2) Pixel Error (PE): Evaluates the pixel surface error across the ceiling, floor, and walls. (3) 3D Intersection over Union (3D IoU): Calculates the intersection of the predicted surfaces over their union between the ground truth and predicted layout. When trained and tested on the PanoContext dataset, our 3D IoU accuracy is 74.88% while the corner error is 1.05%. When trained on both the datasets, and tested on the PanoContext dataset, our 3D IoU accuracy is 76.26%. When tested on the Stanford dataset, our 3D IoU accuracy is 79.43%. On both the datasets, we perform better than state of the art (Tables 2 and 3).

4.3 Accuracy

When trained on the PanoContext dataset, our model performs as shown in Table 2. Zou et al. [14] used double decoders with skip connections between the corner decoder and boundary decoder followed by a post-optimization scheme. The proposed approach does marginally better than LayoutNet when both trained and tested on PanoContext Dataset. Our model performs much better when trained on the combination of both datasets as seen in Table 1. The increase in performance can be attributed to the use of gradient descent optimizer, decrease in complexity, and number of redundant features learned by the model (Table 4).

On PanoContext, the 3D IoU for LayoutNet is 74.48% while for IndoorNet it is 76.26%. On the Stanford 2D–3D dataset, the 3D IoU for LayoutNet is 75.12% and

Table 2 Performance on Panocontext

Method	3D IoU (%)	CE (%)	PE (%)
PanoContext	67.23	1.60	4.55
LayoutNet	74.48	1.06	3.34
Ours	74.88	1.05	3.07

Bold indicates best results

Table 3 Analysis of model size

Dataset	Model (size)	3D IoU (%)	CE (%)	PE (%)
PanoContext	IndoorNet (2048)	76.26	0.98	3.07
PanoContext	IndoorNet (1024)	74.91	0.98	3
PanoContext	IndoorNet (4096)	74.75	1.25	3.34
Stanford 2D-3D	IndoorNet (2048)	79.43	0.9	2.73
Stanford 2D-3D	IndoorNet (1024)	73.99	1.03	3.76
Stanford 2D-3D	IndoorNet (4096)	77.79	1.13	3.33

Bold indicates best results

Table 4 Ablation study of Manhattan lines

Dataset	Model	3D IoU (%)	CE (%)	PE (%)
PanoContext	w/ Manhattan lines	76.26	0.98	3.07
PanoContext	w/o Manhattan lines	74.19	1.13	3.71
Stanford 2D-3D	w/ Manhattan lines	79.43	0.9	2.73
Stanford 2D-3D	w/o Manhattan lines	73.18	1.29	3.63

Bold indicates best results

Table 5 Ablation study of gradient descent optimizer

Dataset	Model	3D IoU (%)	CE (%)	PE (%)
PanoContext	w/ gradient descent optimizer	76.26	0.98	3.07
PanoContext	w/o gradient descent optimizer	75.53	1.07	3.35
Stanford 2D-3D	w/ gradient descent optimizer	79.43	0.9	2.73
Stanford 2D-3D	w/o gradient descent optimizer	76.41	0.97	3

Bold indicates best results

Table 6 CPU runtime performance

Method	Approx. CPU time (s)
PanoContext [13]	>300
LayoutNet [14]	44.73
Ours ^a	$11.72 + 4.23 + 3.4 + 2 = 21.35$

^aThe runtime for other methods is evaluated on Intel Xeon 3.5GHz (6 cores CPU). Ours is computed on Intel Xeon 2.2GHz (1 core)

that for IndoorNet is 79.43% Our proposed gradient descent optimization approach gives a $15 \times$ speedup as compared to the optimization scheme proposed by [14]. The proposed gradient descent optimization scheme takes 2s as opposed to 30s on CPU taken by “Manhattan Layout Optimization” scheme of LayoutNet.

4.4 Runtime and Complexity

Post-optimization takes up the majority of the processing time for LayoutNet [14]. Our model, on the other hand, is much faster. As shown in Table 6, using only CPU with PyTorch, the total processing takes less than 24s, of which optimization takes 2s (Table 5).

5 Discussion

For an in-depth study, we developed different variants of IndoorNet which include (i) ablation study of Manhattan lines, (ii) ablation study of optimization, (iii) variation in model size.

Table 4 shows the performance of our model improves significantly when Manhattan lines are used. This can be attributed to the fact that Manhattan lines help provide geometric context to the CNN which helps predict more accurate layouts.

We also implement a gradient descent optimizer. Table 5 shows how using the optimizer helps improve the 3D IoU accuracy by more than 3% in the Stanford 2D-3D dataset and by more than 0.75% in the PanoContext dataset.

We evaluate models with varying number of channels in the first layer and the last layer of encoder (a) 16-1024 (b) 32-2048 (c) 64-4096. As shown in table 3, the performance largely improves by increasing the model size from (a) to (b); however, it drops on further increasing the model size from (b) to (c). It means that from (a) to (b), the model learned useful filters that improved the accuracy; however, from (b) to (c), it started overfitting and hence accuracy decreased.

6 Conclusion

We propose IndoorNet, an indoor layout generation algorithm using a single panorama image. Our approach has higher accuracy as compared to previous works while being faster and significantly lighter. The post-processing optimization strategy proposed significantly increases the efficiency of the pipeline. Future work includes developing an approach that benefits from the global context for more accurate results.

References

- Mirowski P, Pascanu R, Viola F, Soyer H, Ballard AJ, Banino A, Denil M, Goroshin R, Sifre L, Kavukcuoglu K, Kumaran D, Learning to navigate in complex environments. arXiv preprint [arXiv:1611.03673](https://arxiv.org/abs/1611.03673)
- Savva M, Chang A, Dosovitskiy A, Funkhouser T, Koltun V (Dec 2017) MINOS: multimodal indoor simulator for navigation in complex environments. [arXiv:1712.03931 \[cs\]](https://arxiv.org/abs/1712.03931)
- Xiao J, Furukawa Y (2014) Reconstructing the world's museums. Int J Comput Vis 110(3):243–258
- Tuttenel T, Bidarra R, Smelik R, De Kraker K (2009) Rule-based layout solving and its application to procedural interior generation. In: CASA workshop on 3D advanced media in gaming and simulation
- Coughlan J, Yuille A (1999) Manhattan World: compass direction from a single image by Bayesian inference. In: Proceedings of the seventh IEEE international conference on computer vision
- Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495

7. Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille A (2018) DeepLab: semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* 40(4):834–848
8. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: Computer science medical image computing and computer-assisted intervention—MICCAI, pp 234–241
9. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE conference on computer vision and pattern recognition
10. Mallya A, Lazebnik S (2015) Learning informative edge maps for indoor scene layout prediction. In: 2015 IEEE international conference on computer vision (ICCV), Santiago, Chile, pp 936–944
11. Dasgupta S, Fang K, Chen K, Savarese S (2016) DeLay: robust spatial layout estimation for cluttered indoor scenes. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA
12. Lee C, Badrinarayanan V, Malisiewicz T, Rabinovich A (2017) RoomNet: end-to-end room layout estimation. In: 2017 IEEE international conference on computer vision (ICCV), Venice
13. Zhang Y, Song S, Tan P, Xiao J (2014) PanoContext: a whole-room 3D context model for panoramic scene understanding. In: Computer vision ECCV 2014 lecture notes in computer science, pp 668–686
14. Zou C, Colburn A, Shan Q, Hoiem D (2018) LayoutNet: reconstructing the 3D room layout from a single RGB image. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, Salt Lake City, UT
15. Fernandez-Labrador C, Perez-Yus A, Lopez-Nicolas G, Guerrero J (2018) Layouts from panoramic images with geometry and deep learning. *IEEE Robot Autom Lett* 3(4):3153–3160
16. Fernandez-Labrador C, Facil J, Perez-Yus A, Demonceaux C, Guerrero J (2018) PanoRoom: from the sphere to the 3D layout. [arXiv:1808.09879\[cs\]](https://arxiv.org/abs/1808.09879)
17. Zhao H, Lu M, Yao A, Guo Y, Chen Y, Zhang L (2017) Physics inspired optimization on semantic transfer features: an alternative method for room layout estimation. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI
18. Armeni I, Sax S, Zamir A, Savarese S (2017) Joint 2D-3D-semantic data for indoor scene understanding. [arXiv:1702.01105](https://arxiv.org/abs/1702.01105)

Chapter 7

Lightweight Random Number Generation for Elliptic Curve Cryptography for Use in IoT



Aruna Gawade and Rushabh Vinchhi

1 Introduction

The development in the IoT sector has been phenomenal and is now being increasingly applied in fields beyond academics such as smart grid [1], e-health [2], and e-home [3]. According to Ning et al. [4], and the architecture they proposed, suggest that all the nodes involved in the IoT system require increased data gathering and sharing between them. However, along with all the required data distribution, one expects high requirements of data security. This perspective presents one of the most challenging aspects of IoT.

One of the most common ways and also the most secure way for maintaining data security is by using the Advanced Encryption Standard (AES). However, in embedded IoT devices, which are severely resource and memory constrained, the use of AES can take up as much as thirty percent of RAM and makes inefficient use of the ROM available. Also the use of exponent-based Diffie Hellman or RSA proves very expensive given the modular operations involved in both [5]. Alternative methods of key generation and its exchange must, thus, be looked at, which minimize the resource usage and at the same time, maintain a certain expected level of security. In this paper, the use of elliptic curve cryptography Diffie Hellman (ECCDH) is suggested since ECC algorithm has much stronger bit security than RSA as well as other exponential-based public key cryptographic algorithm, and it is easy to be realized on hardware or a chip. Since it replaces the expensive bilinear pairing operation with point scalar multiplication on elliptic curve, it can meet the lightweight requirement and is suitable for IoT [6].

A. Gawade · R. Vinchhi (✉)
Dwarkadas J Sanghvi College of Engineering, Vile Parle (west),
Mumbai 400056, India
e-mail: rushabh.vinchhi@djsce.edu.in

A. Gawade
e-mail: aruna.gawade@djsce.ac.in

True random and pseudorandom number generators (TRNG and PRNG, respectively) are two of the most important building blocks of cryptosystems. They are used to generate confidential keys, challenges, and nonces. In the constrained devices of the IoT applications, cryptographically secure PRNGs are difficult to attain due to hardware or software limitations.

In this paper, the use of a lightweight PRNG is introduced for its subsequent use in the ECC system. The rest of this paper is organized as follows. In Sect. 2, the most important published works about the use of PRNG and ECC in constrained devices are reviewed. Our proposed hybrid system that combines lightweight random key generation for the ECC system is then presented in depth in Sect. 3. The characteristics of its and some results are discussed in Sect. 4. Conclusions are presented in Sect. 5.

2 Related Work

Umpteen ECC-based communication protocols have been researched and proposed to resolve the challenges within IoT devices.

Kalra et al. [7] put forth a method for multiserver authentication that was based on ECC. Xuanxia et al. [5] also put forth the use of attribute-based ECC that uses one or more attributes of the IoT node for public key generation and subsequent encryption. However, this method generally slows down the overall encryption process and also does not provide enough flexibility. A number of signature generation and authentication algorithms over ECC have been proposed by Pessl et al. [8] and He et al. [9]. Dhillon et al. [10] in their paper have proved through their findings that ECC fits well with real-time IoT devices, with constrained environments. Goyal [14] proves beyond doubt the superiority of ECCDH over exponent-based DH and RSA algorithms. Also, the generation of a shared secret key lowers down the key generation costs for subsequent communications between the two nodes. Thus, in this paper, the use of ECCDH is used for key generation for communication between two nodes.

Random number generators for generating the private key on resource-constrained device is a memory as well as power-consuming task because of the sheer number of rounds a random number generator takes which to achieve an expected level of security. Several works have been done in the lightweight category for random number generation. In [11], The Warbler PRNG for low cost smart devices was proposed that was based on nonlinear feedback shift register. However, the entire family of this PRNG is susceptible to cryptanalysis and not secure enough. A PRNG called LAMED was discussed in [12] for RFID application in wireless sensor devices. Although Lopez proposed the use of the additive lagged Fibonacci random number generator (ALFG) in his paper due to its inherent advantage against the PRNG's discussed below. The paper implements all the above discussed PRNGs and compares their respective hardware complexities to then choose ALFG has the most obvious choice for resource-constrained IoT device. The findings can be summarized using the table shown below in Table 1.

Table 1 Comparison between PRNGs

PRNG	Key Size (Bits)	Hardware Complexity
Lamed [11]	32	1585
Warbler [12]	32	1238
ALFG[13]	128	1402

Table 2 Comparison between algorithms

Algorithm	Power Consumption (mW)	Delay (ms)
ECCDH	0.570	22.4
DH	0.832	49.6

In this paper, as part of the hybrid lightweight encryption system, based on the above findings, I propose the use of the PRNG based on lagged Fibonacci generator. With the use of this generator, computational costs are saved while maintaining a level of security, as validated by the number of statistical tests conducted on it [13].

Having selected a lighter algorithm, we further need to select a key exchange mechanism to establish a secret key between two communicating parties. Two obvious choices for key exchange are modular Diffie Hellman (DH) and elliptic curve-based Diffie Hellman (ECDH). The ECDH algorithm has found much favor with IoT devices due to its robustness, added security and low power consumption. It also offers an inherent computational advantage over DH as it does not make use of modular arithmetic. Goyal et al. [14] have been able to summarize the advantages of using ECDH over DH well and the following table in Table 2 further substantiates the claim.

Thus in the next section, we propose a hybrid scheme that makes use of the ALFG PRNG to generate a private key for its further use in our chosen algorithm—ECDH.

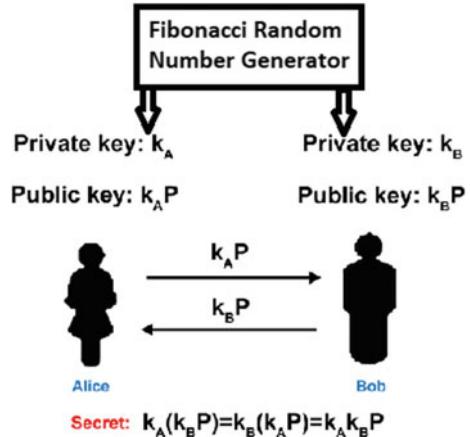
3 Scheme Description

This section walks the reader through the proposed hybrid scheme that makes use of elliptic curve cryptography system along with lightweight PRNG. Consider two IoT nodes Alice and Bob who wish to communicate with each other (Fig. 1).

3.1 Setup

Cryptography is transformation of plain message to make them secure and immune from intruders. Elliptic curve cryptography (ECC) is a public key cryptography developed independently by Victor Miller and Neal Koblitz in the year 1985. In elliptic curve cryptography, we will be using the curve equation of the form

Fig. 1 Scheme



$$y^2 = x^3 + ax + b$$

which is known as Weierstrass equation, where a and b are the constant with

$$4a^3 + 27b^2 \neq 0.$$

3.2 Key Generation

The generator G is defined over the elliptic curve that is used to generate public key and private key for Alice and Bob. This G is determined by the authority center within the IoT network and then communicated to all other devices within the network such that both Alice and Bob agree over a common elliptic curve and G .

Further, let nA and nB be private keys of Alice and Bob, respectively. Normally, these private keys are generated randomly using PRNG or TRNG, but in this hybrid model, we propose the introduction of Fibonacci PRNG, that generates private keys that are random but computationally less heavy. The lightweight private key generation using PRNG is shown as follow.

$$S_n = S_{n-1} + S_{n-2}$$

The above relation depicts Fibonacci sequence. To produce a random number, the above recurrence relation can be generalized:

$$S_n \equiv S_{n-j} \star S_{n-k} \bmod m, 0 < j < k$$

The operator denotes any binary operation. These generators also have an underlying generating polynomials.

Algorithm 1: ALFG

```

Result: key
lag1, lag2  $\leftarrow$  primepair ; modulo  $\leftarrow$  bitsize ; S  $\leftarrow \phi[lag2]$  ; array  $\leftarrow S$ 
if seed  $\leq$  lag1 then
| key  $\leftarrow$  ALFG(seed - lag1) + ALFG(seed - lag2) mod modulo
else
| key  $\leftarrow$  data[seed];
end
key  $\leftarrow$  PKCS(key)

```

The initialization vector of the above proposed PRNG could be counter based on the internal clock of the IoT node. Once nA and nB are both found using lightweight random number generator, public keys kA and kB are generated using the formula given below.

$$kA = nA \cdot G$$

$$kB = nB \cdot G$$

These public keys are then shared between Alice and Bob. The security of ECC is based upon a hard number theoretic problem called elliptic curve discrete logarithms (ECDLP), which means that it is hard to find k such that $Q = k \cdot P$ for a given elliptic curve and points P and Q on the curve. The hardness of ECDLP defines the security level of all ECC protocols, and no sub-exponential algorithm which can solve the ECDLP is known. Thus, both set of private keys and public keys are thus determined.

3.3 Generation of Shared Secret Key

Alice computes point

$$(x_k, y_k) = nA.kB$$

Bob computes point

$$(x_k, y_k) = nB.kA$$

It turns out the point calculated by both is the same and is the shared secret key.

Table 3 Comparative analysis

Parameter	Proposed PRNG	Existing PRNG
Key Generated	@326de728	[B@6e1ec318
Time (ns)	47,028,957	261,192,004
Memory Consumption (bytes)	29,464	29,3616

4 Characteristics and Results

In this section, the proposed random number generator is tested against the existing elliptic key generation technique. The implementation is based on Java and uses existing Sun security packages available. We also make use of existing elliptic curve cryptography packages that SHA1 secure pseudo random number generators. Effectively, this leaves the comparison between lagged Fibonacci generator and SHA1PRNG, the default secure random number generator, supplied by Sun provider. Table 3 shows a comparison between the two based on different parameters for a standard output key of 128 bits.

On the computational front, the proposed system provides considerable advantages. It proves to be both faster and memory efficient than the existing system. This makes it suitable for use in lightweight devices for key generation that are memory and resource constrained. On the security front, the additive lagged Fibonacci generator is susceptible to brute-force attacks. The linear complexity of the least significant bits of the words of a conventional LFG is equal to the value of the lag r ; the linear complexity of the successive significant bits increases progressively, reaching a value close to $2^{(N-1)}(2^r - 1)$ for the most significant bits. If two conventional LFGs, with lags r_1 and r_2 with bitwise addition, the linear complexity of the bits of the resulting sequence is nearly equal to the sum of the respective complexities of the bits of both LFGs. This fact allows the easy determination of several of the least significant bits of the seed, thus simplifying a brute-force attack for the complete determination of the seed [13].

5 Conclusions

A hybrid crypto-system consisting of lightweight, fast, and secure pseudo random number generator as a feed to generate private keys in ECC is proposed. The resulting system is thus computationally efficient in resource-constrained environments. At the same time, there is scope to improve it further to make it more secure against attacks. The above observations make this hybrid system suitable for IoT applications.

References

1. Yun M, Yuxin B (2010) Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. In: Advances in energy engineering, ICAEE, pp 69–72
2. Ali ST, Sivaraman V, Ostry D (2014) Authentication of lossy data in body-sensor networks for cloud-based healthcare monitoring. Future Gener Comput Syst 35:80–90
3. Dlodlo N (2012) Adopting the Internet of Things technologies in environmental management in South Africa. In: Proceedings of the international conference on environment science and engineering, Singapore, vol 3, pp 45–55
4. Ning H, Liu H (2013) Cyberentity security in the internet of things computer. IEEE Comput Soc 46(4):46–53
5. Sehgal A, Perelman V, Kuryla S, Schonwalder J (2012) Management of resource constrained devices in the internet of things. IEEE Commun Mag 50(12):144–149
6. Yao Xuanxia, Chen Zhi, Tian Ye (2015) A lightweight attribute-based encryption scheme for the internet of things, future generation computer systems. Elsevier, Amsterdam
7. Kalra S, Sood S (2013) Advanced remote user authentication protocol for multi-server architecture based on ECC. J Inf Secur Appl 18(2–3):98–107
8. Pessl P, Hutter M (2014) Curved tags a low-resource ECDSA implementation tailored for RFID, radio frequency identification: security and privacy issues lecture notes in computer science, pp 156–172
9. He D, Zeadally S (2015) An analysis of RFID authentication schemes for internet of things in healthcare environment using elliptic curve cryptography. IEEE Int Things J 2(1):72–83
10. Dhillon PK, Kalra S (2016) Elliptic curve cryptography for real time embedded systems in IoT networks. In: 2016 5th international conference on wireless networks and embedded systems (WECON), Rajpura, pp 1–6
11. Mandal K, Fan X, Gong G (2016) Design and implementation of Warbler family of lightweight Pseudorandom number generators for smartdevices. ACM Trans Embedded Comput Syst 15(1):1–28
12. Peris-Lpez P, Hernndez-Castro JC, Estvez-Tapiador JM, Ribagorda A (2009) LAMEDA PRNG for EPC class-1 generation-2 RFID specification. Comput Standards Interfaces 31(1):88–97
13. Ore Lpez AB, Hernndez Encinas L, Martn Muoz A, Montoya Vitini F (2017) A lightweight Pseudorandom number generator for securing the internet of things. IEEE Access 5:27,800–27,806. <https://doi.org/10.1109/ACCESS.2017.2774105>
14. Goyal TK, Sahula V (2016) Lightweight security algorithm for low power IoT devices. In: 2016 international conference on advances in computing, communications and informatics (ICACCI), Jaipur, pp 1725–1729. <https://doi.org/10.1109/ICACCI.2016.7732296>

Chapter 8

Improvement of Lightweight Integrity Verification Algorithm Using TDHA



Anushka Gangwal, Dhyey Mehta, Soham Khedekar and Aruna Gawde

1 Introduction

Integrity is one of the most crucial aspects of computer security. The main idea behind integrity is the maintenance and assurance of the uniformity and correctness of data when it is transmitted from the sender to the receiver. It is the quality of information to remain unaltered and untampered between transmission, storage and usage. Data can be exposed to compromise in any of these three phases. Thus, various techniques are used to ensure that integrity is maintained and that any breach of integrity can be immediately detected by the concerned entities. The most commonly used methods are using digital certificates and hashing algorithms.

Digital signatures are claimed to be too expensive to be feasible extensively. Even some of the existing hashing algorithms are unaffordable as a result of being computationally pretty expensive due to a large number of rounds through which data passes before it is rendered secure. This is where the factor of weight comes into the picture. Lightweight integrity algorithms, in effect, make the process of maintaining integrity more cost-friendly, computation-friendly and, thus, security-friendly. In this paper, we explore lightweight alternatives to the general integrity algorithms, namely MD5 algorithm, Digital Signature Algorithm and RSA algorithm. We will

A. Gangwal · D. Mehta · S. Khedekar (✉) · A. Gawde

Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai,
India

e-mail: sohamsk97@gmail.com

A. Gangwal

e-mail: anushkagangwal10@gmail.com

D. Mehta

e-mail: dhyeymehta04@gmail.com

A. Gawde

e-mail: aruna.gawade@djsce.ac.in

first present an overview of all the lightweight algorithms, and at the end, we propose certain alterations that could make them even more lightweight and efficient than they are.

2 Literature Survey and Overview of Algorithms

The literature survey divides the algorithms that are to be discussed into three domains: Integrity verification algorithms, hash algorithms and signature algorithms.

2.1 *Integrity Verification Algorithms*

Data integrity verification allows clients to know whether the data in the system is secure or not. The algorithms mentioned below are used to maintain the integrity of the data that is being distributed among nodes.

2.1.1 RSA

The National Bureau of Standards (NBS) algorithm was replaced by a more secure cryptographic algorithm called RSA. It implements a public-key cryptosystem as well as digital signatures. Public-key encryption removes the need for an external entity to deliver keys over another secure channel before transmitting the original message. In RSA, encryption keys are public, unlike the decryption keys, and hence, in the absence of a correct decryption key, an encrypted message cannot be deciphered. The keys are designed such that the decryption key cannot be obtained from a given public key.

Digital signatures come in handy when a transmitted message has to be verified to check whether it actually originated from the sender. The sender's decryption key is used for this, and the public encryption key can be used to verify the signature. This avoids forging. Further, it takes away possible deniability from the sender.

There have been no successful attempts at breaking RSA because it is very difficult to factor large prime numbers. The security of RSA is so far intact.

2.1.2 LIVE

Some of the most popular integrity algorithms of today are Digital Signature Algorithm (DSA) and RSA. Both of these algorithms can be very slow in cases where large data needs to be encrypted by the same computer. The processes of key generation and subsequent encryption can introduce significant computation overhead. To better the implementation of these verification algorithms, a new architecture was proposed

in [1]. It is called lightweight integrity verification (LIVE). It is an extension of the mainstream algorithms used today.

Inspired by one-time signature algorithms [2, 3], LIVE extends hash functions to generate tokens and produce and verify signatures. The first step is token generation and its distribution. LIVE generates a random number and a hash algorithm such as SHA-1 to generate the token. Once this is done, the content is signed using the one-time signature algorithm and the tokens. The content is encrypted and sent over to the receiver. The receiver decrypts the message using a public key and carries out content verification using the OTS algorithm. The LIVE algorithm is as follows (Fig. 1):

By introducing altered schemes for encryption and signing, LIVE reduces the delay of traditional public-key signature schemes by over 20 times. [1] shows that LIVE only incurs average 10% delay in accessing contents (Table 1).

Performance of RSA and LIVE compares as follows according to [1].

2.2 Hash Algorithms

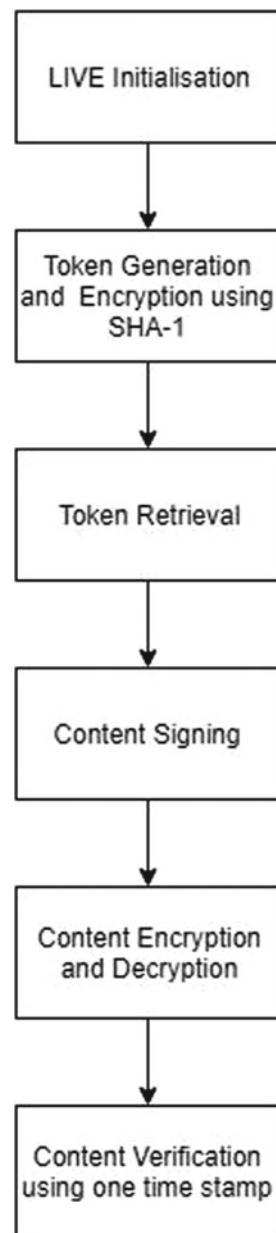
A function that takes some message of any length as input and transforms it into a fixed-length output is called a hash function. This output could be a hash value, a message digest, a checksum or a digital fingerprint. Ronald Rivest developed MD5 in 1991 [4] and it has since become one of the most widely used hashing algorithms. MD5 is an improved version of MD4. It produces a fixed length of hash values. Yet, MD5 is almost deprecated and it should not be used for sensitive projects. MD5 is vulnerable to brute force attacks and can be cracked in hours, or even in seconds depending on the hardware you use. Hence, we turn towards better lightweight alternatives such as the Secure Hash Algorithm (SHA), the lightweight one-way cryptographic hash algorithm (LOCHA) and timestamp defined hash algorithm (TDHA).

2.2.1 SHA

Created by the NIST in 1993 [5], SHA-1 is similar to MD5 with a 160-bit message digest. SHA-1, when introduced, was very time efficient and robust. However, it is not used for most cryptographic uses anymore, after a 2010 attack by created by Marc Stevens, which was able to produce hash collisions on SHA-1 with a complexity of 2⁶¹ operations [6].

SHA-2 was formulated by the NSA [7]. It has two hash functions, namely SHA-256 (uses 32-bit words) and SHA-512 (uses 64-bit words). SHA-2 is more difficult to crack as compared to SHA-1, but it is not as time efficient as the SHA-1 algorithm.

Secure Hash Algorithm 3 (SHA-3) is the latest Secure Hash Algorithm, released by NIST in 2015 [8]. SHA-3 is the latest addition to the SHA series, yet unlike the previous two versions, SHA-3 varies internally. SHA-3 uses sponge construction [9], wherein data is “absorbed” into a sponge, and the result is “squeezed” out of

Fig. 1 Traditional LIVE**Table 1** Performance of integrity verification algorithms [1]

Algorithm	Computation overhead (in terms of time) (μ s)	
	Content signing	Content verification
RSA	14,133.90	10,186.40
LIVE	439.4	386.1

it. During absorption, a permutation function f is used to transform subsets of the state. These subsets are essentially XORed message blocks. In the “squeeze” phase, the same process is carried out, but the permutation function is replaced by a state transformation function.

SHA-3 and SHA-2 are likely to give much better results when compared to SHA-1. SHA-2 and SHA-3 have not been shown to be susceptible to the attacks that SHA-1 is vulnerable to. SHA-3 is not vulnerable to length extension attacks either, which affect all M-D hashes like MD5, SHA-1 and SHA-2.

2.2.2 LOCHA

The lightweight one-way cryptographic hash algorithm (LOCHA), a lightweight hashing scheme, was proposed in [10]. A short, fixed hash digest is laid down and created using the input message of random length by this scheme.

In LOCHA, input pre-processing is done by translating the ASCII codes of constituent characters to their respective binary representation. Unambiguous padding is also employed at the least significant position of the input. This makes the input a multiple of 512. Once it is established that the input is divisible by 512, 512 extra 0s are added to make the algorithm more robust. Now, a nested approach is used to split the pre-processed message thrice. The first split produces blocks that are 512-bits each, the second split produces blocks that are 64-bits each and the third split produces blocks that are 8-bits each.

Following this, three successive conversions are carried out on the processed message. Blocks of 512-bits are divided into eight blocks of 64-bits each. This 64-bit block then generates eight blocks of 8-bits each. A prime number is then selected from the first s-table and is used to substitute each of the 8-bit blocks. Further, these 8 substituted values are used to generate a number for each 64-bit block. This number is the output of the first conversion.

The second conversion uses the second s-table. To maintain consistency in conversion and to decrease overheads, the values of the second s-table are randomly generated. The third conversion uses the outcomes of the first two conversions. Here, the result is converted into a 3-digit hexadecimal value. In a 3-level exchange of each 512-bit block, a 24-digit hexadecimal number is carried out. Once this is done, a terminal hash digest is produced for the 512-bit blocks. Every 512-bit block will now have a 96-bit hash digest. Modular arithmetic is used to add these 96-bit digests to generate the final hash digest.

2.2.3 TDHA

The timestamp defined hash algorithm (TDHA) is a lightweight alternative to all the aforementioned algorithms. It was proposed in [11].

The process of TDHA is as follows. TDHA considers 192-bit messages (M). M is first padded with a 32-bit timestamp value to generate M' (224 bits). M' is then

divided into 14 blocks, each of size 16 bits. Further, each block is divided into two smaller blocks of 8 bits each. The 28 blocks thus generated are then divided into two segments each having 14 blocks. The blocks are then swapped among themselves (confusion). This output is also known as deformed M'. An XOR operation is then carried out between the two parts of each block. These blocks are then XORed and complemented to reduce the number to a total of 8 blocks. The final output is taken as the incomplete message digest (IMD). The receiver generates the message digest from this IMD.

Unlike LOCHA, TDHA does not require large prime numbers. Also, since a timestamp value is considered for padding, there is no need to store or compute this key beforehand. This reduces the computational overhead as well. Each message contains a digital signature of the sender to ensure that the message is circulated between authentic clients only.

Performance of MD5, SHA-1, LOCHA and TDHA can be compared as follows.

2.3 *Signature Verification Algorithms and Techniques*

2.3.1 One-Time Signature

The one-time signature algorithm was suggested by Merkle in [2]. To sign a message M with the Merkle signature scheme, the message M is signed with a one-time signature scheme, resulting in a signature sig' , first. Either of the public- and private-key pairs (X_i, Y_i) are used to get this done. The public-key pub, the message M and the signature sig' are known to the receiver. At first, the one-time signature of the message M is verified by the receiver. If the signature is valid, the receiver computes the hashing tree by hashing the public key of the one-time signature. If the hashing trees match, then the signature is valid.

2.3.2 Leapfrog Technique

In the leapfrog technique, the integrity of the packet of the previous node is verified by the cluster head using the secret key. Such a technique has been implemented in [12].

A generalized version of this mechanism is presented in the paper, wherein it is assumed that the available header space is $2t + 1$ bits. This space is divided into three fields, namely a one-hop neighbour authentication field (ONAF) (t bits), a two-hop neighbour authentication field (TNAF) (t bits) and a flag (1 bit) field. A secret key $k(x)$ is shared by all the cluster heads.

When a cluster head wants to send a packet P to be forwarded by a neighbouring cluster head, it sets the fields using a cryptographic hashing function. The protocol then follows the leapfrog approach. The packet modification by the previous node is verified by each cluster head by checking the hash value of the packet generated

by the node two hops far from it. If there is some change in the packet, that is, if the integrity has been disturbed, it is indicated by a verification failure.

3 Proposed Changes

We propose a modified approach to maintain the integrity and check for inconsistencies in the data transferred. This method involves the replacement of SHA-1 in the token generation and encryption process of LIVE, by a better and improved TDHA algorithm for hashing the token generated. It also involves the incorporation of the leapfrog technique in LIVE for content verification, instead of the traditionally used one-time signature method. The LIVE algorithm would be altered as shown in Fig. 2.

Using this instead of SHA-1 can further improve the lightweight factor of the algorithm, since the TDHA technique is much lighter, require lesser computational costs and improve processing capabilities.

Moreover, we can use the leapfrog technique. In this method, the data is clustered and the cluster head verifies if the integrity of the previous cluster. In the case of one-time signature algorithm that is used in LIVE, the signature size varies with the content size. In the leapfrog technique, the integrity of the packet of the previous node is verified by the cluster head by using a secret key. Analysis and simulation of leapfrog results show that the protocol needs a minimal number of header bits, namely three bits, resulting in insignificant bandwidth overhead. This will make the process extremely lightweight and will improve integrity.

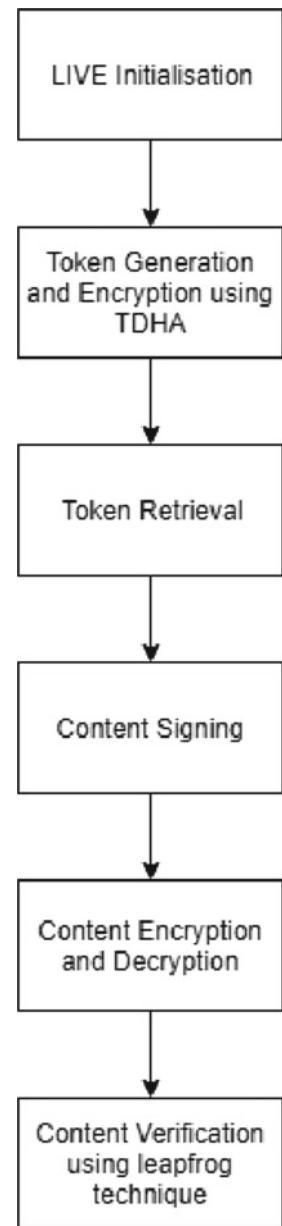
4 Results

On altering the token generation process by replacing the SHA-1 hashing algorithm with the TDHA algorithm, the following results were obtained (Fig. 3):

Using the lightweight alternative of TDHA, we observed a significant reduction in performance time. TDHA took one-fifth of the time that SHA-1 took for hashing an input digest of 54 bits. Also, as demonstrated in Table 2, TDHA offers an advantage in terms of the storage and computation overhead.

5 Conclusion

In this paper, we present a study on the prevalent lightweight data integrity algorithms like RSA and lightweight integrity verification (LIVE). Concluding that LIVE significantly outperforms RSA, we have presented improvements to LIVE to make it even more lightweight and improve performance. LIVE uses SHA-1 for token generation. Our analysis shows that TDHA significantly outperforms SHA-1. Using

Fig. 2 Suggested LIVE

(a)

```
[Sohams-MacBook-Pro:Downloads sohamkhedekar$ javac TDHA.java]
```

```
[Sohams-MacBook-Pro:Downloads sohamkhedekar$ java TDHA]
```

HashCode Generated by TDHA for:

```
13019076753627368 : 00000000 00000000 11001101 00000000 0  
00000000 00000000 10111000 00010001
```

Time Taken: 3907297

```
[Sohams-MacBook-Pro:Downloads sohamkhedekar$ javac SHA1.java]
```

```
[Sohams-MacBook-Pro:Downloads sohamkhedekar$ java GFG]
```

HashCode Generated by SHA-1 for:

```
13019076753627368 : 2911c8a615108933d1d83f26eb9d237862c06599
```

Time Taken: 23859910

(b)

SHA-1 vs TDHA

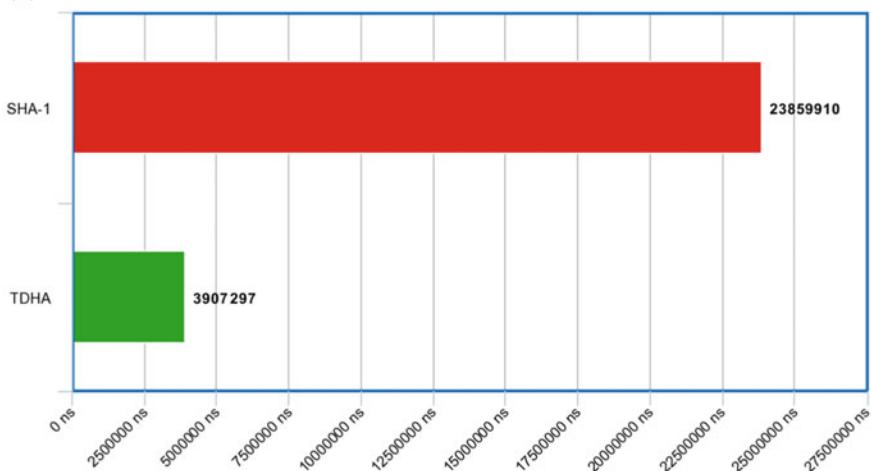


Fig. 3 Performance of SHA-1 and TDHA

this instead of SHA-1 can further improve the lightweight factor of the algorithm, since the TDHA technique is much lighter, require lesser computational costs and improve processing capabilities. Moreover, to further enhance security while keeping the algorithms lightweight, we can use another strategy called the leapfrog technique. In this method, the data is clustered and the cluster head verifies if the integrity of the previous cluster. In the case of one-time signature algorithm that is used in LIVE, the signature size varies with the content size. In the leapfrog technique, each cluster head checks if its previous node has preserved the integrity of the packet using the

Table 2 Performance of hash algorithms [11]

Scheme	MD5	SHA 1	LOCHA	TDHA
Computation overhead (clock cycles)	36,360	84,272	2952	159
Communication overhead (bits)	128	160	96	288
Storage overhead (bits)	12 registers (each of 32 bits)	12 registers (each of 32 bits)	4 registers (each of 16 bits) and 18 registers (each of 8 bits)	3 registers (each of 8 bits)

secret key. This will make the process extremely lightweight and will further improve the integrity mechanism.

References

- Li Q, Zhang X, Zheng Q, Sandhu R, Fu X (2015) LIVE: lightweight integrity verification and content access control for named data networking. In: IEEE transactions on information forensics and security. IEEE, pp 308–320
- Merkle R (1987) A digital signature based on a conventional encryption function. In: Advances in cryptology—CRYPTO. Springer, pp 369–378
- Zhang K (1998) Efficient protocols for signing routing messages. In: Proceedings of the network and distributed system security symposium—NDSS. The Internet Society, pp 1–7
- Rivest R (1992) The MD5 message-digest algorithm. In: Internet RFC 1321. RFC Editor
- National Institute of Standards and Technology (1993) FIPS 180-1 Secure Hash Standard (SHS). FIPS-PUBS
- Stevens M (2012) Cryptanalysis of MD5 & SHA-1. In: 80th anniversary of breaking the enigma code—return to the roots. Military University of Technology, Warsaw
- National Institute of Standards and Technology (2001) FIPS 180-2 Secure Hash Standard (SHS). FIPS-PUBS
- Dworkin M (2015) SHA-3 Standard: permutation-based hash and extendable-output functions. Federal Information Process Standards. NIST FIPS
- Bertoni G, Daemen J, Peeters M, Van Assche G (2007) Sponge functions. In: Ecrypt workshop on cryptographic hash functions. Barcelona
- Chowdhury A, Chatterjee T, DasBit S (2014) LOCHA: a lightweight one-way cryptographic hash algorithm for wireless sensor network. In: Elsevier science procedia computer science. ScienceDirect, pp 497–504
- Mondal A, Mitra S (2016) TDHA: timestamp defined hash algorithm for secure data dissemination in VANET. In: International conference on computational modeling and security. ScienceDirect, Bangalore, pp 190–197
- Durresi A, Paruchuri V, Kannan R, Iyengar S (2005) A lightweight protocol for data integrity in sensor networks. Int J Distrib Sens Netw 1(2):205–214

Chapter 9

Catchment Area Detection and Optimization



**Richard Joseph, Sanket Gokhale, Akash Hasamnis, Grishma Gurbani
and Rishil Kirtikar**

1 Introduction

1.1 Motivation

The Marathwada region of Maharashtra, a perennially drought-plagued area, is once again staring at a severe water scarcity. Over 41% of the villages in this region of central Maharashtra have reported ‘average yield’ below 50 paise. This will force the government to mull augmenting foodgrain and water supply in the region. A data compiled by the revenue department revealed that more than 3,500 villages in Marathwada have reported average yield less than 50 paise. This means the scarcity of basic resources such as foodgrain production and water is less than 50% of its actual capacity. Erratic showers are generally the main cause of less average yield, a senior revenue official said. Of the 153 days of the monsoon in Marathwada last year, 94 days were dry. This clearly means there were more dry days in the monsoon season than wet. There are many such reasons in India that are agricultural and need a good supply of water but due to scarcity, these areas are becoming barren and the conditions are not worth living for any living creature. The biggest challenge in such areas is to harvest the water and to increase the number of catchment areas that can lead to more storage of water.

R. Joseph · S. Gokhale (✉) · A. Hasamnis · G. Gurbani · R. Kirtikar
Vivekanand Education Society’s Institute of Technology, Hashu Advani Memorial Complex,
Collector’s Colony, Chembur, Mumbai, Maharashtra 400074, India
e-mail: sanketgokhale97@gmail.com

1.2 Relevance

This project focuses on the issues related to water scarcity in the rural parts of Maharashtra, where there are limited water resources like lakes and rivers. The groundwater level of these areas is prone to reducing drastically on account of various environmental issues like global warming, etc. The project aims at increasing the catchment areas in places which have actual or potential drought-like conditions and thereby increasing the groundwater level, which will eventually help in the long run. We aim at providing solution to make such regions more sustainable. Such areas not only lack rainfall but also the way the water is stored and its supply managed.

These results would not only benefit our work, but also the work of thousands of other practitioners and researchers, who find it difficult to analyse the changes in the environment. The application is intended to be open source.

2 Methodology

The proposed system works in the following phases.

2.1 Data Gathering and Mapping

This step is about gathering the data related to environmental factors from different sources. This data is collected w.r.t. latitude and longitude, to understand the environment of a particular location. Here, we gathered data related to following parameters: rainfall, temperature, soil type, groundwater level, humidity and altitude. The data for parameters temperature and altitude was collected with the help of weather APIs, whereas the data for parameters humidity, soil type, rainfall and groundwater level has been collected from Government weather portals and Indian Meteorological Department datasheets.

2.2 Data Pre-processing and Feature Extraction

Data extracted from APIs will not need much pre-processing and can be directly used as features. It will just need to be transformed into a suitable format for further analysis. The rainfall data obtained was monthly data from 1996 to 2017 with a number of missing values. Firstly, missing values were handled by least square regression. Further for clustering, we used the mean value of rainfall over the years for a district. The soil type was extracted by comparing with soil maps available online. Elevation

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	DISTRICT	TEH_NAM_BLOCK_NAME	LAT	LONG	SITE_NAME	YEAR_OBS	TOTAL_RAINFALL	RAINFALL	GROWTR	SOIL_TYPE	Soil_Score	Elevation	Region_Score	Indexing1	catchment1	Indexing2	catchment2
2	Ahmadnagar	Ahmednagar Akola	19.525	73.99	Malijap	1996	965.08	15	4.0125	Black	5	649	1	440.19	1	4204.51	0
3	Ahmadnagar	Ahmednagar Akola	19.525	73.925	Rajur	1996	965.08	14	3.23	Black	5	649	1	440.19	1	4204.51	0
4	Ahmadnagar	Bhingar Jamkhed	18.73333	75.31667	Jamkhed	1996	965.08	24	4.6125	Black	5	650	1	404.8375	0	4209.158	0
5	Ahmadnagar	Bhingar Jamkhed	18.73333	75.31667	Jamkhed	1996	965.08	23	4.6125	Black	5	650	1	404.8375	0	4209.158	0
6	Ahmadnagar	Deoli Pr Karjat	18.64444	74.89567	Bhikkewadi	1996	965.08	19	5.01	Black	5	515	1	370.33	0	4145.83	0
7	Ahmadnagar	Deoli Pr Karjat	18.51667	74.93333	Kolwadi	1996	965.08	22	8.925	Black	5	515	1	378.275	0	4154.595	0
8	Ahmadnagar	Deoli Pr Karjat	18.63333	75.1	Pategaon	1996	965.08	21	1.83	Black	5	515	1	351.05	0	4131.37	0
9	Ahmadnagar	Deoli Pr Karjat	18.43333	74.93333	Rassan	1996	965.08	20	5.7375	Black	5	515	1	352.7125	0	4145.033	0
10	Ahmadnagar	Deoli Pr Karjat	18.63333	74.88333	Walwa	1996	965.08	17	2.125	Black	5	515	1	345.875	0	4134.195	0
11	Ahmadnagar	Ghulewad Kopargaon	19.89444	74.35694	Kolpewadi	1996	965.08	18	2.505	Black	5	509	1	336.015	0	4132.335	0
12	Ahmadnagar	Ghulewad Kopargaon	19.88333	74.475	Kopargaon-	1996	965.08	16	2.175	Black	5	509	1	355.025	0	4131.345	0
13	Ahmadnagar	Ghulewad Kopargaon	19.95	74.28333	Manjur	1996	965.08	21	9.70	Black	5	509	1	351.78	0	4154.1	0
14	Ahmadnagar	Ghulewad Rahata	19.76667	74.43333	Rahata	1996	965.08	15	8.0225	Black	5	509	1	344.5675	0	4148.888	0
15	Ahmadnagar	jamkhet Nagar	19.60556	74.63333	Chas	1996	965.08	14	2.125	Black	5	596	1	390.375	0	4174.695	0
16	Ahmadnagar	jamkhet Nagar	19.60556	74.63333	Khodchandi	1996	965.08	19	2.625	Black	5	596	1	399.5	0	4174.293	0
17	Ahmadnagar	jamkhet Nagar	19.60556	74.63333	Khodchandi	1996	965.08	21	3.405	Black	5	596	1	410.35	0	4176.0	0
18	Ahmadnagar	jamkhet Nagar	19.23333	74.66667	Dherva	1996	965.08	23	1.025	Black	5	596	1	367.075	0	4171.395	0
19	Ahmadnagar	jamkhet Nagar	19.10556	74.63056	Jahangon	1996	965.08	14	9.77	Black	5	596	1	405.19	0	4197.531	0
20	Ahmadnagar	jamkhet Nagar	19.1225	74.80556	Jeur	1996	965.08	17	7.715	Black	5	596	1	399.145	0	4191.465	0
21	Ahmadnagar	jamkhet Nagar	19.03333	74.83333	Takle-Ka	1996	965.08	17	5.9375	Black	5	596	1	4113.825	0	4186.133	0
22	Ahmadnagar	Kopargaon Nevasa	19.34722	74.88154	Ghodegaon	1996	965.08	22	3.105	Black	5	515	1	348.815	0	4137.135	0
23	Ahmadnagar	Kopargaon Nevasa	19.51194	74.85417	Gonezaon	1996	965.08	18	10.15	Black	5	515	1	369.95	0	4158.27	0

Fig. 1 Data set

data was gathered manually using APIs. Groundwater data was extracted from the India water resources website. For classification, we required yearly total from 1996 to 2017 for each district of Maharashtra. Similar set of operations were conducted for groundwater data. A screenshot of the dataset is provided below. Rainfall, soil type, groundwater and elevation were the dimensions used to train the model (Fig. 1).

2.3 Identification of Drought-Prone Areas

Feature extracted will now be used to train the neural network algorithm to identify the drought-prone areas. In this section, we present how the two-class support vector machine will train the model to identify the drought-prone regions w.r.t. latitudes and longitudes.

We use neural networks as its goal approaches to combinatorial optimization to formulate the desired objective function being optimized, such that it can be viewed as a ‘natural’ energy minimization problem. We have many parameters that are very closely related in this scenario, for example, the evaporation rate is directly proportional to temperature, humidity in soil is related to soil type, etc. All of these parameters with combinations will give different results and we need an optimized result that would help us detect the drought-prone area.

2.4 Identification of Areas Suitable for Catchment

As we identify the areas that are hit by drought, our research to find a source of water starts exactly from the point where we found the hit for drought. We can help such areas in two ways. We can either help such areas to increase the groundwater level or we can build some kind of reservoir for the area. All of these depend on many factors that will eventually help to harvest and conserve water for long duration of time.

There are several factors that lead to drought-like conditions. Few are natural and few are man-made. With the help of unsupervised learning and by using competitive learning rule, we can come out with combinations of parameters that might give us the best result for the region as all the combinations that are considered as output compete with each other and finally the winner takes all concept is implemented.

3 Literature Survey

Randomized decision trees have seen plenty of usage in machine learning applications. However, if the data size becomes huge, the number of nodes in the tree grows exponentially with increasing depth. This becomes especially important when the memory available is limited. This may, in turn, affect the accuracy of the application, and [1] proposes an alternate solution to this in the form of decision jungles, revisiting the idea of ensembles of rooted decision directed acyclic graphs (DAGs). In a DAG decision jungle, there can be multiple paths from a root node to a leaf node, unlike conventional decision trees. [1] shows that decision jungles perform better in terms of memory requirements and generalizations as compared to decision jungles. Our application had the climatic data of the entire state at a high granularity, thus having a significant memory requirement. Hence, the solution provided in [1] proved to be more efficient and accurate compared to other methods.

The authors of [2] present a way in which combination of already existing modules in Microsoft Azure Machine Learning Studio is used to obtain maximum accuracy. A generalized flow is created by the authors by modifying and combining the modules in MAMLS with other modules in ‘R’ language. This generalized flow analyses a small part of a big data set. High multi-class and binary classification accuracies are obtained with minimal manual intervention. Dimensionality reduction modules and decision-making modules are also used and combined with the MAMLS modules.

The objective of [3] is to determine the condition and classification of the water catchment area in Semarang city. Semarang city is one of the biggest cities in Indonesia where one of the main problems is the shortage of water in the dry season. The rate of groundwater replenishment depends on two conditions: (i) water recharge rate and (ii) catchment area condition. The catchment area conditions are determined by five parameters which are soil type, land use, slope, groundwater potential and rainfall intensity. The final result in [3] divides the water catchment into six criteria ranging from good to very critical condition, which is calculated for every point using the five parameters listed above and assigning a weight to each of the factors depending on their impact on catchment conditions.

4 Project Proposal

Various weather parameters will be collected for the state of Maharashtra. These include rainfall, temperature, humidity, groundwater, etc. Further, the elevation of a place is identified using APIs. The soil type of a place is identified and scored according to water soaking capacity. All of the collected data is cleaned and missing values are handled using least square regression. Further, the parameters are evaluated and a score is generated for a region these scores are used to label the data. Finally, labelled data is trained and model is tested for accuracy for various algorithms on Azure ML Studio. The model with best accuracy is selected and REST API is generated for the same. The new points are tested and results are reported.

4.1 Flow Chart for the Proposed System

See Fig. 2.

5 Comparative Analysis with the Existing Algorithms

See Table 1.

5.1 Inference Drawn from the Test

Training accuracy is greater than testing accuracy in case of only three of the studied algorithms. In these three algorithms, two-class decision forest algorithm is clearly overfitting because its training accuracy is 99.6%. Out of the remaining two algorithms, two-class decision jungle [4] has maximum testing accuracy.

Decision provided us with better accuracy along with being more memory efficient.

Hence, our selection of the two-class decision jungle algorithm is justified.

6 Type of Testing Considered with Justification

6.1 Attributes Used for Training the Model

Rainfall, groundwater, soil score, elevation, latitude and longitude.

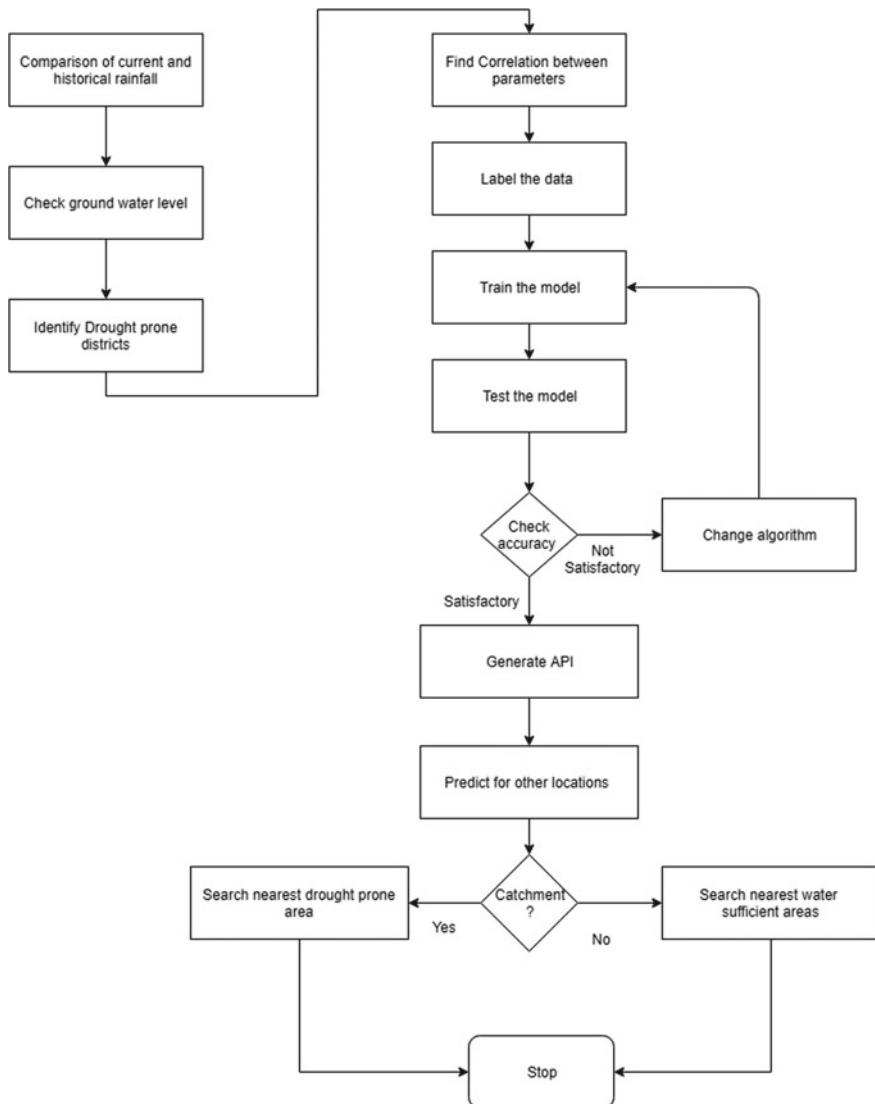


Fig. 2 Flow chart

Target variable: catchment

Possible values of target variable:

1. 0 (means the given area is not suitable for catchment)
2. 1 (means the given area is suitable for catchment)

Algorithm used for training the model: two-class decision jungle.

Table 1 Comparison of different algorithms considered for study

Algorithm	Ratio of data	Accuracy	True positive	True negative	False positive	False negative
Two-class support vector machine	70:30	0.924	1036	6646	49	579
	80:20	0.923	692	4420	34	394
	90:10	0.921	332	2218	17	203
Two-class boosted decision tree	70:30	0.967	1473	6565	130	142
	80:20	0.968	996	4364	90	90
	90:10	0.969	487	2197	38	48
Two-class decision forest	70:30	0.967	1442	6595	100	173
	80:20	0.968	973	4387	67	113
	90:10	0.964	466	2203	32	69
Two-class decision jungle	70:30	0.973	1450	6632	63	165
	80:20	0.969	970	4401	53	116
	90:10	0.969	475	2209	26	60
Two-class locally deep support vector machine	70:30	0.938	1189	6607	88	426
	80:20	0.937	810	4383	71	276
	90:10	0.932	381	2201	34	154
Two-class logistic regression	70:30	0.925	1038	6645	50	577
	80:20	0.923	695	4417	37	391
	90:10	0.920	333	2215	20	202
Two-class neural network	70:30	0.942	1144	6680	15	471
	80:20	0.935	843	4337	117	243
	90:10	0.936	407	2187	48	128

6.2 Impact of Attributes

Initial accuracy by considering all attributes for training.

Accuracy: 97.2%

Algorithm used two-class decision jungle (Fig. 3, Table 2).

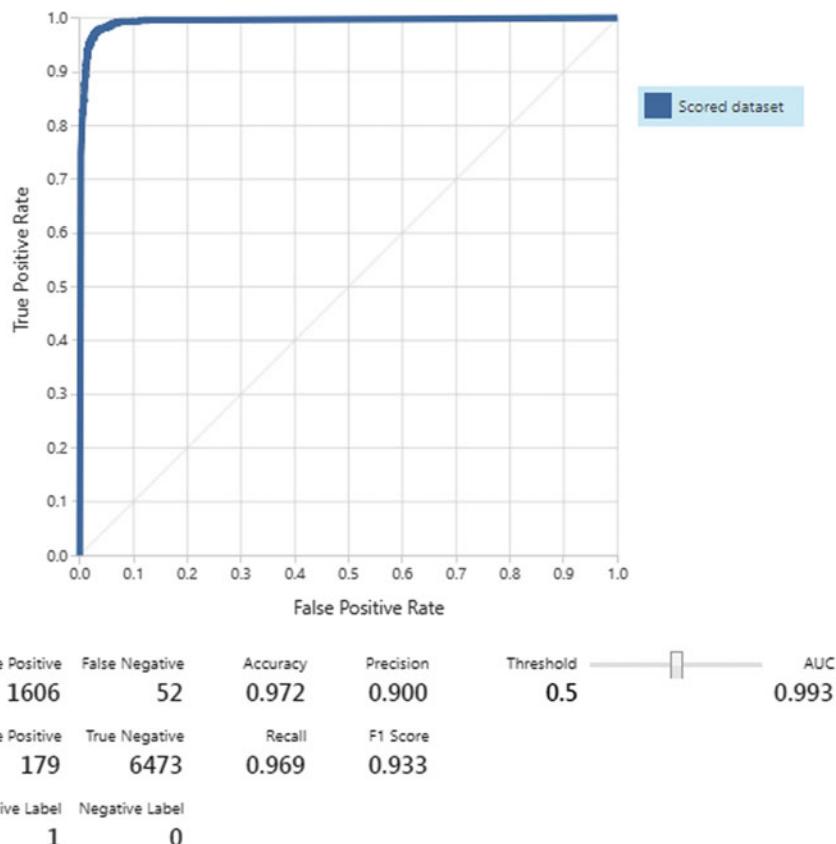


Fig. 3 Initial accuracy

Table 2 Impact analysis of attributes and comparison with weights assigned previously

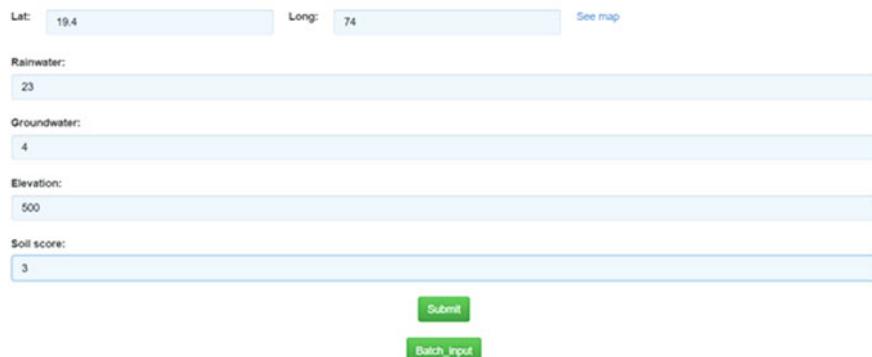
Removed attribute	Current accuracy	Change in accuracy	Weight assigned
Rainfall	95.30	1.90	4
Groundwater	96.90	0.30	3
Soil score	97.10	0.10	2
Elevation	97.10	0.10	0.5

7 User Interface

7.1 Single Input

See Fig. 4.

Inputs to predict catchment :



A screenshot of a web-based input form for predicting a catchment area. The form includes fields for Lat (19.4), Long (74), Rainwater (23), Groundwater (4), Elevation (500), and Soil score (3). It also features a 'See map' link and two green buttons at the bottom: 'Submit' and 'Batch_Input'.

Lat: 19.4 Long: 74 See map

Rainwater:
23

Groundwater:
4

Elevation:
500

Soil score:
3

Submit

Batch_Input

Fig. 4 Single input



A screenshot of a web-based batch input form. It features a file upload field with the path 'C:\Users\rag\Desktop\jrunion\ufsample_batch_input\sample_batch_input.csv' and a 'Browse...' button. Below the file field is a green 'Submit' button.

Upload csv file for batch input:
C:\Users\rag\Desktop\jrunion\ufsample_batch_input\sample_batch_input.csv

Submit

Fig. 5 Batch input

7.2 *Batch Input*

See Fig. 5.

7.3 *Output Visualization*

The results were plotted on Google Maps using folium library in Python [5] (Fig. 6).

8 Evaluation of the Developed System

See Fig. 7.

8.1 *Graphical Outputs of the Various Scenarios Considered*

See Figs. 8 and 9.

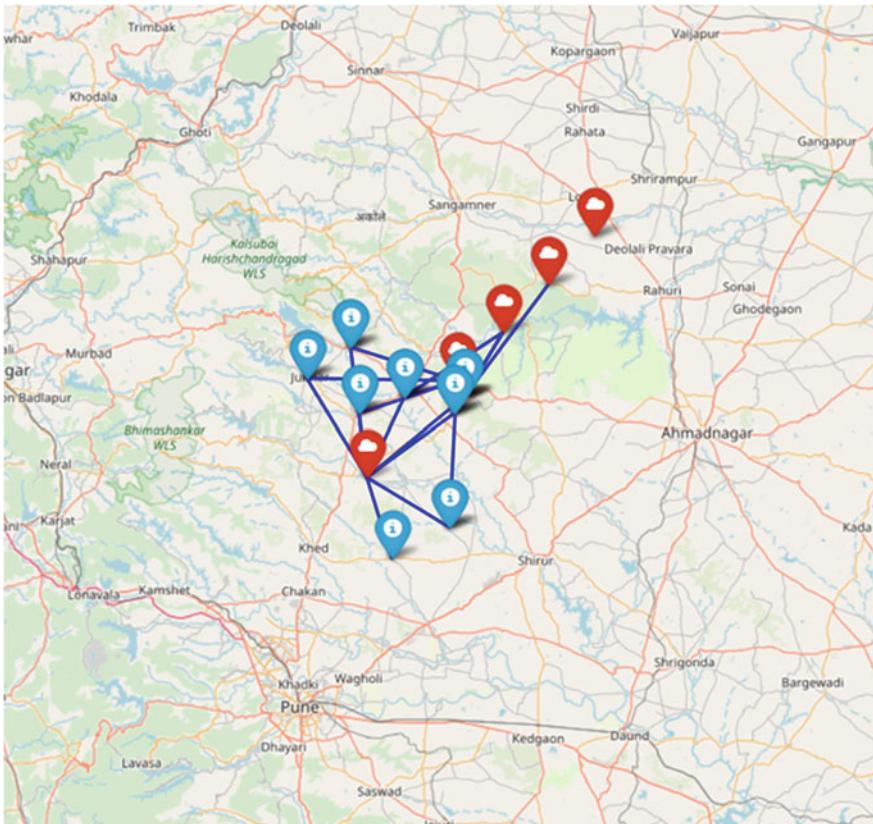
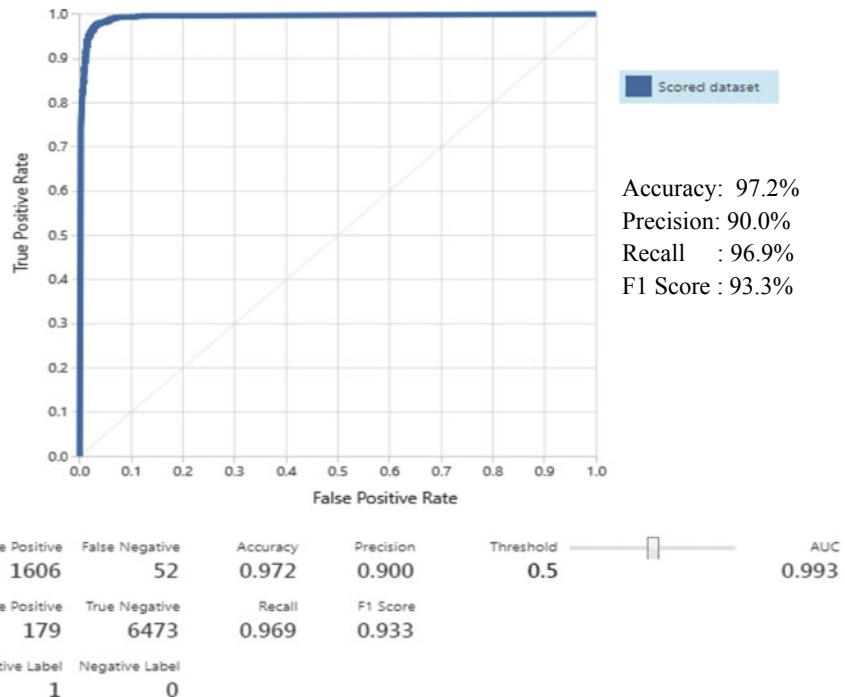
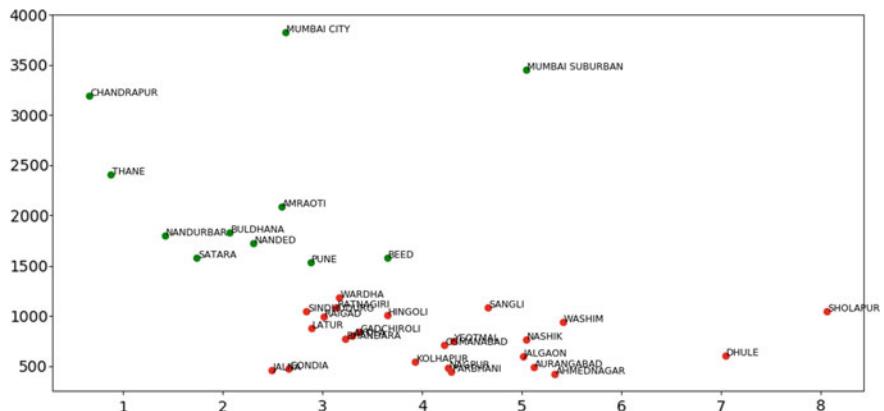


Fig. 6 Map visualization

9 Conclusion

The geographical and climatic diversity of India leads to an uneven distribution of water in its different parts. Areas which are very near to each other face a completely different climate. This combined with the recent climatic changes because of pollution and global warming has created water scarcity in many regions. The main focus of our project is to deal with this problem by finding alternate sources of water supply for a region facing water scarcity. The application will be open source benefitting many environmental scientists and government officials working on water conservation.

**Fig. 7** Result evaluation**Fig. 8** Rainfall versus monsoon groundwater k-means clustering

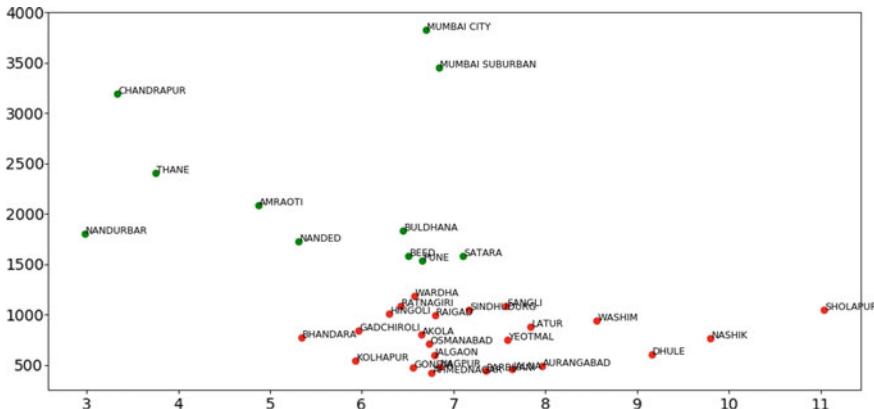


Fig. 9 Rainfall versus pre-monsoon groundwater k-means clustering

10 Future Scope

- Further advancement in these models would allow us to predict the type of catchment that could be developed, i.e. the predicted catchment would be a self-sustained one or would require human intervention.
- Also, once we determine whether an area is water sufficient, we need to also consider how suitable that area is for water supply. For that, many more factors like urbanization, forest cover, distance from the drought-prone region, etc. need to be considered.
- This will also help us to determine that for a given region, if multiple places can be used for supplying water, which among them is best suited or is the optimal place for water supply.

References

1. Shotton J, Sharp T, Kohli P, Nowozin S, Winn J, Criminisi J (2013) Decision jungles compact and rich models for classification
2. Bihis M, Roychowdhury S (2016) A generalized flow for multi-class and binary classification tasks: an Azure ML approach
3. Prasetyo Y, Gunawan SA, Maksum ZU (2016) Determination of the water catchment area in Semarang city using a combination of object based image analysis (OBIA) classification. In: SAR and geographic information system (GIS) methods based on a high-resolution SPOT 6 image and radar imagery
4. <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-decision-jungle>
5. AnkitRai01: <https://www.geeksforgeeks.org/python-plotting-google-map-using-folium-package/>

Chapter 10

Automation in Healthcare Using IoT and Cryptographic Encryption Against DOS and MIM Attacks



Prajakta Kamble and Aruna Gawade

1 Introduction

In IoT-based smart healthcare applications, many entities introduced like analogue sensors, microcontrollers, databases, android applications, as well as user-friendly graphical user interface. Such applications carried out some classification algorithms, which works on synthetic as well as time series data sets, the traditional IoT systems, generate incorrect or approximate values due to environmental factors or machine errors. To eliminate such problems, many authors have introduced various machine learning algorithms during disease prediction on real-time data. When system works with few analogue sensors like one or two, it does not require such a kind of machine learning algorithms. We have proposed an IoT-based patient monitoring system with security mechanism. First, IoT model is introduced where system collects the two parameters via wearable devices from patient body. The temperature and pulse rate sensor are used as parameters. After specific time interval, both sensors give desired value to microcontroller. Once microcontroller received any data from sensors, it converts from analogue to digital with the help of ADC. We introduced cloud data storage to store data received from microcontroller.

In second layer, we developed Q-learning machine learning algorithm, which monitors all the data received from cloud server. The specific threshold has initialized during the execution for both parameters. The minimum and maximum thresholds have been set for body temperature as well as pulse rate. According to linear regression approach of Q-learning algorithm, it generates reward and penalty when violates the Q-learning policy from input values. At the final stage of algorithm, it will generate the penalty weight for entire time series data set, and if this weight higher than algorithm's threshold, it generates critical alert for specific patient. Once such event has generated, it will update on another dataset that continuously synchronizes into

P. Kamble (✉) · A. Gawade

Computer Engineering Department, SVKM's DJSCE, UOM, Mumbai, India

the doctor's application. The basic benefit of such event prediction scenario is that doctors can immediately get information about event of a specific patient.

In the final phase of system, we introduced two different network attacks like denial-of-services (DOS) and man-in-the-middle (MiM) attack. Basically, both attacks have generated on database, when DOS has generated, it completely destroyed the entire database and when MiM attack changes entire values with some dummy or invalid values. In the proposed system, we carried out prevention mechanism from both attacks. In admin model, we prevent both attacks using role-based access control (RBAC) policy mechanism and cryptographic encryption algorithm AES 128. In multiple experiment analysis, we confirm our defence mechanism having the ability to prevent both attacks in secured environment.

There are many different ideas and solutions which can be applied in cloud computing. Encryption is one of the effective solutions for data security and integrity. A cryptographic algorithm is proposed using an integration of symmetric and asymmetric cryptographic techniques, viz Blowfish and RSA encryption algorithms which strengthens the encryption algorithm. Technical experts and authors are mainly concerned about the privacy and security of sensitive data transfer. It provides confidentiality to the data as in no stage data is exposed in plain text. However, it also has disadvantages such as complexity and multi-regional regulatory and legal issues. System lacks resistance to other attacks such as distributed denial-of-service (DDoS) and it is difficult to determine which party performed message encryption or decryption. DDoS is a major attack, which completely blocks the network, and hence users cannot access their data. There are various DDoS defence mechanisms to counter DDoS attacks. System also proposed a DDoS defence mechanism using software-defined networking, this centralizes the network and enables configuration network configuration. Hence, attack detection and attack prevention rates are increased.

2 Literature Survey

Encryption strategies are subdivided into two types as, symmetric encryption algorithm and public-key encryption algorithms. Asymmetric encryption algorithms provide robust security. However, as processing and memory are restricted resources in sensor nodes, the complexity and high-power usage of asymmetric encryption algorithm makes them difficult to use in wireless sensor networks. Symmetric encryption algorithms are mostly exercised in wireless sensor networks as they have simple and little amount of computations. There are many lightweight executions of AES encryption available to protect Cloud-IoT systems from hostile attacks, i.e. Trojan [1]. This algorithm technology is quite mature, but weak in security.

A node or a device takes numerous characteristics that may not really be legal. It does not imitate any node, yet quick it just accepts the identity of another among a few nodes, causing redundancies in the routing convention. Sibil attack is a security threat when a node in a network claims multiple identities.

DoS Attacks: In IoT, DDos attack [2] is one of the most common network attacks. Generally, the solution is to upgrade the framework and exercise DoS attack prevention and detection practices.

Data Transit Attacks: numerous attacks on privacy and integrity of data during transmission in access or core networks, for example, sniffing, and man-in-the-middle.

ARP vulnerabilities: The attacker can misdirect the in/out movement of casualty VM to a faulty VM by manipulating the address resolution protocol does not need proof of existence.

DNS poisoning attack: In this attack, the attacker abuses the susceptibilities in the network to redirect the entire traffic away from authentic servers and towards the malicious ones.

Sniffer attack: At the correspondence level, snooping (also called sniffing) alludes to purposefully tuning into confidential exchange over the communication links [3]. Because of this, the attacker gets highly important data after decryption. In such circumstances, when packets additionally carry access control data, like node configuration, shared network password and node identifiers, sniffing may give this critical information. The attacker may misuse this collected data to plan other adapted attacks.

For instance, if attacker can effectively extricate the data that is required to add another node to the arrangement of approved nodes, he will effortlessly have the capacity to add a malevolent node to the framework.

Man-in-the-middle: Here, the attacker secretly relays him/herself into a communication between two devices, and possibly alters the communication between two parties who believe they are directly communicating with each other. Ziegeldorf et al. [4] have presented end-to-end secured IoT framework through applications and devices in IoT. The architecture of IoT mainly consists of three main characteristics, namely IoT application, IoT devices an IoT broker. The function of IoT broker is to manage connected devices and aggregating data from sensor. The IoT application mainly carries out the function of providing users with IoT services. We can access the IoT services, only if it has access to sensor data. Therefore, real-time healthcare services demand privacy and security because patient's medical data is one of the most sensitive data which is highly susceptible for attacks.

Choi et al. in [5] have also studied the privacy issues, which are explained in the Internet of Things in detail. His work explains new features and trends in Internet of Things. Ziegeldorf's study goal was to scrutinize all privacy and security issues. Meanwhile, he classified and examined the threats involved in his new research. The main focus of his new research is to make Internet of Things a highly secured technology in the real world by overcoming the privacy threats and challenges.

The change in body temperature is used to identify homeostasis, which forms an essential part in healthcare services. In a medical IoT, a TelosB mote is used which has an embedded sensor for recording body temperature. Jian and colleagues [6] proposed a distinctive system that uses a home gateway to monitor body temperature. The home gateway uses infrared detection to transfer the generated data of body temperature. This system primarily includes an RFID module, which is associated

with a body temperature sensor device to function. Smartphones are increasingly used as integral parts of a medical IoT. All the newer electronic devices are now controlled by smartphones. Several healthcare-related hardware products have been integrated and many software applications have been developed for smartphones [7]. Image analysis algorithms are widely used in healthcare systems and healthcare applications.

Many people continuously wear live monitoring devices which track medical parameters and keep track of their health status. In such scenarios, security is vital concern, as a violation of the network data will lead to life threat [8]. Therefore, there is no compromise in security and integrity of the information obtained from different sources. To have a secured data transfer, new secured theories and technologies should be implemented, which ensures patient's data security, integrity and confidentiality [9].

There are different characteristics like confidentiality, integrity and availability of people's personal data that should be guaranteed in an healthcare system using Internet of Things. Another key requirement is proficient security of available resources. The healthcare systems based on IoT should be equipped with foolproof structure which utilize minimum resources with high security performance. The devices used in the healthcare system should have tamper-proof packaging [9].

Any third party or foreign attacker can easily take control of a device and alter the system network so as to obtain crucial data. Additionally, the routing algorithms used should be properly controlled to safeguard the transmitted data. The nodes of the network are very prone to attacks. Therefore, secured routing protocols are needed to transmit the data through network [10]. The medical devices with IoT technology have the mechanism to connect and access cloud services [8]. But alongside the services should be configured and monitored properly to track and control the patient's data accurately.

3 Proposed Framework

First system collects the current information states from every sensor, which is converted from analogue to ADC, and received by microcontroller, to store in the database. In the meantime, the Q-learning algorithm analyses and classifies the received data from the sensors and show it on graphical user interface (GUI). It will check all values and compare it with the threshold value, if the received value is above or below the threshold values, it will send critical alert on the android application.

Figure 1 shows entire execution of proposed system. Once IoT system has enabled, it will generate temperature and pulse rate in a specific time interval and sends to ADC and it will forward to Raspberry Pi, the details information has given below.

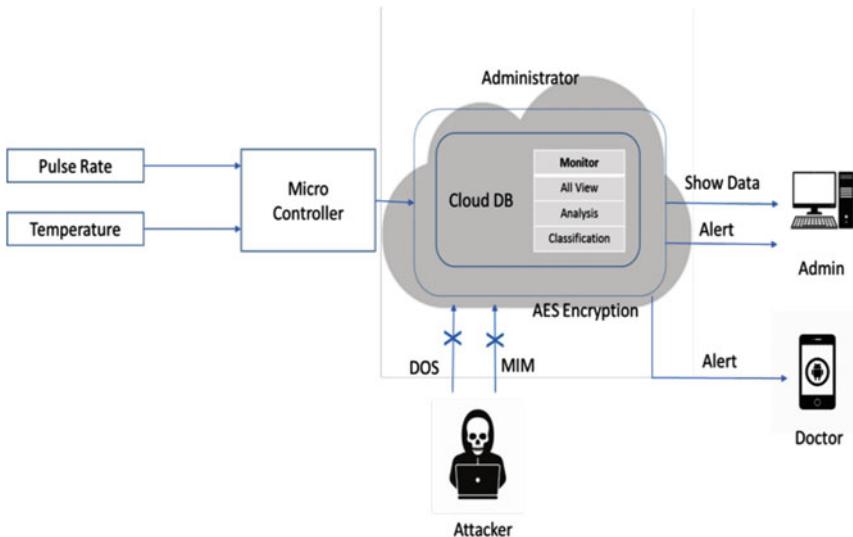


Fig. 1 Healthcare system framework

- ADC converts those values from analogue to digital and transfer to power Raspberry Pi, at the same time we have written the script which will post this data into the cloud server.
- In our Java Web application, we continuously monitor active patient data in Web GUI.
- The Q-learning-based machine learning algorithm has been implemented to predict the dangerous event according to the given parameters, the system will send alert message automatically when number of penalties given by algorithm on generated data.
- If Q-learning sends dangerous alert to cloud server, it will automatically shows on doctor's device with entire patient information.
- This system is able to eliminate DOS as well as MiM attacks simultaneously. The DOS attack shuts down a network, making it inaccessible to its intended users. Dos attack accomplish it by flooding the target with traffic that triggers a crash of database and Man-In-Middle attack secretly relays messages and change alter actual value of temperature and Pulse rate.
- An intrusion detection and prevention framework will eliminate those attacks and provide original data to our system.
- We have used AES 128 encryption algorithm to encrypt end-to-end data from sensors to cloud and mobile application.

3.1 Algorithm 1: Reinforcement Learning (Q-Learning Algorithm)

Input: i/p[1 ... n]: i/p values obtained from sensors,
 $T_{low}[1 \dots n]$: Minimum Threshold value from sensors
and $T_{high}[1 \dots n]$: Maximum value from sensors,

Th: Ideal Threshold value

Output: Output device triggered

Procedure:

- 1: Retrieving data from database (Db)
 - 2: Parts [] \leftarrow Split (Db)
 - 3: CVal = $\sum_{k=0}^n$ parts[k]
 - 4: Compare CVal. Within range of $T_{low}[1 \dots n]$ and $T_{high}[1 \dots n]$)
 - 5: Obtaining current state with Time stamp (T)
 - 6: Logic:
if CVal < Th or CVal > Th \rightarrow Penalty (Tp)
if CVal = Th \rightarrow Reward (Tr)
 - Read all values obtained of penalty Tp and reward Tr
 - else Continue T++
 - 7: Evaluating Penalty Score (Ps) using equation: (Tp * 100/Total states)
 - 8: if (Ps > = Th):
 - Generate event
- End for

Algorithm 2: AES 128

Key generation

Input: Required key size as Keybit {128, 192, 256}

Output: Key pair generation for encryption as well as decryption as pKey

Step 1: Slat[] = $\sum_{k=0}^n$ Rand[k-value]

Step 2: Key \leftarrow AddRoundKey(Keybit, salt[]])

Step 3: Return key

Encryption

Input: plaintext data as PlainData, Key for Encryption pKey

Output: encrypted data as cipherData

Step 1: Cipher [] \leftarrow Encrypt (pKey, PlainData)

Step 2: Cipherdata \leftarrow String64Encoder(Cipher)

Step 3: Return Cipherdata

Decryption

Input: encrypted data as cipherData, key for Decryption as pKey

Output: plaintext data as plainData

Step 1: $T \leftarrow \text{Decrypt}(pKey, \text{CipherData})$

Step 2: $\text{plainData} \leftarrow \text{String64Encoder}(T)$

Step 3: Return plainData

3.1.1 Dataset Information

For the proposed system, we have build own IoT module with temperature as well as pulse rate sensor, according to programming strategy, system generates both values (Body temperature and pulse rate) based on the specific time interval, and generated data is stored into the cloud database simultaneously. Basically, the system works like real-time healthcare patient monitoring system which generates runtime dynamic values according to current scenario. The experiment has done on 100,000 records with 100 patients.

3.1.2 Experimental Set up

For the system performance evaluation, calculate the matrices for accuracy. The system is executed on java 3-tier architecture framework with INTEL 2.8 GHz i3 processor and 4 GB RAM with public cloud Amazon EC2 cloud database. IoT model has been built with Raspberry Pi 3.0 which is connected to cloud database. When microcontrollers initiate the system, all sensors get activated to generate data. This analogue data is then transferred to the cloud server and stored in cloud database. In this system, both MySQL and Apache run on the same virtual machine (VM) in the cloud server. IoT devices generate data layers. Communication network is set between application layer and data layer. Java UI and android application are displayed by the system which works like service-oriented architecture (SEO).

4 Results and Discussion

The proposed system has been implemented on three different platforms; initially, we have created one IoT module that consists of two analogue sensors, including microcontroller (Table 1).

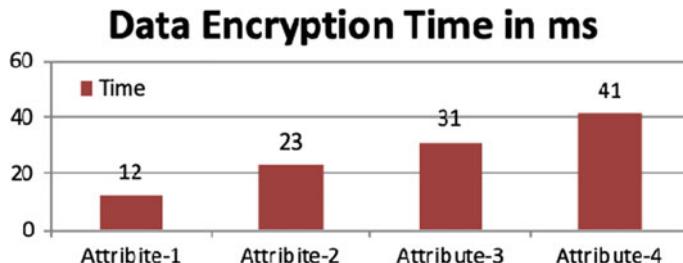
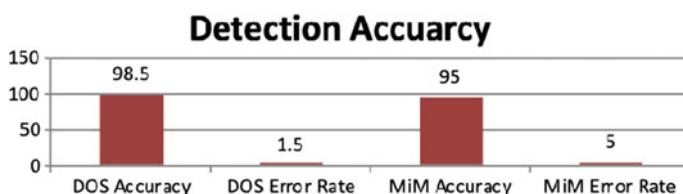
Second, we made one mobile application where doctor can continuously monitor associated patients data, this application also illustrates the scenario when data has trash for hack by attackers. In the third phase, we have developed one graphical user interface for administrator, doctor as well as patience, respectively. Administrator

Table 1 Data encryption and decryption time

Data size (KB)	Encryption time (Ms)		Decryption time (Ms)	
	Existing (MD5)	Proposed (AES)	Existing (MD5)	Proposed (AES)
5	595	515	724	612
10	1120	1026	1132	1033
15	1680	1547	1687	1556
20	2260	2064	2231	2033

can monitor all patients' data, doctors' data and he can prevent system from malicious connections. In this section, we discuss about the system performance analysis with internal as well as external functionalities. Figure 2 shows time required for encryption when system generates a security view, which is illustrated in milliseconds, and Fig. 3 shows attack detection and prevention accuracy for proposed DOS and MiM, respectively.

The figure shows the system performance evaluation has done with various experiment analysis. In entire system, we have done different experiments with all models and calculated the average results, which is shown in Figs. 2 and 3. Finally, we conclude that the proposed system can able to handle role-based access control with data security in non-trusted execution environment as well as it also able to defence various kind of networks attack like DOS or MiM, etc.

**Fig. 2** Time required for data encryption in milliseconds with various attribute set**Fig. 3** Attack detection and prevention accuracy for DOS and MiM

5 Conclusion

In traditional approach, what we have found that the majority of the tests are intrusive, which is inconvenient to the patient and this makes them ignorant towards their health. Later, it is very difficult for them to deal with severe health conditions. Objective of this research is to give them convenient healthcare service where every rich or poor patient will get their health checked in timely manner. In this situation, patient can connect with the specialist 24 × 7 and also gets notified in unfavourable health condition. By performing various test analysis, we calculated the proposed parameters. Finally, we conclude that our system is able to eliminate network-based intrusion attacks as well as host-based intrusion attacks; the proposed prevention mechanism also automatically recovers trash data from external attacker. Our future work will be to provide attack prevention from different types of network attacks and threats.

References

1. Mestiri H, Kahri F, Bouallegue B, Machhout M (2015) A high-speed AES design resistant to fault injection attacks. *Microprocess Microsyst*
2. Maheswari SU, Usha NS, Anita EAM, Devi KR (2016) A novel robust routing protocol Raed to avoid dos attacks in WSN. In: Proceedings of 2016 international conference on information communication and embedded systems (ICICES)
3. Mukherjee A (2015) Physical-layer security in the internet of things: sensing and communication confidentiality under resource constraints. In: Proceedings of the IEEE, vol 103, no 10, pp 1747–1761
4. Ziegeldorf JH, Morschon OG, Wehrle K (2013) Privacy in the internet of things: threats and challenges, security and communication networks, 1002/sec. 795
5. Choi J, In Y, Park C, Seok S, Seo H, Kim H (2016) Secure IoT framework and 2D architecture for end-to-end security. *J Super Comput* 1227-016-1684-0
6. Jian Z, Zhanli W, Zhuang M (2012) Temperature measurement system and method based on home gateway CN 201110148247
7. Sivagami S, Revathy D, Nithyabharathi L (2016) Smart health care system implemented using IoT. *Int J Contemp Res Comput Sci Technol* 2
8. Samuel RE, Connolly D (2015) Internet of things-based health monitoring and management domain-specific architecture pattern. *Issues Inf Syst* 16:58–63
9. Rahmani A-M, Thanigaivelan NK, Gia TN, Granados J, Negas B, Liljeberg P, Tenhunen H (2015) Smart e-health gateway: bringing intelligence to internet-of-things-based ubiquitous healthcare systems. In: Proceedings of the annual IEEE consumer communications and networking conference, IEEE, NY
10. Agrawal S, Vieira D (2013) A survey on internet of things. *Abakós, Belo Horizonte* 1:78–95
11. <https://economictimes.indiatimes.com/tech/internet/internet-users-in-india-to-reach-627-million-in-2019-report/articleshow/68288868.cms>
12. https://www.downtoearth.org.in/dte-infographics/61322-not_enough_doctors.html

Chapter 11

Intelligent System to Diagnose LBP Using Genetic Algorithm and Support Vector Machine



Mittal Bhatt and Vishal Dahiya

1 Introduction

The field of medical science always an important application area for the implementation of newly evolves technology related to field of reasoning and learning [1–5]. Diagnosing for diseases is an art as it is differing verities of parameters from person to person. To carry out the procedure of diagnosis of diseases, medical practitioner has to identify relationship between reasons and responses and is hardly one to one. Designing intelligent system in domain of medical science for diagnosis will increase capability of system.

1.1 Medical Background

As shown in Fig. 1, human body spine is comprised of 33 vertebrae and its structure is like numbers of bones arranged in stack. As it is depicted in Fig. 1, lower lumbar spine is made up of five vertebrae named as L1–L5. The disks between vertebrae are fibrocartilage pads of cylindrical shapes that lie between the vertebral bodies, responsible for giving flexibility and stability to spine.

Aging is unavoidable parameter of human life; during this period, structure of spine undergoes several changes which affect its functionality. In recent studies, it has been observed that heredity and effects of genetic influences play key role in degeneration of disk [6, 7].

M. Bhatt (✉)
CHARUSAT University, Changa, Anand, India
e-mail: bhattmittal2008@gmail.com

V. Dahiya
Indus University, Ahmedabad, India

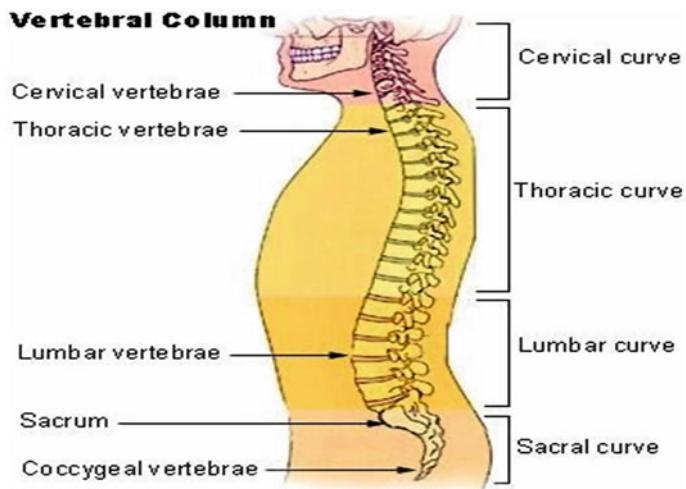


Fig. 1 Spine structure of human body [WIKI2010]

Fig. 2 shows overview of lower spine problems; in western industrialized societies, it was described as “a major health problem” [8]. Out of which disk herniation disk is a disk that ruptures. This allows the jelly-like center of the disk to leak, irritating the nearby nerves. This can cause back pain and spondylolisthesis causes one of the lower vertebrae to slip forward onto the bone directly beneath it. It is a painful condition but treatable in most cases.

1.2 SVM and GA

Under the category of supervised learning algorithms, best fine results were given by SVM, by defining hyperplane, responsible to separate elements with maximum margin. The core part in it is the kernel function that maps samples from training set to generate output of classification using product of matrix. As it is core part of SVM, it highly affects accuracy as it defines mapping of input space into function space [9].

2 Literature Review

LBP falls into the category of disorder, not diseases and is neglected as less important to look after. There are various reasons for LBP to occur like; injury in bones, ligaments; the nerve root visiting legs might be irritated; muscle straining in spine, etc.

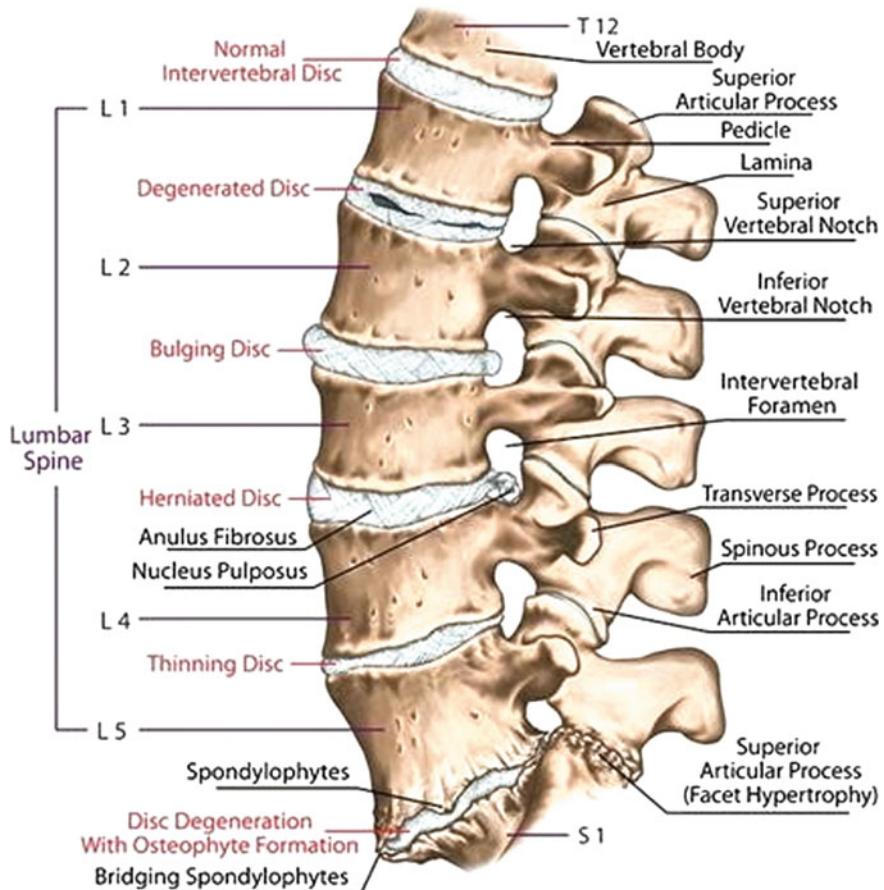


Fig. 2 Degenerative disk problems (<https://www.pinterest.com/pin/437693657523882190/>)

The numbers of studies were done in this area in order to classify lower spine diseases with different machine learning algorithms and techniques. Rule-based fuzzy type-2 system is designed to diagnose degenerative disk diseases [10]. In one of the study the impairments due to motor control a mechanism was diagnosed [11]. One study aimed at performing diagnosis based on dataset build from magnetic resonance images of lower back spine and results in diagnosis of LBP diseases. In one of the studies, lower spine dataset is used from UCI machine learning repository and findings were the most effective feature for making classification of diseases in the best way [12].

3 Proposed System

Stage–1 Preprocessing:

Data is collected from UCI machine learning library, consists of six feature vectors and 310 samples, describing spine condition in disk hernia, spondylolisthesis and normal spine. Under the cleaning of data, noisy values were treated and thereafter normalization of data being done for a good result. Algorithms used Euclidean distances are sensitive to magnitude of data, so all data columns need to be scaled at one specific level. After that, scaled data is splattered into training and testing sets.

Stage–2 Optimization:

From Stage-1, preprocessed data is fed into the optimization stage; under this stage, results were obtained by implementing principal component analysis (PCA) and also GA. Using GA, values for “C” and “Gamma” are optimized and passed into SVM’s classification results improvement.

Stage–3 Classifications:

After the optimization using GA, two meta-heuristics parameters are passed in SVM and results were obtained by using 10-fold cross-validation (Fig. 3; Table 1).

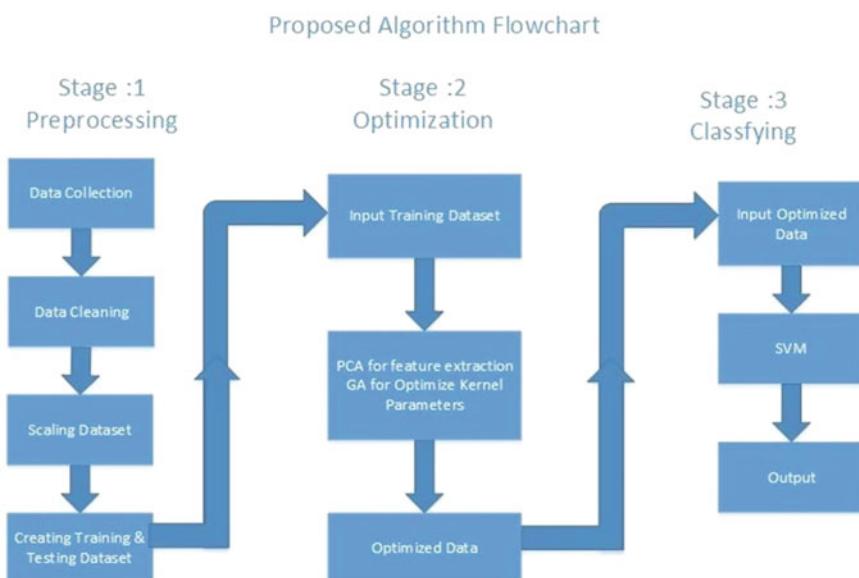


Fig. 3 Flow of proposed system

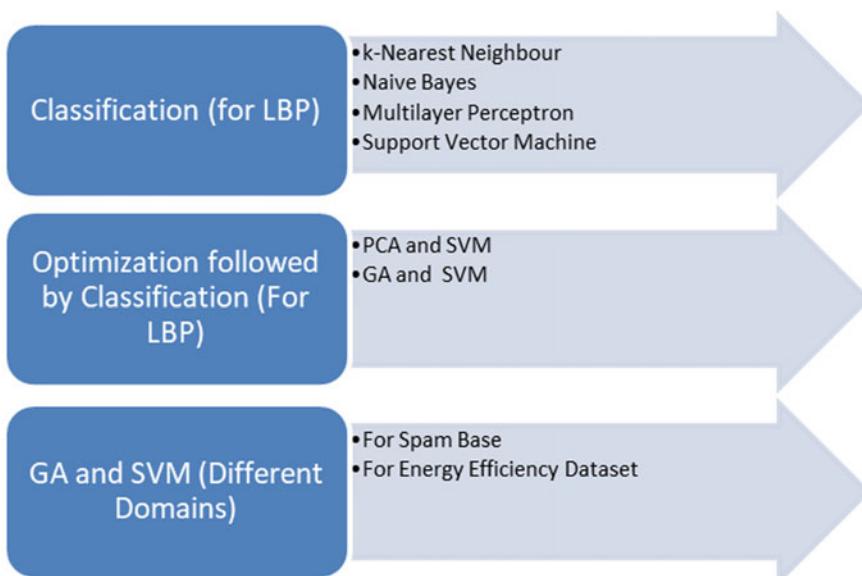
Table 1 Dataset attributes description

Sr. No.	Name of attribute	Description
1	pelvic_incidence	Angle between two lines, one perpendicular to the sacral plate at its mid-point and other connecting this point to the femoral heads axis (26.1–130)
2	pelvic tilt	The orientation of pelvis with respect to the thighbones and the rest of the body (−6.55 to 49.4)
3	lumbar_lordosis_angle	The normal inward lordotic curvature of the lumbar and cervical regions of the human spine (14.0–126)
4	sacral_slope	Angle between sacral plate and a horizontal line (13.4–121)
5	pelvic_radius	Pelvic Radius Angle (PRA) was measured from the PR line to the horizontal (70.1–163)
6	degree_spondylolisthesis	Slipped vertebral body “Spondylo” = vertebrae, “listhesis” = slippage (−11.6 to 419)

4 Results and Validations

Results were obtained by implementing the following sequence of techniques as shown in Fig. 4.

The result is depicted using classification report, as it is visualization of precision, recall, F1 and support score. The metrics are defined by true and false positive

**Fig. 4** List of implementations

and false negative. Here, positive and negative are comprehensive names for the classification contains two classes. When the evaluation of one class value is true; referred as true positive, as is the estimated class.

Receiver operating characteristic (ROC) is also one of the important depictions used to represent results of learning accuracy. It represents reciprocate of true positive and false positive using different probability thresholds plot (Fig. 5).

Out of four classification algorithms, implementation SVM gives best result of 86.27%.

PCA-SVM Hybridization:

As shown in Table 2, features are extracted using PCA and then classification is implemented using SVM, gives the best results with five components and is plotted in Fig. 6, Table 3.

GA SVM hybridization; GA is used to optimize the values of “ c ” and “gamma,” so that SVM performance can be improved. The procedure of GA is as follows:

G Numbers of Generations

PZ Population Size

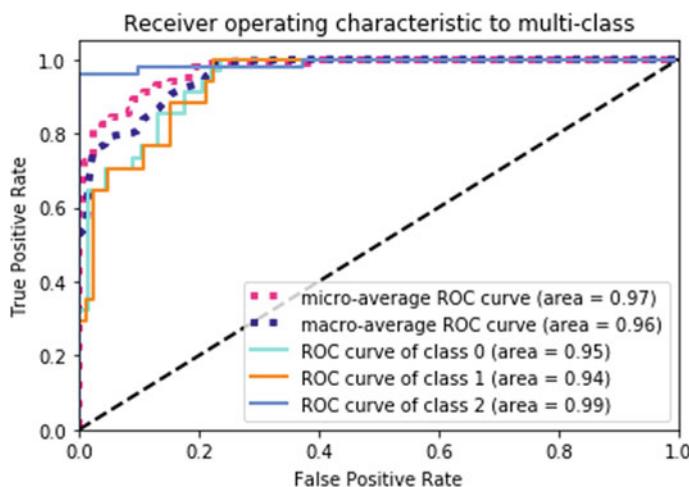
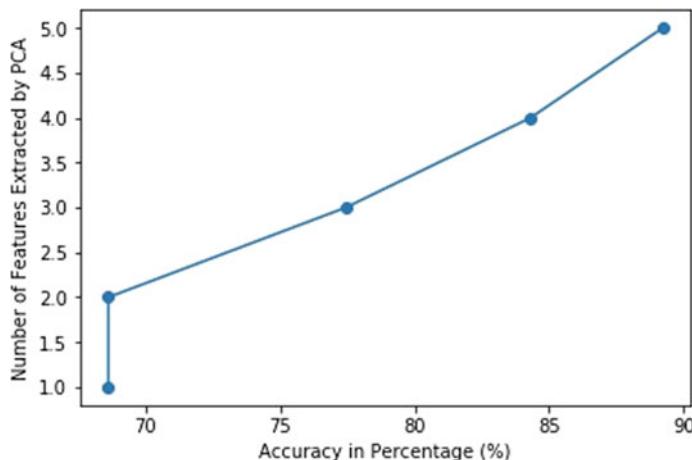


Fig. 5 ROC curve for SVM multiclass classification

Table 2 Different classification algorithms implementation results

Sr. No.	Classification algorithm	Achieved accuracy
1	k -nearest neighbor	81.3725
2	Multilayer perceptron	77.6699
3	Naïve Bayes	83.3333
4	Support vector machine	86.2745

**Fig. 6** Plotting of PCA and SVM hybridization**Table 3** PCA and SVM hybridization with various numbers of components

Sr. No.	Number of columns	PCA and SVM accuracy (%)
1	5	89.2156
2	4	84.3137
3	3	77.4509
4	2	68.6274
5	1	68.6274

Prob_c Probability of Crossover

Prob_m Probability of Mutation

T Tournament selection, choosing three warriors for best optimal value selection from calculation of error.

len String size of gene for obtaining two hyperparameters values.

Step-1: Set G, PZ, Prob_c, Prob_m, len, and T for the tournament selection.

Step-2: Create PZ population randomly of size length.

Step-3: $g = 1$, $pz = 1$.

Step-4: Select two parents through selection.

Step-5: Crossover the parents' genes to get children at prob_c.

Step-6: Mutate children at Prob_m.

Step-7: Calculate fitness of mutated children and save fitness values.

Step-8: Repeat steps 4–7 for $pz/2$ times.

Step-9: Create a new generation of mutated children.

Step-10: $g = g + 1$.

Step-11: Repeat steps 4–10 G times.

Accuracy for 10 Fold SVM Without GA: 0.919048

Accuracy for 10 fold SVM With GA: 0.971429

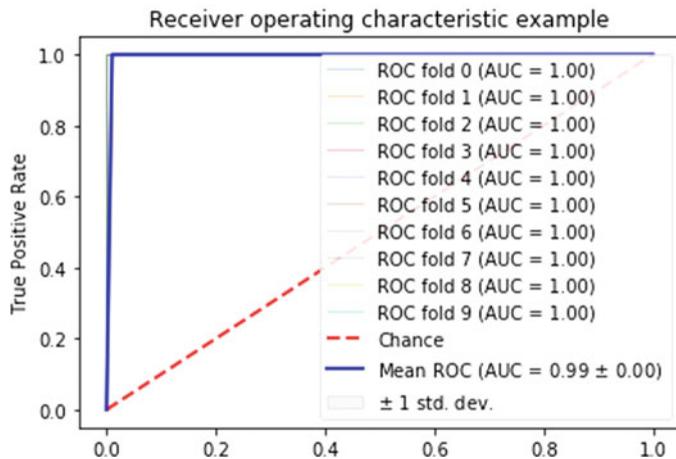


Fig. 7 GA and SVM hybridization

Step–12: Choose the best fitness value from the last generation to keep track of the best chromosome in each generation for final answer.

As shown in Fig. 7, SVM is implemented after passing “c” and “gamma” values obtained from GA gives us the best results.

Now, the model needs to be validated in different domains for its performance evaluation. The domains were taken as classifying mail needs to be spam or not, in which spam base was used from UCI machine library. The domain is taken where heating and cooling requirements in construction of buildings, where energy efficiency dataset is used from UCI machine library. In both domains, the developed model gives the best accuracy till date as 93.15% and 93.73%, respectively as shown in Figs. 7, 8 and 9.

5 Conclusion

The objective of research is to diagnose LBP at its first occurrence before it becomes chronic. The gap of identifying disease that causes LBP is diagnosed with the best performance is achieved from the designed computational model. The performance achievement is validated for two more applications of spam mailing and finding heating and cooling requirements for construction of building and comparison study as follows (Fig. 10).

Accuracy for SVM Without GA: 0.836994

Accuracy for SVM With GA: 0.931536

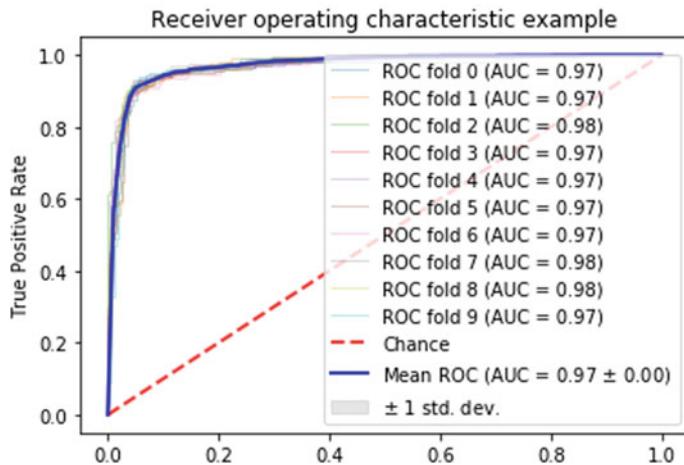


Fig. 8 GA and SVM implementation for spam base

Accuracy for SVM W/O GA: 0.872113

Accuracy for SVM W/ GA: 0.937348

Fig. 9 GA and SVM implementation for energy efficiency dataset

Classification Results on Various Datasets

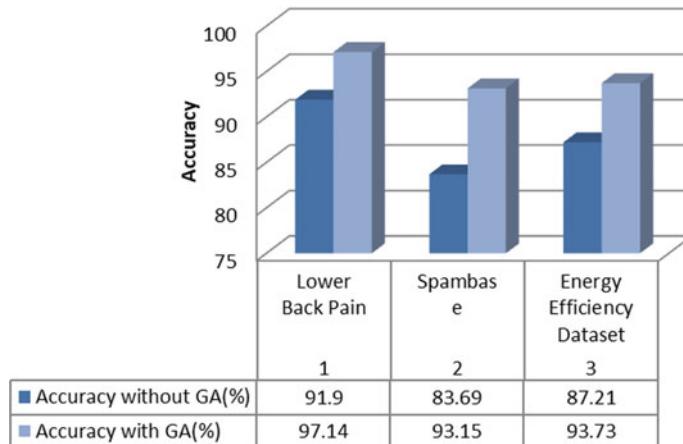


Fig. 10 Comparison of SVM with and without GA

References

1. Gorry G (1973) Computer-assisted clinical decision making. *Methods Inf Med* 12:45–51
2. Robert S, Led L, Lusted B (1959) Reasoning foundations of medical diagnosis. *Science* 130(3366):9–21
3. Nordyke R, Kulikowski C, Kulikowski C (1971) A comparison of methods for the automated diagnosis of thyroid dysfunction. *Comput Biomed Res* 4:374–389
4. Schwartz WB (1970) Medicine and the computer: the promise and problems of change. *N Engl J Med* 283:1257–1264
5. Shortliffe EH, Davis R, Axline SG, Buchanan BG, Green C, Cohen N (1975) Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system. *Comput Biomed Res* 8(4):303–320
6. Battie C, Haynor R, Fisher D, Gill K, Gibbons E, Videman T (1995) Similarities in degenerative findings on magnetic resonance images of the lumbar spines of identical twins. *J Bone Joint Surg Series A* 77:1662–1670
7. Battie C, Videman T, Parent E (2004) Lumbar disc degeneration: epidemiology and genetic influences. *Spine* 29:2679–2690
8. Urban J, Roberts S (2003) Degeneration of the intervertebral disc. *Arthritis Res Ther* 5:120–130
9. Bhavsar H, Ganatra A (2014) Increasing efficiency of support vector machine using the Novel Kernel function: combination of polynomial and radial basis function. *Int J Adv Comput Theory Eng (IJACTE)* 3(5):17–24
10. Rahimi S, Damirchi D, Fazel M, Izadi M (2013) Type-2 fuzzy hybrid expert system for diagnosis of degenerative disc disease. *Amirkabir Univ Technol (Tehran Polytechnic)* 45(2):53–62
11. Sullivan P (2005) Diagnosis and classification of chronic low back pain disorders: maladaptive movement and motor control impairments as underlying mechanism. Body-logic physiotherapy, Australia School of Physiotherapy, Curtin University of Technology, Perth, Western Australia
12. Goankar A, Kulkarni R, Caytiles R, Iyengar N (2017) Classification of lower back pain disorder using multiple machine learning techniques and identifying degree of importance of each parameter. *Int J Adv Sci Technol* 105(2):11–24
13. Toprak A, Koklu N, Ozcan R (2017) Comparison of classification techniques on energy efficiency dataset. *Int J Intel Syst Appl Eng* 5(2):81–85
14. Cortes C, Vapnik V (1995) Support vector networks. *Mach Learning* 20:273–297
15. Novakovic J, Sinisa R (2011) Classification performance using principal component analysis and different value of the ratio R. *Int J Comput Commun Control* 6(2):317–327

Chapter 12

Area Analysis for Dengue Prediction



Aniket Milind Banginwar^{ID}, Shreyas Nanaware^{ID}, Kalpita Bhagat^{ID}
and Deepshikha Chaturvedi

1 Introduction

Human life has been the most precious asset of all time. In the past six decades, there have been many advances in medical sciences that have helped us to reduce the human mortality rate from 19.5 to 8.1 per 1000 individuals. The cures of so many diseases are not known yet. Dengue is one such disease. Dengue is a viral mosquito-borne disease. The numbers of cases of dengue have increased from 908 cases in 1955–1960 to 925,896 cases in 2000–2007. The current statistics are about 390 million cases are reported every year. About 20,000 deaths per year occur worldwide due to the dengue virus. The dengue virus is spread through a particular mosquito species, *Aedes aegypti*, the yellow fever mosquito.

The overall purpose of this study is to exploit the power of machine learning in order to help prevent a potential dengue epidemic. National Oceanic and Atmospheric Administration (NOAA) has provided with a dataset that contains various parameters like climate variables, population statistics, satellite data, etc. that can be used to figure out the correlation between the parameters and dengue cases. The predictions will help foretell the number of cases possible in a given week, and if the number of cases exceeds the threshold, then the authorities can be intimated about the same.

A. M. Banginwar (✉) · S. Nanaware · K. Bhagat · D. Chaturvedi
Shah and Anchor Kutchhi Engineering College, Mumbai 400088, India
e-mail: aniketbanginwar@gmail.com

S. Nanaware
e-mail: shreyas368@outlook.com

K. Bhagat
e-mail: kalpitab98@gmail.com

D. Chaturvedi
e-mail: deepshikha.chaturvedi@sakec.ac.in

This will help them take early action and reduce the impact or avoid the catastrophe altogether.

2 Literature Review

Dengue, a viral disease, is spread throughout the world with the number of cases increasing with each year. Due to the risk of severe symptoms and death, it is important to predict the outbreak of this disease. Many machine learning models have been used and developed to help in this prediction. Roziqin et al. [1] show the use of two regression models, Monte Carlo linear regression and dynamic polynomial regression for this prediction on the basis of the historical data on the dengue fever cases and weather data. Predictions obtained showed an error level of 1%. Mathulamuthu et al. [2, 3] make use of k -means clustering and ordinary least squares (OLS) to study the correlation between the dengue cases and the climate variables. Both the papers show that the use of clusters gives a good predictive analysis. Mathulamuthu et al. [4] make use of dimensionality reduction technique of manifold learning to reduce the dimensions of the data making it easier to model the data. This helps in increasing the accuracy of the model and also makes the models time efficient.

This paper makes use of 19 machine learning algorithms including 13 linear algorithms, four ensemble algorithms, support vector regression (SVR) and partial least squares (PLS) regression. These algorithms are applied to the input dataset and the results are compared to find the best algorithm, that is, the algorithm which gives the least error.

3 Methodology

Through extensive research made on the topic of dengue prediction, a number of factors are considered to have an influence on the number of dengue cases that can occur in a particular area in a particular type of season and environment. These influencing factors were categorized as input variables. The output variables, on the other hand, represent the number of cases. The five steps to be followed in the study are:

- a. Obtaining the dataset (NOAA)
- b. Data preprocessing and cleaning
- c. Training the machine learning models
- d. Testing and validation
- e. Prediction.

3.1 Dataset

The first step is obtaining the dengue dataset from the NOAA website. Table 1 gives the detailed dataset obtained from the NOAA website. The table shows the attributes' names and descriptions of the environmental data collected from Iquitos, Peru. This data is from a variety of sources (ground observations, remote sensing

Table 1 Summarized results of testing

Comparison table	Mean absolute error	Root mean squared error	Mean squared log error	Median absolute error
ARD regression	5.791557273	6.537730134	1.191609048	5.498055602
Bagging regression	5.525000000	6.784492121	1.075361645	4.700000000
Bayesian ridge	5.282899305	6.047789766	1.105809595	5.333937034
Elastic net	4.828066559	5.608159483	1.080350409	4.567938165
Extra trees regression	3.591666667	4.426622866	0.733702881	3.112500000
Gradient boosting	4.095899395	5.192546423	0.844832934	3.097042575
Huber regression	3.314153133	4.049159863	0.688257765	2.955774261
LARS	4.660008684	5.294884507	1.026983744	4.526363309
Lasso LARS	4.604639241	5.246398941	1.023420991	4.206993719
Orthogonal matching pursuit	4.778118646	5.491883205	0.974145069	4.523800025
Passive aggressive regression	6.471917148	14.3501777	1.32350095	3.918002946
PCA with lasso regression	4.796127543	5.584370616	1.072381791	4.639407276
PCA with linear regression	5.202343023	6.076795229	1.144717698	4.841966396
PCA with PLS	5.271224265	6.058804758	1.121162607	5.056702326
PCA with ridge regression	5.268364346	6.055624452	1.120821164	5.057795602
Perceptron	3.466666667	4.833908012	0.94581397	2.000000000
Polynomial with PCA	4.840138013	5.579234791	1.015917793	4.635037646
Random forests	3.541666667	4.263542541	0.659855388	3.060000000
Support vector regression	2.956933922	4.014756241	0.573898229	2.167352887

The bold highlights that particular error (number) is the lowest for that metric (column)

and reanalysis). The dataset contains weekly data for the city of Iquitos, Peru. The time period of data collection is from 1990 to 2010 dengue seasons. Features are as follows:

Date indicators

- a. week_start_date—Date is given in the yyyy-mm-dd format.

NOAA's GHCN daily climate data weather station measurements

- a. station_max_temp_c—Maximum temperature
- b. station_min_temp_c—Minimum temperature
- c. station_avg_temp_c—Average temperature
- d. station_precip_mm—Total precipitation
- e. station_diur_temp_rng_c—Diurnal temperature range.

PERSIANN satellite precipitation measurements (0.25 × 0.25 degree scale)

- a. precipitation_amt_mm—Total precipitation.

NOAA's NCEP Climate Forecast System Reanalysis measurements (0.5 × 0.5 degree scale)

- a. reanalysis_sat_precip_amt_mm—Total precipitation
- b. reanalysis_dew_point_temp_k—Mean dew point temperature
- c. reanalysis_air_temp_k—Mean air temperature
- d. reanalysis_relative_humidity_percent—Mean relative humidity
- e. reanalysis_specific_humidity_g_per_kg—Mean specific humidity
- f. reanalysis_precip_amt_kg_per_m2—Total precipitation
- g. reanalysis_max_air_temp_k—Maximum air temperature
- h. reanalysis_min_air_temp_k—Minimum air temperature
- i. reanalysis_avg_temp_k—Average air temperature
- j. reanalysis_tdtr_k—Diurnal temperature range.

Satellite vegetation—Normalized difference vegetation index (0.5 × 0.5 degree scale) measurements (NDVI—NOAA's CDR Normalized Difference Vegetation Index)

- a. ndvi_se—Pixel southeast of city centroid
- b. ndvi_sw—Pixel southwest of city centroid
- c. ndvi_ne—Pixel northeast of city centroid
- d. ndvi_nw—Pixel northwest of city centroid.

3.2 Data Preprocessing

The dataset obtained is a raw data and needs to be transformed into a consistent and noise-free data. The Inter-Quartile Range (IQR) range of all the attributes in the dataset is first calculated as given in Eq. (1).

$$\text{IQR} = Q3 - Q1 \quad (1)$$

Here, $Q1$ and $Q3$ are the first and the third quartile, respectively. The low outliers are below the value $Q1 - 2.1 * \text{IQR}$ and the high outliers are above the value $Q3 + 2.1 * \text{IQR}$. All the outliers found out are removed and the dataset is then checked for null values. The NaN values are filled with the last known values of the corresponding columns. After this preprocessing, we get the final dataset which is used for training purposes.

3.3 Machine Learning Models (Training)

The preprocessed dataset is used for training the machine learning algorithms to develop the machine learning models. For this purpose, 80% of the dataset, called the training dataset, is used. The deciding parameters of the algorithms are modified with respect to the input parameters to make the predictions more accurate and efficient. The details of these parameters are given below for each of the 19 algorithms.

In regression algorithms, to improve the accuracy of prediction, the principal component analysis (PCA) method is applied before performing regression. Linear dimensionality reduction uses singular value decomposition (SVD) of the dataset and casts it to a lower dimensionality level. LAPACK implementation is generally used.

In bagging regression, the number of base estimators is set to 20 and all processors are used. In elastic net, the ratio of $L1:L2$ regularization is set to 0.7. This means 70% of the regularization used is $L1$ and 30% is $L2$. Regularization is used to reduce the complexity of the model. It does this by penalizing the loss function. $L1$ regularization performs feature selection.

In gradient boosting, the number of boosting stages to perform is set to 1500. Gradient boosting is fairly robust to overfitting, so a large number usually results in better performance. The loss function is set to “lad” (least absolute deviation) which is a highly robust loss function solely based on order information of the input variables. The learning rate is set to 0.01 as this is found to be optimal. The minimum number of samples required to split an internal node is set to 7.

In LARS, the target number of non-zero coefficients is set to 1. In Lasso LARS, the alpha $L1$ regularization is set to 0.1. In Passive Aggressive Regression, the maximum number of iterations is set to 10 as the model converges within 10 iterations. In

lasso regression, the regularization parameter alpha is set to 0.5. This means the regularization term is halved.

In the case of single-layer perceptron, the maximum number of iterations is set to 9. Tolerance is set to 0. The stopping criteria for a perceptron are when the loss exceeds (previous loss—tolerance). When tolerance is set to 0, the perceptron stops strictly when the loss exceeds the previous loss. For polynomial features, we set the degree of polynomial to 3. Going further does not improve the accuracy of the model significantly and may cause overfitting.

In the case of random trees, the number of trees in the forest is set to 100. The oob_score parameter is set to true. The max features parameter is set to use 60% of the number of features in the training dataset. The minimum number of samples required to be at a leaf node is set to 22. A split point at any depth will only be considered if it leaves at least 22 training samples in each of the left and right branches. This may have the effect of smoothing the regression model. It is set to use all processors.

In support vector regression, the kernel used is radial basis function. This is used since it can be used effectively to fit nonlinear data. Penalty parameter C of the error term is set to 59. This set the margin used to fit the kernel to the given data. A larger penalty term results in a smaller margin and greater precision. This, in turn, affects the accuracy of our model.

3.4 Testing and Validation

Here, 20% of the dataset, called the testing set, is used to test the performance of the predictor model. It provides an unbiased evaluation of the final model fit on the training dataset. The error of all the models is calculated. There are various metrics used to see how the algorithms have fit the data. The model assessment criteria that we used are mean absolute error (MAE), median absolute error (MedAE), root mean squared error (RMSE), mean squared log error (MSLE).

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (2)$$

$$\text{MedAE}(y, \hat{y}) = \text{median}(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n|) \quad (3)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (4)$$

$$\text{MSLE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (\log_e(1 + y_i) - \log_e(1 + \hat{y}_i))^2 \quad (5)$$

4 Results and Analysis

We have 13 linear models, four ensemble models, PLS regression and support vector regression (SVR) in total. Here are the results of the testing and validations done on every algorithm (Fig. 1).

The implementation of the prediction for the NOAA's Iquitos dataset gives support vector regression as the best algorithm with error (mean absolute error) of just 2.956933922.

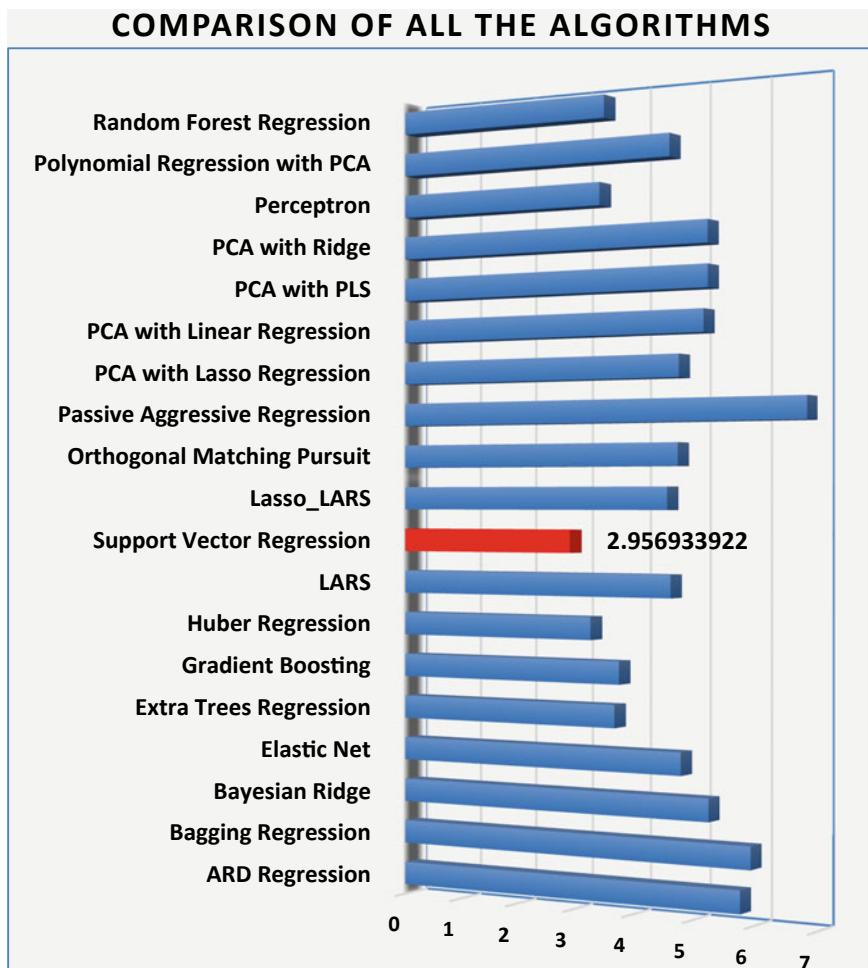


Fig. 1 Comparison of all algorithms on the basis of mean absolute error. Support vector regression with least MAE is the clear winner

4.1 Prediction

After testing, the accuracy of the models is calculated and compared. This gives us the model with the best prediction which is then used for predicting the number of dengue cases. As stated in the above analysis, support vector regression yields the best performance as compared to other models. So, support vector regression will be the preferred model for this particular dataset.

4.2 Feature Coefficient/Importance Analysis

We will be doing feature analysis for linear models and PLS regression and feature importance for ensemble models.

This analysis cannot be conducted for polynomial regression because there are 1771 features. It cannot be conducted for support vector regression because we are not using a linear kernel. It cannot be conducted for bagging regression because the feature importance cannot be calculated for bagging regression (Table 2).

Looking at the feature dependency of these algorithms, we can observe that many of them are primarily dependent on “specific humidity” and “precipitation amount.”

Table 2 Models and their feature dependence

Model name	Feature on which prediction majorly depends
ARD regression	reanalysis_specific_humidity_g_per_kg
Random forests	reanalysis_specific_humidity_g_per_kg
Passive aggressive regression	reanalysis_max_air_temp_k
Extra trees regression	weekofyear
Bayesian ridge regression	reanalysis_specific_humidity_g_per_kg
Huber regression	ndvi_se
Gradient boosting	reanalysis_specific_humidity_g_per_kg
Elastic net regression	reanalysis_precip_amt_kg_per_m2
Orthogonal matching pursuit	reanalysis_precip_amt_kg_per_m2
Lasso LARS regression	reanalysis_precip_amt_kg_per_m2
LARS regression	reanalysis_precip_amt_kg_per_m2
Linear regression	reanalysis_air_temp_k
Lasso regression	reanalysis_specific_humidity_g_per_kg
Ridge regression	reanalysis_specific_humidity_g_per_kg
PLS regression	reanalysis_precip_amt_kg_per_m2
Perceptron	ndvi_ne, ndvi_nw, ndvi_se, ndvi_sw

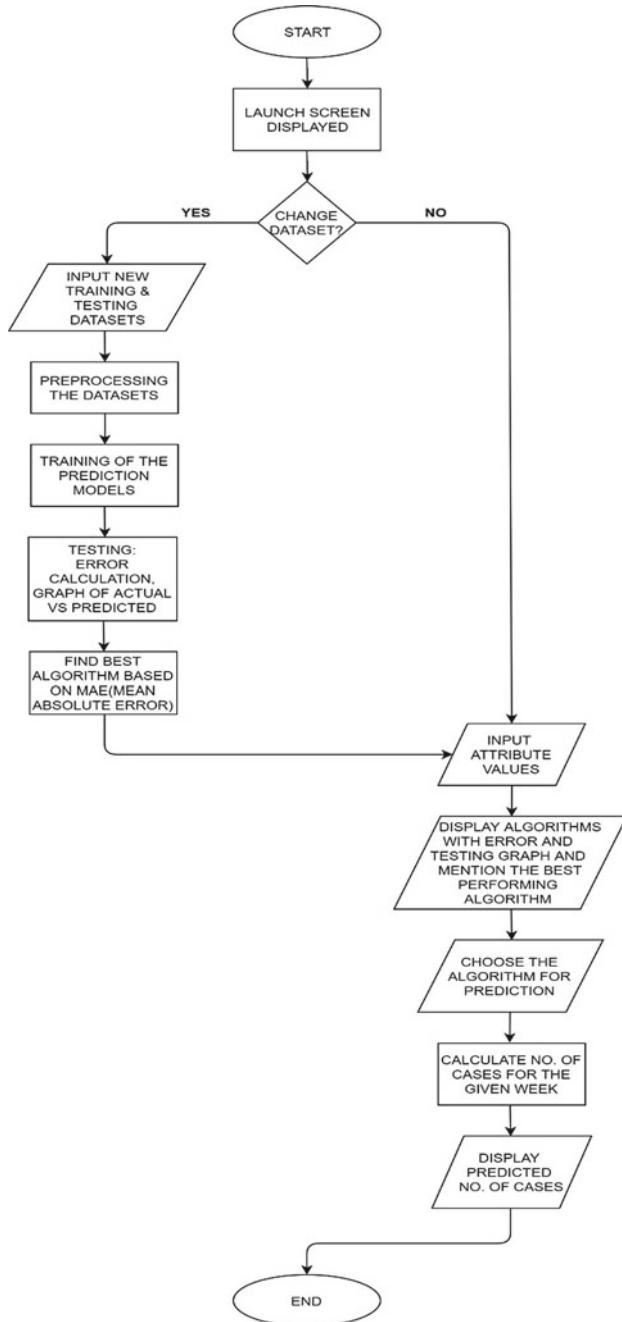


Fig. 2 Flowchart demonstrating the flow of the proposed system

4.3 Proposed Model Flow

Once the launch screen is displayed, we ask the user if they want to update or change the datasets (Fig. 2).

If the user chooses to do so, the proposed software inputs the datasets from the address specified by the user. Then we preprocess the datasets to remove noise and fill the missing values such that the last observation is carried forward. The models are trained on the new datasets. While testing, we calculate the error (MAE) as well as create a graph of the actual versus predicted values on the testing dataset. We determine the best algorithm based on mean absolute error values.

If datasets are not changed or after changing the datasets, we input the attribute values. We then display the list of available algorithms with error values and testing graphs. Once the user chooses the algorithm, the number of cases for the given week is calculated and displayed.

5 Conclusion

The purpose of this study is to choose an efficient prediction model to predict the number of dengue cases occurring in the city of Iquitos, Peru, for a particular week. The proposed method is believed to assist in informing the city's medical centers and strengthening the current dengue management practice of the city. The study shows that the support vector regression model gives the lowest mean absolute error of 2.9569339.

References

1. Roziqin MC, Basuki A, Harsono T (2016) A comparison of montecarlo linear and dynamic polynomial regression in predicting dengue fever case. In: 2016 international conference on knowledge creation and intelligent computing (KCIC), Manado, pp 213–218
2. Mathulamuthu SS, Asirvadam VS, Dass SC, Gill BS, Loshini T (2016) Predicting dengue incidences using cluster based regression on climate data. In: 2016 6th IEEE international conference on control system, computing and engineering (ICCSCE), Batu Ferringhi, pp 245–250
3. Mathulamuthu SS, Asirvadam VS, Dass SC, Gill BS (2016) Cluster based regression model on dengue incidence using dual climate variables. In: 2016 IEEE conference on systems, process and control (ICSPC), Bandar Hilir, pp 64–69
4. Mathulamuthu SS, Asirvadam VS, Dass SC, Gill BS (2017) Predicting dengue cases by aggregation of climate variable using manifold learning. In: 2017 IEEE international conference on signal and image processing applications (ICSIPA), Kuching, pp 535–540

Chapter 13

Crowdsourcing for Urban Laborers and Time Optimization



Nihit Natu, Ayush Gupta, Viraj Mahadik and Amiya Kumar Tripathy

1 Introduction

Myriad of people travels to urban cities for job opportunities. It has been found that there are large groups of laborers migrating from rural areas to urban cities in search of various job opportunities. They are unaware of the job opportunities and even being aware they spend a hefty amount of time in finding the jobs based on their skillset. After finding an appropriate job, location traversing to the job location is a serious problem faced by the workers as cheaper routes or modes of transport are unknown for them in the new city. Since most of them are daily workers, they tend to exhaust most of their wages by taking expensive travel routes. A contractor relationship model for the worker can take away the maximum part of the worker's daily wage by absorbing a hefty amount as a contractor compensation.

The world is currently moving toward smarter and better predictions, and hence, Internet and technological giants are therefore developing algorithms to provide better throughput to their clients to sustain global user base. Recommendation systems (RS) are currently shaping the online shopping services, etc. There are two types of RS models, content-based filtering and collaborative filtering. Content-based methods [1] use the user's history and a rating method to rate jobs and the recommendation is done based on the user's profile preferences. Matrix factorization (MF) and restricted

N. Natu (✉) · A. Gupta · V. Mahadik · A. K. Tripathy

Department of Computer Engineering, Don Bosco Institute of Technology, Mumbai, India

e-mail: nihit97natu@gmail.com

A. Gupta

e-mail: guptaayush7232@gmail.com

V. Mahadik

e-mail: vmakadik7@gmail.com

A. K. Tripathy

e-mail: tripathy.a@gmail.com

Boltzmann machines (RBM) are the further two techniques used by content-based filtering. MF makes use of latent vectors of the user and the items from the rating matrix and captures the interaction between them [2].

1.1 *Recommendation Systems*

Recommender systems are commonly used in social networking platforms like Facebook and Netflix. The most commonly used algorithms for building the recommender systems are restricted Boltzmann machine (RBM) and autoencoders. Various papers related to social media recommendations helped to get in-depth working of their recommender engine. Most of them were based on comparing the profiles of various users and finding similarities between them. Pearson's similarity coefficient was commonly used and forms the fundamental for finding the similarity between two profiles [3]. Content- and collaborative-based filtering were the two most significant approaches used. Considering the low efficiency provided by the two approaches, real Boltzmann machine proved to be suitable for the system [4, 5]. Matrix factorization is another essential component of recommender systems. Matrix factorization is a class of collaborative filtering algorithms used in recommender systems. Matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices [6].

1.2 *Location Services*

Location services are used to provide cost-efficient travel routes to the laborers. Along with cost efficiency, the travel time should also be optimized by finding the shortest route between the two nodes. There are various algorithms used for calculating the shortest path between two nodes. The focus while calculating these paths was on the cost factor. Dijkstra's algorithm was fundamental for calculating the cost between nodes although alternations were needed based on our system requirement. Implementing the new feature in the already existing Google Map API's by adding our data based on the local transport services available in a region was the main task that needed extensive research [7].

2 Model Overview

Basically, our system is split into two sections: (1) recommendation systems for providing appropriate jobs based on the worker's profile and (2) location services for providing cost-efficient routes for traversing from current location to the job location. Figure 1 shows the prescribed architecture for the system.

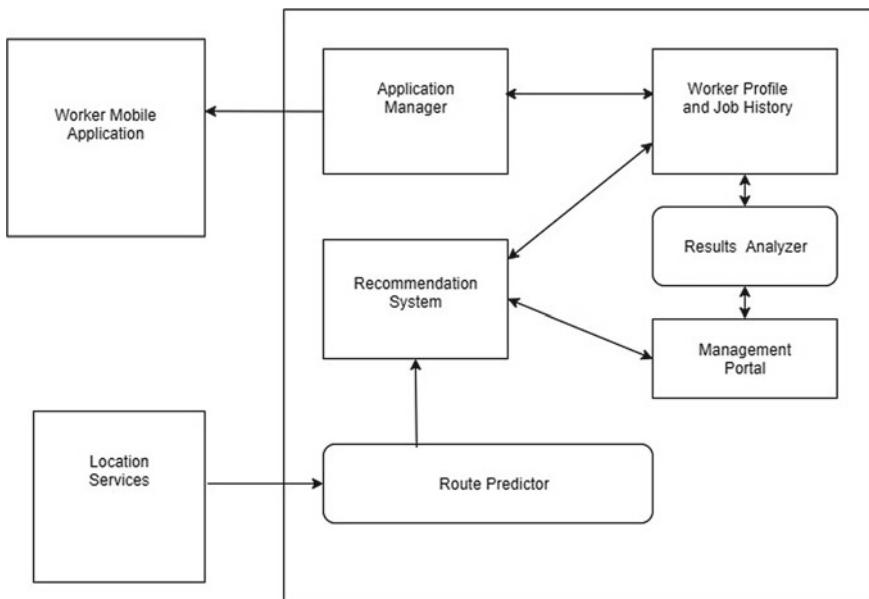


Fig. 1 Architecture of the system. The application will display the jobs and suggested routes to the user. The backend consists of recommender systems and location services

The recommendation system algorithms need two sets of inputs. One is the user's dataset which primarily consists of user id, skillsets, and location and other is the jobs dataset which consists of job id and the rating given by a user to that job. These datasets need to be combined to form a matrix structure with the user in its rows and jobs representing the columns. The cells of the matrix will be filled with the ratings given by the user for that job and the jobs not rated with will have a null value.

2.1 Matrix Factorization

The feedback is given by the user to a certain job done by him (ratings for the job range from one to five), and this feedback is shown in the form of a matrix, where the rows are used to represent the workers (users), while each column represents different jobs. The matrix will be sparse since not everyone is going to do all the jobs present in the system database.

Matrix factorization takes into consideration of the implicit feedback from the user, information that is not directly given but can be derived by analyzing user behavior. Using this strength, we can estimate if a user is going to like a movie that (he/she) never saw. And if that estimated rating is high, the system can recommend that job to the worker [6] (Fig. 2).

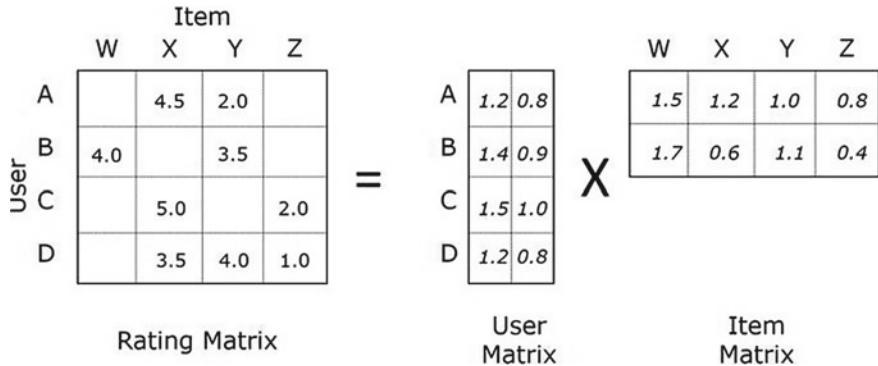


Fig. 2 Idea behind MF. Matrix A has dimensions (m, n) and can also be considered a dot product between dimensions (m, k) and (k, n)

The concept of matrix factorization can be written mathematically in the following way:

$$\hat{r}_{ui} = q_i^T p_u \quad (1)$$

where \hat{r} represents the rating matrix, q and p represent the user and item matrix, respectively.

Then we can create an objective function with respect to q and p , which are (m, k) and (k, n) matrices. The main objective of the matrix factorization is to minimize the objective function.

$$\min_{q, p} \sum_{(u, i) \in k} (r_{ui} - q_i^T p_u)^2 + \sigma (||q_i||^2 + ||p_u||^2) \quad (2)$$

where u and i are used to represent the matrix index for the user and item matrix, respectively.

The term on the right is the regularization term, and this is added since we do not want the decomposed matrix q and p to over-fit to the original matrix. Since the goal is to generalize the previous ratings in a way that predicts future, unknown ratings, we should not over-fit our model [6].

One obvious method to find matrix q and p is the gradient descent method. Since we have the loss function defined, take the partial derivative respect to q and p to optimize those values.

There is also a bias associated with some of the jobs. There might be some users that won't rate a particular job than a designated rating, and hence, the bias is needed to be added. If we add the bias, the original equation would be given by:

$$\hat{r}_{ui} = q_i^T p_u + b_i + b_u \quad (3)$$

where b_i and b_u are the bias terms for the items and users entities, respectively. The objective function can also be altered based on the bias added [6].

2.2 Restricted Boltzmann Machines (RBM)

RBM is the first stage of a recommendation engine. Figure 3 shows the basic architecture behind an RBM.

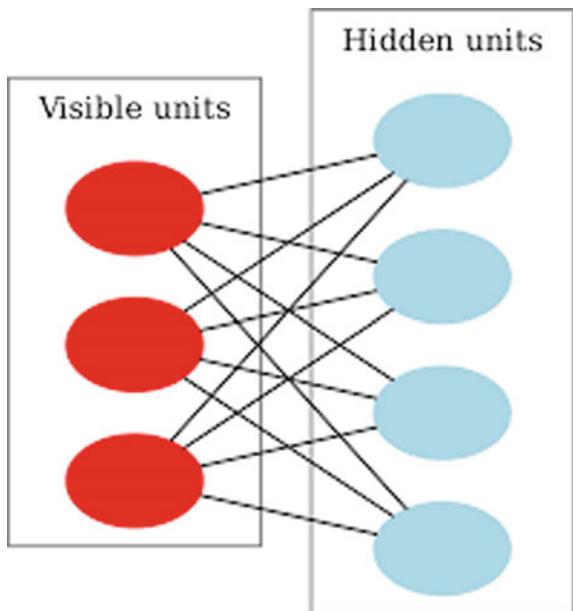
Boltzmann machines (BMs) are a particular form of log-linear Markov random field (MRF), i.e., for which the energy function is linear in its free parameters and to make them powerful enough to represent complicated distributions, and we consider that some of the variables are never observed. By having more hidden variables, we can increase the modeling capacity of the Boltzmann machine (BM). Restricted Boltzmann machines further restrict BMs to those without visible-visible and hidden-hidden connections [8]. RBM tries to find the hidden nodes representing the features apart from the visible features. The hidden nodes can be calculated from the visible nodes and vice versa. Bernoulli's principle is used to calculate hidden nodes values, and it can only take values between 0 and 1.

The hidden node can be calculated by:

$$p(h = 1|v) = \sigma(W^T v + b) \quad (4)$$

Fig. 3 RBM representation.

It consists of visible and hidden nodes, and all the nodes are connected with each other



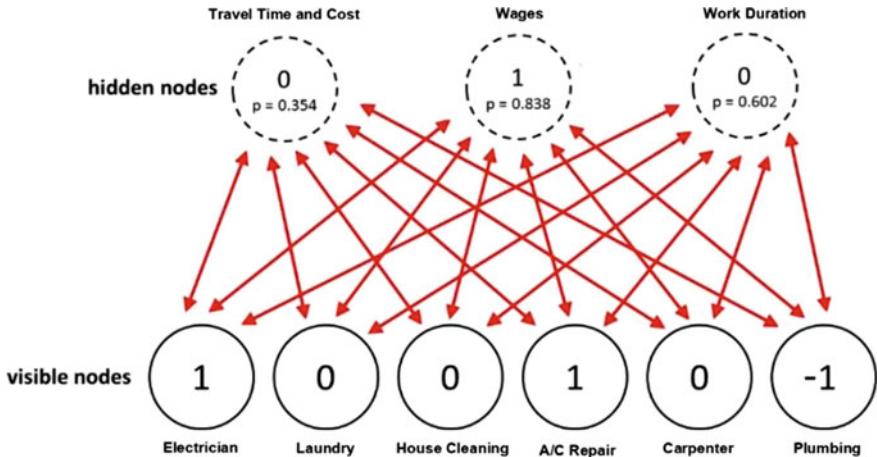


Fig. 4 An example of working of restricted Boltzmann machine for the recommendation of jobs based on the characteristics of the jobs. The probabilities of worker choosing a job are calculated before recommending

The visible can be recalculated after obtaining the hidden node, and it is given by:

$$p(v=1|h) = \sigma(W^T h + c) \quad (5)$$

where h hidden node, v is visible node, W is weight matrix, b and c are biases, and σ is learning constant (Fig. 4).

Examination of the data tells that there are three basic types of categories that worker considers while taking a job: salary (Type A), those who like travel (Type B), and work duration (Type C). There are variations of these types of people. RBM will figure out that there are, in fact, three kinds of people. These three types of people, A, B, C , get encoded as $(1, 1, 0)$, $(1, 0, 1)$, and $(0, 1, 1)$, respectively. And if you feed an encoding such as $(1, 0, 1)$ to RBM, it will predict which job will be liked: $0, 0, 1, 1, 0, 0$.

For RBMs, S consists of the set of visible and hidden units. However, since they are conditionally independent, one can perform block Gibbs sampling [8]. In this setting, visible units are sampled simultaneously given fixed values of the hidden units. Similarly, hidden units are sampled simultaneously given the visible [9]. A step in the Markov chain is thus taken as follows:

$$h^{n+1} = \text{sigm}(W' v^n + c) \quad (6)$$

$$v^{n+1} = \text{sigm}(W h^{n+1} + b) \quad (7)$$

where $h(n)$ refers to the set of all hidden units at the n th step of the Markov chain. The graphical representation of the same is given in Fig. 5.

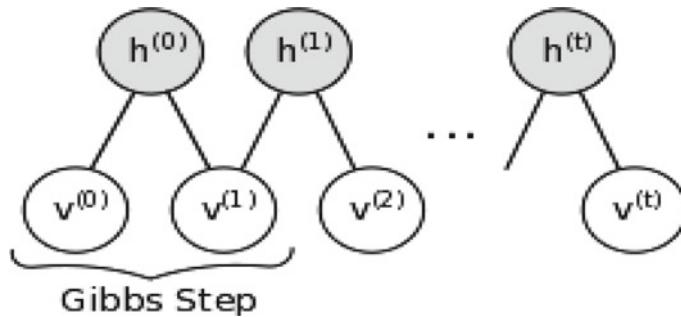


Fig. 5 Representation of the Gibbs sampling in RBM

2.3 Deep Learning Model for Recommender Systems

The deep learning model helps to enhance the efficiency of the recommender systems. It functions in two layers. The first layer is the RBM functioning, and the second layer forms the ranking systems to filter the recommended jobs. The representation of deep learning model is shown in Fig. 6.

The model consists of two layers: candidate generation and ranking system. The candidate generation layers filter the input dataset by shortlisting the selected jobs that can be chosen by the worker. It considers the worker's history to shortlist the candidates, and it passes this data to the next layer. The ranking system gives ranking to the shortlisted jobs by considering the features associated with each job like high pay and less travel. It also considers the user's history along with the features associated with those jobs and compared to the currently available jobs to improvise the shortlisted candidates and enhance the recommendations [10, 11].

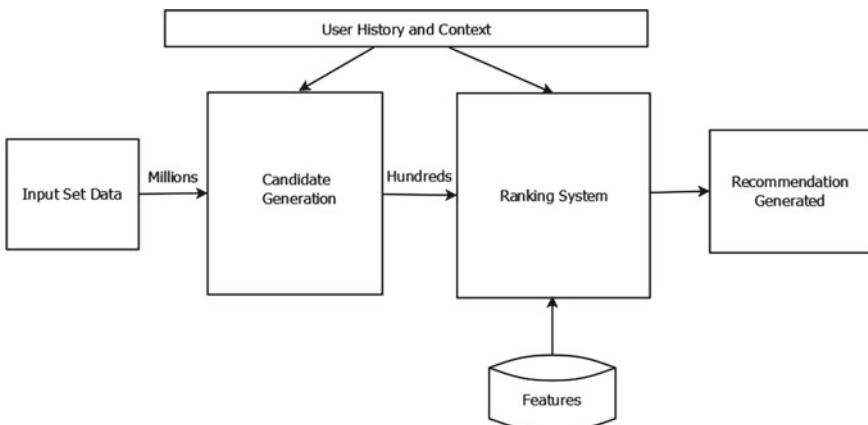


Fig. 6 Deep learning model. It consists of two layers, candidate generation and ranking systems that help to improve the existing recommendations

Machine-learned ranking (MLR) is the application of machine learning, typically supervised, semi-supervised, or reinforcement learning, in the construction of ranking models for information retrieval systems. Training data consists of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment for each item. The ranking model's purpose is to rank, i.e., produce a permutation of items in new, unseen lists in a way which is similar to rankings in the training data in some sense [12]. The system uses ES-Rank in order to filter the recommended jobs and gives rank to each job based on the worker's profile [13–15].

3 Location Services

The next task is to provide the workers with the different transit modes to go the workplace. The route should be chosen in such a way that it helps the workers to reach the place in the shortest time and with maximum compensation. The android application of our system will suggest the users the shortest route with the compensation needed for different transit modes.

For reaching from source to any desired place, we heavily depend on the map's technology unlike any classical methods of routing through any peers or through asking pedestrian or by following the street notations, etc. This may be a way to reach the destination if the distance from source to destination is less, but it becomes a serious drawback if the distance is large and the possible nodes or routes are quite large [16]. This is overcome by using any digital maps' technology which is in current demand right now and people mostly rely on this technology in urban areas, but in rural areas, the use of it is seeing a high growth in terms of its usage [17].

Figure 7 illustrates the current location display of the worker. The current location further helps in determining the local transport modes around that location. The information can also be used by the recommendation system to suggest jobs closer to the worker's location. This feature enables in creating a virtual schedule for the worker.

Figure 8 shows the list of various rickshaw stands, and a worker can use to go to the destination place. The rickshaw stands help workers to take cheaper transport routes. For convenience, our system also maps the nearby available hospitals as it is needed in case of any injury. Our system also has the details of restaurants that are low-cost ones as worker cannot afford costly restaurants.

4 Conclusion

The main purpose of this paper is to ease the worker's load of finding jobs based on their skillset along with fare wages for their work. This will help the workers to increase their daily compensation and motivate them to work harder. The travel costs

Fig. 7 Current location of the worker

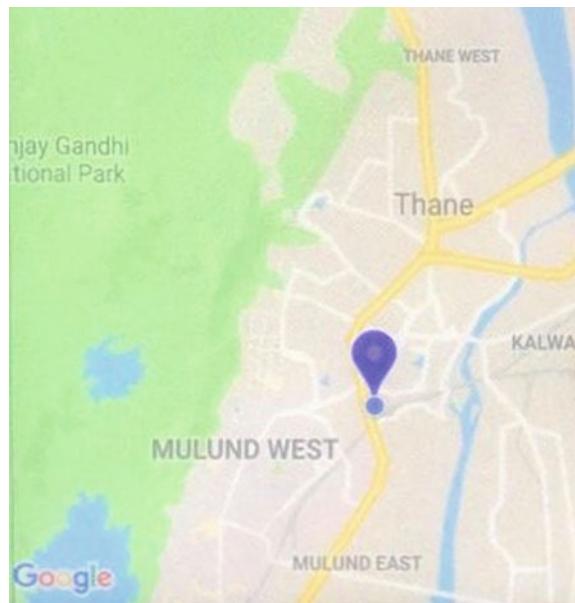
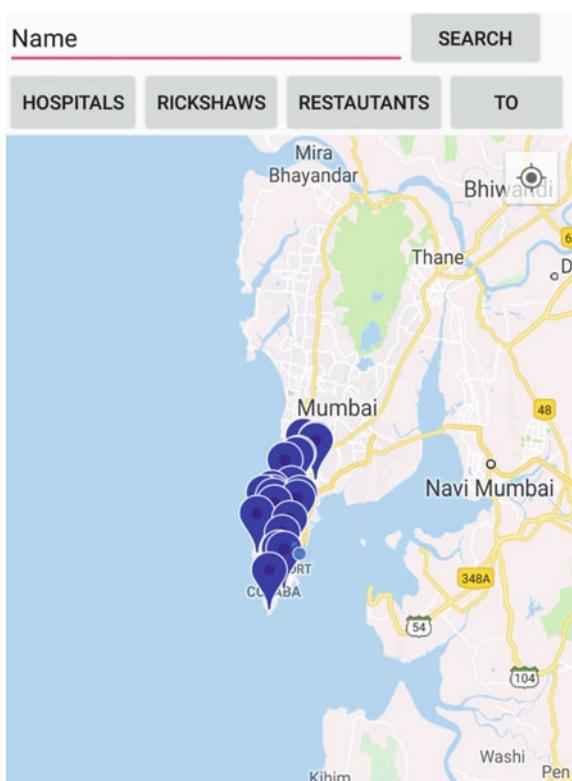


Fig. 8 Location of the nearby rickshaw stands



may decrease due to the cheaper routes offered by the system. This system will be more beneficial to relatively new workers who migrated to an urban city in search of jobs. Increase in the number of users will lead to more accurate recommendations from the systems since it can compare various profiles and recommended similar jobs to workers having similar profiles but no background history in the system. The job providers will find it easy to get the workers with the best possible skillset. A feedback feature can also be implemented further like the one in Uber, Ola, where the driver is given feedback. The worker's rating will be used for recommending jobs, and users will be provided with highly rated workers. The deep learning model suggested by the paper will help the system to evolve and incorporate new features into the system.

References

1. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput* 7(1):76–80
2. Restricted Boltzmann Machines. https://en.wikipedia.org/wiki/Restricted_Boltzmann_Machines
3. Recommender Systems. <http://recommendersystems.org/collaborativefiltering>
4. Recommender Systems. <http://recommender-systems.org/content-basefiltering>
5. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable and high-performance recommender systems. <https://www.researchgate.net>
6. Matrix Factorization Techniques for Recommender Systems. <https://towardsdatascience.com/paper-summary-matrix-factorization-techniques-for-recommender-systems-82d1a7ace74/>
7. Geo Google Maps working model. <https://geoawesomeness.com/the-famous-algorithmthat-made-navigation-in-google-maps-a-reality>
8. Restricted Boltzmann Machines. <http://deeplearning.net/tutorial/rbm.html>
9. Restricted Boltzmann Machine. <http://deeplearning.net/tutorial/rbm.html>
10. Gao T, Li X, Tang Y, Chai Y (2016) Deep learning with consumer preferences for recommender system international conference on information and automation, Ningbo China, vol 5 no 1, pp 11–20, August. 2016
11. Liu J, Dolan P, Pedersen ER (2010) Personalized news recommendation based on click behavior. In: Proceedings of ACM 15th international conference on intelligent user interfaces, Hong Kong, pp 31–40
12. Learning to rank. https://en.wikipedia.org/wiki/Learning_to_rank
13. Covington P, Adams J, Sargin E (2016) Deep neural networks for Youtube recommendations. In: Proceedings of the 10th ACM conference on recommender systems, pp 181–184
14. Kardan AA, Ebrahimi M (2013) A novel approach to hybrid recommendation systems based on association rules mining for content-based recommendation in asynchronous discussion group. *Inf Sci—Inf Comput Sci, Intell Syst, Appl, Inf Sci* 219:93–110
15. Autoencoders. <https://en.wikipedia.org/wiki/Autoencoder/>
16. Google Maps. <https://mathsection.com/how-google-maps-calculates-the-shortest-route/?cookie-state-change=1551188613952/>
17. Location and Maps. <https://developer.android.com/guide/topics/location>

Chapter 14

Implementation of Residual Network (ResNet) for Devanagari Handwritten Character Recognition



**Mandar Mhapsekar, Prathamesh Mhapsekar, Aniket Mhatre
and Vinaya Sawant**

1 Introduction

The study in the field of optical character recognition can be traced back to mid-1940s and has ever since gaining the attention of various industries and sectors. Optical character recognition has been highly used in banks, post offices, libraries, and publishing houses. The main challenge in OCR is handwritten character recognition. The research on handwritten character recognition began in the late 1960s, and at that time, only the handwritten numeric characters were addressed by the system [1]. Over the years, the technological approach for solving this problem has developed, thus improving the accuracy of the system [2]. Handwritten OCR for the English language which comes under the Latin script has almost developed into a full-fledged system. The research on handwritten OCR for Devanagari script is very limited compared to Latin script. Devanagari script includes languages like Hindi, Marathi, and Nepali. The most widely used Devanagari script language is Hindi with over 500 million people using this language [3].

In Devanagari script, there are 13 vowels and 36 consonants as shown in Fig. 1. There are 14 modifiers in Devanagari, out of which 11 are of vowels and 3 are of

M. Mhapsekar · P. Mhapsekar · A. Mhatre (✉) · V. Sawant

Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Vile Parle, Mumbai, India

e-mail: mhatreaniket121@gmail.com

M. Mhapsekar

e-mail: mhapsekarmandar@live.com

P. Mhapsekar

e-mail: prathmesh1297@gmail.com

V. Sawant

e-mail: vinaya.sawant@djsce.ac.in

क ख ग घ डं च छ ज झ ब ट ठ ड ढ ण त थ द ध न प फ ब भ म य र ल व श ष स ह क्ष त्र ज्ञ	अ आ इ ई उ ऊ ऋ ए ऐ ओ औ अं अः Vowels ऊ ऊ ऊ ऊ ऊ ऊ ऊ ऊ
Consonents	Modifiers

Fig. 1 Devanagari characters

'rakars.' These modifiers are combined with the consonants to form a modified character (the character with a modifier). Apart from vowels, consonants, and modified characters, we also have compound characters. Compound characters are formed by combining two or more simple characters. The compound characters are more complex in structure than the simple characters. There are 10 digits in Devanagari script. Devanagari is written from left to right, and there is no concept of uppercase/lowercase. Figure 2 shows a word in Devanagari script. Every character in Devanagari consists of a line above it which is called as the 'Shirorekha.' A character in Devanagari is divided into four sections, namely the top section, main section, side section, and the bottom section. The top section is the above the Shirorekha which consist of some modifier which is known as the upper modifier; the main section consists of the main character; the side section and the bottom section also



Fig. 2 Devanagari word

consist of some modifier which is known as the side modifier and the lower modifier, respectively [2].

The use of multilayer perceptron network is considered as a milestone in the field of handwritten character recognition but it needs a good feature extractor to extract relevant features, in which the multilayer perceptron can work to classify the character [4]. A better approach to this is to use a deep neural network. Convolutional neural network (CNN) is one of the classes of deep neural network. It does not require a feature extractor. It has an inbuilt feature extractor that works directly on the image and extracts the best feature from it for the classification [4]. In CNN, the classification accuracy increases as we increase the number of layers in the network; but at one point, above which if we increase the number of layers, the accuracy will start to saturate and eventually degrade. This is caused due to the vanishing gradient problem so it seems like the shallower network performs better than the deeper network. This problem is called the degradation problem [5]. Residual network (ResNet) was introduced to solve this problem. In ResNet, we have shortcut connections. Shortcut connections are those connections that skip one or more layers. The shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers [6].

In this paper, we have used residual network for Devanagari handwritten character recognition and showed through experiment how much the accuracy of the classification increases by using ResNet compared to the current state-of-the-art method.

The paper is organized into four sections. Section 1 gives an introduction to the paper. Section 2 deals with the work related to Devanagari handwritten character recognition. Section 3 contains the details about residual network which is the proposed approach in the paper. Experiments and results comparing the residual network with the current state-of-the-art method are shown in Sect. 4.

2 Related Work

In this section, we have summarized various techniques and methods used in Devanagari handwritten character recognition over the years which have given good results and performances.

2.1 Support Vector Machine (SVM)

Support vector machine is a supervised machine learning classification algorithm which when provided with labeled training data outputs a hyperplane that categorizes the new data into different classes. In two-dimensional space, this hyperplane is defined as a line dividing a plane into two parts, wherein each class lays on either side [7]. SVM requires an explicit feature extractor which extracts the features from

an image and produces a feature vector which is used by the SVM classifier. The paper [8] has used SVM as a classifier with Zernike moment as a feature extractor. It achieved an accuracy of 98.37% using their own database consisting of 9600 characters.

2.2 Artificial Neural Network (ANN)

Artificial neural network is a collection of nodes called as artificial neurons which resembles the neurons in the human brain. These neurons are connected to each other. ANN consists of three layers: the input, intermediate hidden layer, and the output layer. Each connection in the network has a weight associated with it. These connection weights are updated until the network is able to perform the task for which it is trained using a method called backpropagation [9]. ANN requires an explicit feature extractor like HOG, Zernike moment.

In HOG feature extractor, the distribution of directions of the gradient is used as a feature. The gradient is useful because the gradient value is high at the edges and corners. In HOG, the image is divided into cells where in each cell we calculate the magnitude and direction of gradients. Histogram of each cell is calculated based on the magnitude and direction of the gradient. A specific group of cells is combined into blocks in which normalization is performed. After normalization, values in the block are combined to form a single feature vector [10]. The features extracted by the extractor are applied to any classifier like ANN and SVM for the classification. The paper [10] has implemented HOG as a feature extractor along with ANN as a classifier. It achieved an accuracy of 82.66% using the ISI handwritten character database with input image of size 32×32 .

2.3 Bidirectional Long Short-Term Memory (BLSTM)-Based Recognition

BLSTM is used in RNN. Unlike ANN and CNN, RNN is not a feedforward neural network in which the data flow in one direction from input to output one layer at a time. In RNN, the output of the layer is added to the next input and fed back to the same layer. In LSTM, the node in RNN is replaced by an LSTM cell which has the ability to remember or forget previous contexts by using several gates. Similarly, BLSTM is bidirectional LSTM in which the learning sequence is in forward as well as backward direction [11]. BLSTM is a classifier so it needs a feature extractor to provide feature vector as input to it. BLSTM can be used with CNN as well as HOG-based feature descriptor which is proposed in the paper [10]. It has achieved accuracy of 94.56% and 79.54 with CNN and HOG as a feature extractor, respectively. It used ISI handwritten character database with input image of size 32×32 .

2.4 Convolutional Neural Network (CNN)

Convolutional neural network is a class of deep neural network, which takes an input image, assigns importance to various aspects in the image and be able to differentiate one from the other. CNN does not require an explicit feature extractor. In CNN, the first operation performed is the convolutional operation which is performed by the convolutional layer. In this layer, it has different feature detector/filter/kernel to detect features from the image by performing the convolutional operation. Thus, the output of this layer is different feature maps for each feature detector. Then it uses an activation function like ReLU to maintain the nonlinearity in the image. Next step is max pooling to make the model flexible enough to find the feature from the image even in an improper condition of the image. Then it performs flattening to make the feature in single vector form. At the last, it creates a full connection layer (ANN) for the classification of the image [12]. The paper [4] has proposed the use of CNN for Devanagari handwritten character recognition. It achieved an accuracy of 98.47% using the Devanagari handwritten character dataset with input image of size 32×32 .

3 Proposed Approach

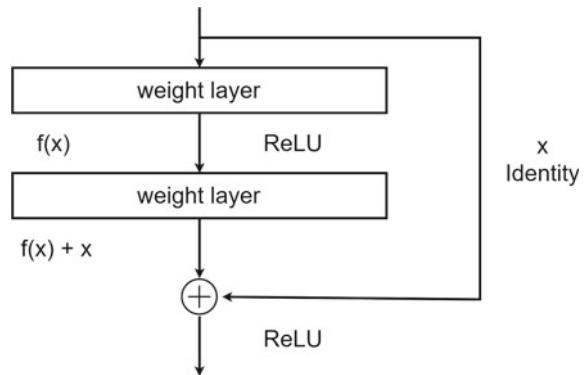
In this section, we have given details of residual network which is the method we have used for Devanagari handwritten character recognition.

3.1 Residual Network (ResNet)

Over the years, deep convolutional neural networks have made a series of breakthroughs in the field of image recognition and classification. Networks are going deeper to solve more complex tasks but due to the problem like vanishing gradient, if we have a sufficiently deeper network, it may not be able to learn even the simpler problems. If we keep increasing the layers of a model, at one point the accuracy will start to saturate and eventually degrade. This is called as the degradation problem [5].

He et al. [13] first demonstrated the depth problem and proposed a remarkable solution which has since allowed the training of over 2000 layers with increasing accuracy. Residual network consists of residual blocks. Figure 3 shows the structure of the residual block. In Fig. 3, we can see some layers using skip connection. Consider a neural network, which has x as input and approximates $H(x)$. Let us denote the difference between these as $R(x)$ whose equation is given below

$$R(x) = H(x) - x \quad (3.1.1)$$

Fig. 3 ResNet block

$R(x)$ is a residual function. If one hypothesizes that multiple nonlinear layers can asymptotically approximate complicated functions, then it is equivalent to hypothesize that they can asymptotically approximate the residual functions. We can see that the layers in the residual block are trying to learn the residual function $R(x)$. From this, we get an equation.

$$F(x) = H(x) - x \quad (3.1.2)$$

So the original function becomes $F(x) + x$ which are evident in Fig. 3. Because of these skip connections, we can propagate larger gradients to initial layers and these layers also could learn as fast as the final layers, giving us the ability to train deeper networks, solving the problem of vanishing gradient [6, 9].

4 Experiments and Results

In this section, we compared the current state-of-the-art method for Devanagari handwritten character recognition which is the convolutional neural network with the proposed method of residual network. We have obtained results corresponding to the various architectures of ResNet and CNN which are described in detail below.

4.1 Dataset

The dataset used for the experiment is Devanagari handwritten character dataset (DHCD) which is the work of Acharya [4]. The dataset contains 92,000 images of handwritten Devanagari characters. The dataset comprises 46 classes out of which 36 are alphabets and 10 are numbers. There are total of 2000 sample images of each character. Each character image is of size 32 by 32 pixels where 28 by 28 pixels is

character body which is padded by 2 pixels on all four sides. The dataset is divided into 80% train images, i.e., 73,600 images and 20% test images, i.e., 18,400 images.

4.2 Convolutional Neural Network (CNN)

For CNN, we have experimented two architectures having a depth of four layers and eight layers whose details are given below:

CNN with Four Layers

The CNN architecture consists of four layers consisting of two convolutional layers, two fully connected layers. The architecture is shown in Fig. 4. The input to convolutional layer is a 32×32 grayscale image. The convolutional layer uses a 5×5 overlapping kernel producing 16 feature maps of size 28×28 . The activation function used in this layer is ‘ReLU.’ Each feature map has a different set of weights. All the units in a feature map share the same set of weights and so they are activated by the same features at different locations. The convolutional layer is followed by the subsampling layer. Subsampling layer reduces the resolution of the feature map from convolutional layer by max pooling the features covered by a 2×2 filter. This step is important because the position of the feature may vary from image to image; therefore, the model must learn the relative position of the feature instead of the absolute position. The feature map produced by the subsampling layer is of size 14×14 which is then applied to the second convolutional layer. The second convolutional layer also uses a 5×5 kernel which gives 32 feature maps of size 10×10 which is then applied to the second subsampling layer which is the same as the first one. The output of the second sampling layer consists of feature maps of size 5×5 which are flattened to 800 neurons and applied to the 1000-way fully connected layer using ‘ReLU’ activation function with a dropout of 0.4. The last output layer consists of 46 nodes which represent the 46 output classes. It uses the ‘Softmax’ activation function. The fully connected layer is the traditional feedforward network. The model is trained using ‘Adadelta’ optimizer with default learning rate.

CNN with Eight Layers

We have extended the above model up to eight layers consisting of five convolutional layers and three fully connected layers to increase the depth of the CNN architecture.

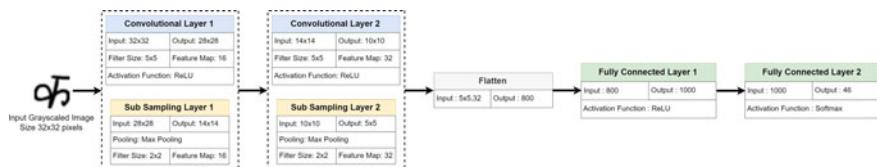


Fig. 4 4-layer CNN architecture

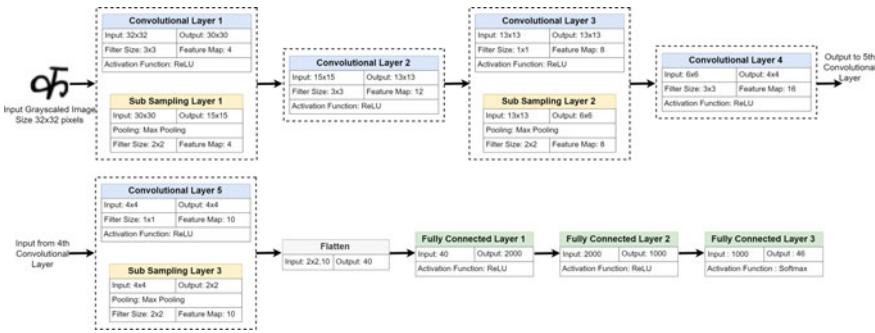


Fig. 5 8-layer CNN architecture

The architecture is shown in Fig. 5. The input to the first convolutional layer is a 32×32 grayscale image. This first layer uses a 3×3 overlapping kernel that outputs 4 feature maps each of size 30×30 . The activation function used in this layer is ‘ReLU.’ The convolutional layer is followed by the subsampling layer of filter size 2×2 that reduces the resolution of the feature map from convolutional layer by max pooling the features. The subsampling layer outputs feature maps of size 15×15 which are applied to the second convolutional layer. The second convolutional layer also uses the 3×3 kernel which gives 12 feature maps of size 13×13 .

The output of the second convolutional layer is directly applied as an input to the third convolutional layer having a kernel of size 1×1 where the number of feature map is 8 of size equal to the input, i.e., 13×13 . Sometimes referred as one-by-one convolutional layer or network in network, this layer increases the depth of the architecture to generate deeper network without simply stacking layers. The second subsampling layer which is similar to the first one takes input from the third convolutional layer and produces feature maps of size 6×6 . The output of the second sampling layer is passed to the fourth convolutional layer that uses a 3×3 kernel and outputs 16 feature maps each of size 5×5 . Similar to the third convolutional layer, the fifth convolutional layer uses a kernel of size 1×1 and has 10 feature maps. The output of the fifth convolutional layer is applied to the third subsampling layer with a filter size of 2×2 which is flattened to 40 neurons. The flattened neurons are applied to the first 2000-way fully connected layer using ‘ReLU’ activation function with a dropout of 0.5. The second 1000-way fully connected layer also uses ‘ReLU’ activation function with a dropout of 0.5. The last output layer consists of 46 nodes which represent the 46 output classes. It uses the ‘Softmax’ activation function. The model is trained using ‘Adadelta’ optimizer with default learning rate.

The final accuracy of the model consisting of four layers obtained after training for 5 epochs is 98.79%, and for the model consisting of eight layers, we obtained a final accuracy of 96.42%. Figure 6 shows the results. So increasing the layers of the CNN model leads to a decline in accuracy which is caused due to the problem of vanishing gradient.

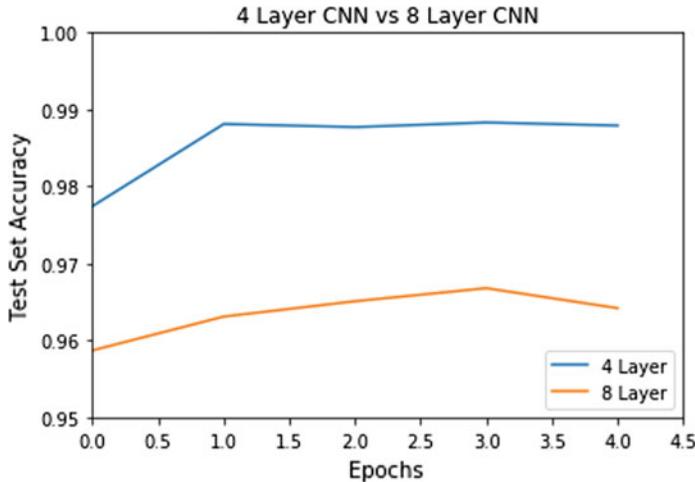


Fig. 6 4-layer CNN versus 8-layer CNN

4.3 *Residual Network (ResNet)*

For ResNet, we have experimented two architectures, namely ResNet 34 and ResNet 50, which consist of 34 and 50 layers, respectively. The architecture detail is given below:

ResNet 34

ResNet 34 has a depth of 34 layers. The architecture is described in Table 1. Like every ResNet, ResNet 34 also consists of a common convolutional layer and a pooling step which are followed by four convolutional layer groups having similar behavior. Each convolutional layer group uses a kernel of size 3×3 and has fixed number of feature maps which are 64, 128, 256, 512, respectively. The first group consists of 3 pairs of convolution. The second group consists of 4 pairs of convolution. The third group consists of 6 pairs of convolution, and the final group consists of 3 pairs of convolution. At the last, we have a 1000-way fully connected layer using ‘Softmax’ activation function before which we have an average pooling layer. The first convolution of each group uses a stride of 2 because of which the size of the feature map reduces by half. Various other parameters are set to default parameters of standard ResNet 34 [6].

ResNet 50

ResNet 50 has a depth of 50 layers. The architecture is described in Table 1. Like every ResNet, ResNet 34 also consists of a common convolutional layer and a pooling step which are followed by four convolutional layer groups having similar behavior. Each convolutional layer group consists of some triples having kernel size as 1.3 and 1, respectively. The first group consists of 3 triples of convolution. The second

Table 1 ResNet 34 and ResNet 50 architecture

Layer name	Output size	34-layer ResNet	50-layer ResNet
Convolutional layer 1	112×112	$7 \times 7, 64$, stride 2	
Convolutional layer 2	56×56	3 \times 3, max pool, stride 2 $\left[\begin{array}{l} 3 \times 3, 64 \\ 3 \times 3, 64 \end{array} \right] \times 2$	$\left[\begin{array}{l} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 2$
Convolutional layer 3	28×28	$\left[\begin{array}{l} 3 \times 3, 128 \\ 3 \times 3, 128 \end{array} \right] \times 2$	$\left[\begin{array}{l} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 2$
Convolutional layer 4	14×14	$\left[\begin{array}{l} 3 \times 3, 256 \\ 3 \times 3, 256 \end{array} \right] \times 2$	$\left[\begin{array}{l} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 2$
Convolutional layer 5	7×7	$\left[\begin{array}{l} 3 \times 3, 512 \\ 3 \times 3, 512 \end{array} \right] \times 2$	$\left[\begin{array}{l} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{array} \right] \times 2$
	1×1	Average pool, 1000-d fully connected Activation function: Softmax	

*Here convolutional layers are represented as *filter size, number of feature maps, stride (optional)*

group consists of 4 triples of convolution. The third group consists of 6 triples of convolution, and the final group consists of 3 triples of convolution. Number of feature maps in each convolution is shown in Table 1. The average pooling and the fully connected layer are the same as ResNet 34. Various other parameters are set to default parameters of standard ResNet 50 [6].

The final accuracy for ResNet 34 obtained after training for 5 epochs is 98.73%, and for ResNet, we obtained a final accuracy of 99.35%. Figure 7 shows the results. So increasing the layers of the ResNet model increases the accuracy, thus solving the problem of vanishing gradient.

4.4 Result Summary

Table 2 gives the summary of experiments performed on different network architectures.

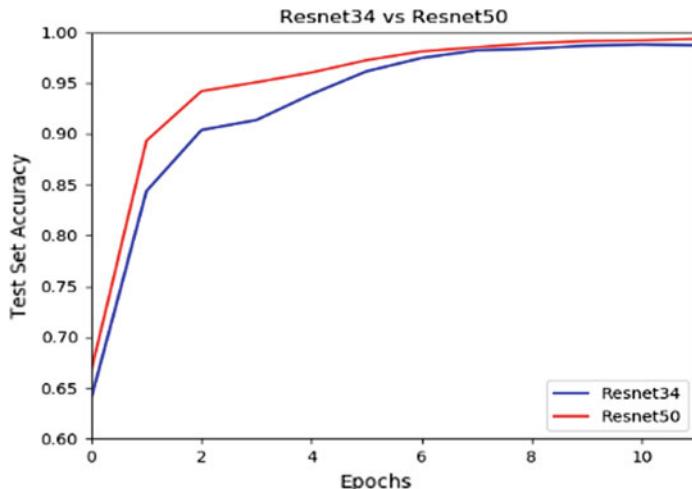


Fig. 7 ResNet34 versus ResNet50

Table 2 Experimental summary

Architecture	4-layer CNN	8-layer CNN	ResNet 34	ResNet 50
Accuracy (in %)	98.79	96.42	98.73	99.35

5 Conclusion

The comprehensive study of residual network for Devanagari handwritten character recognition and its comparison with the current state-of-the-art architecture of convolutional neural network (CNN) is the first of its kind. The proposed implementation using ResNet architecture obtained the highest accuracy of 99.35% which is significantly higher than the current state-of-the-art architecture of CNN. The highest result was obtained using ResNet 50 architecture. The proposed approach of using residual network can be the beginning of the use of the deeper network in Devanagari handwritten character recognition problem.

References

1. IBM. https://www.ibm.com/ibm/history/exhibits/rochester/rochester_chronology2.html
2. Jayadevan R, Kolhe SR, Patil PM, Pal U (2011) Offline recognition of Devanagari script: a survey. IEEE Trans Syst, Man, Cybern Part C (Appl Rev) 41(6):782–796
3. Wikipedia List of Languages by native speakers in India. https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India
4. Acharya S, Pant AK, Gyawali PK (2015) Deep learning based large scale handwritten Devanagari character recognition. In: 2015 9th international conference on software, knowledge,

- information management and applications (SKIMA), Kathmandu, pp 1–6
- 5. Residual blocks—Building blocks of ResNet. <https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec>
 - 6. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, pp 770–778
 - 7. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20:273. <https://doi.org/10.1023/A:1022627411411>
 - 8. Kale KV, Deshmukh PD, Chavan SV, Kazi MM, Rode YS (2013) Zernike moment feature extraction for handwritten Devanagari compound character recognition. In: 2013 science and information conference, London, pp 459–466
 - 9. Understanding Residual Networks. <https://towardsdatascience.com/understanding-residual-networks-9add4b664b03>
 - 10. Chakraborty B, Shaw B, Aich J, Bhattacharya U, Parui SK (2018) Does deeper network lead to better accuracy: a case study on handwritten Devanagari characters. In: 2018 13th IAPR international workshop on document analysis systems (DAS), Vienna, pp 411–416
 - 11. Salehinejad H, Baarbe J, Sankar S, Barfett J, Colak E, Valaee S (2017) Recent advances in recurrent neural networks. arXiv preprint [arXiv:1801.01078](https://arxiv.org/abs/1801.01078)
 - 12. Le Cun Y, Boser B, Denker JS, Howard RE, Hubbard W, Jackel LD, Henderson D (1990) Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems, pp 396–404
 - 13. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. [arXiv:1512.03385](https://arxiv.org/abs/1512.03385)

Chapter 15

An Efficient E-Commerce Design by Implementing a Novel Data Mapper for Polyglot Persistence



Kishan Trivedi, Sambhav Shah and Kriti Srivastava

1 Introduction

Polyglot persistence is the concept of using different database systems within a single application domain, addressing different functional and non-functional needs with each system. In the recent years, we have had to encounter a sort of data invasion. Data coming from various sources, of various types, need to be stored in a single system. In traditional Web applications, relational databases formed the data storage backbone. E-commerce applications tend to receive massive spikes during peak seasons, and increasing the capacity in an elastic manner to store data in a relational structure is a challenging task. Also, achieving required performances in terms of latency and availability becomes cumbersome. Polystores follow the ideal of one size does not fit all [1] and help in tackling the aforementioned problem of e-commerce applications. A relational model is advantageous in providing ACID guarantees, and therefore, relational models prove to be effective where consistency is pivotal. Shared data systems provide certain guarantees such as consistency, availability and partition tolerance (CAP). According to Brewer, it is impossible for a distributed system to provide all three guarantees simultaneously [2]. Based on these principles laid out by Brewer, various NoSQL systems emerged that gave up either consistency or availability and embraced partition tolerance. There are some modules in an e-commerce business model such as orders, payments and inventory that have stringent

K. Trivedi (✉) · S. Shah · K. Srivastava

Department of Computer Engineering, Dwarkadas Jivanlal Sanghvi College of Engineering,
Mumbai, India

e-mail: trivedi.kishan97@gmail.com

S. Shah

e-mail: sambhavshah26@gmail.com

K. Srivastava

e-mail: kriti.srivastava@djsce.ac.in

atomicity, consistency and durability requirements. These requirements are met by relational systems providing ACID guarantees. In order to achieve high performance and quick response times along with consistency in transactions in any e-commerce business model, we should ideally have a combination of SQL and NoSQL databases in the data tier. Our system integrates three different types of databases to form the overall back end of an e-commerce application. To this end, we have created a data mapper that can direct incoming data to its respective database.

2 Related Work

The concept of using multiple programming languages in the development of an application is known as polyglot programming [3]. Similarly, the concept of using multiple databases in one application system is known as polyglot persistence [4]. Modern applications should be designed in such a way that it can store and manipulate data in multiple data stores repeatedly, in real time. This is a difficult task as it increases the complexity of handling queries. Each data store has its own query language and data model, thus further increasing the complexity of the system. To solve these problems, researchers have proposed various solutions to provide easy access to data stores. Some of them are based on making a common API, while others are based on development of frameworks to access different data stores. There exists no single solution in the form of a database that can fulfill the growing needs of big data. No single database, SQL or NoSQL, satisfies the need of big data. Thus, polyglot persistence systems have come into existence. Even small-scaled applications are using different databases to store data. PolyHIS [5] is a framework that implements different types of data models to store healthcare data. The concept of polyglot persistence can also be used to improve the Energy Data Management System (EDMS) [6]. There are many other studies on this concept. Reference [7] gives a hybrid data store architecture that uses a data mapper to map application data to its respective data store. A major issue in polyglot persistent systems is how to implement queries across varied data stores that have different query languages and different data models. There are some federated query languages (example: CloudMdsQL) that help in querying heterogeneous data stores [8]. There are mainly three types of approaches of polyglot persistence: (1) domain specific (2) query language specific (3) framework specific [9]. In the domain-specific approach, the most relevant features are extracted from the system with the help of domain experts. This approach decides the types of data stores based on the extracted features and then uses them accordingly. In the query language-specific approach, there are two possible ways to implement polyglot persistence. One of them is to develop a universal query language, and the second one is to use a converter for queries across different databases. In the framework-specific approach, the idea is to build a multi-model system with the features of identifying various forms of data (XML, JSON and graphs). In the domain-specific approach, the selection of databases is based on the function of CAP characteristics [2]. The selection of NoSQL databases in the polyglot persistent system is generally done

on the basis of the application data. The main types of NoSQL databases are: (1) key-value stores (2) columnar stores (3) document stores (4) graph stores [10]. In key-value stores, the data is stored in the form of key-value pairs, using a hash table. The columnar stores handle the problem of null value columns in SQL. They give rows the flexibility to define their own columns. Document stores are one of the systems that provide highly flexible schemas. They are schema less and can handle complex document structures. Graph stores store data in the form of relationship nodes. They help in finding patterns in data. One of the main challenges in polyglot persistence is analyzing the schema of data and deciding in which database should we store and fire queries on the data. We have designed a data mapper that does this work with the help of some human intervention and a polyglot persistent model.

3 The Need for Polyglot Persistence

The use of polyglot persistence is oriented toward flexibility. This flexibility can be achieved using NoSQL databases. These databases are based on the set theory and generally have less consistency. On the other hand, they provide a flexible schema, scalability and fault tolerance. ACID-based SQL databases provide good consistency and are suitable for secure transactions (financial transactions). A user can perform various tasks in an e-commerce application such as searching for products in the product catalog, adding a product to a cart, buying a product, canceling/returning an order and registering on the application. Various databases are suitable for various functionalities. A generic e-commerce application includes the following modules
(1) Product Catalog: This module consists of details of all the products in the online store. The various attributes of products include brand, rating and cost. A document store is ideal for storing such type of data. MongoDB is a good option to store data of this module. Horizontal scalability and availability provided by MongoDB allow for seamless storage of seemingly inexorable data like that in product catalog. (2) Product Recommendation: Here, we can make use of a graph database like Neo4j that can highlight similarities between products and make recommendations accordingly to the customers. (3) Shopping Cart: In this module, customers can add products to a cart and later on decide whether to buy all the products or not. (4) Orders: Here, a registered customer can buy the products shortlisted in the cart. (5) Checkout: The customers pay for their order with multiple payment options available. Modules (3), (4) and (5) have stringent consistency requirements, and a relational database must be chosen for such tasks. Another kind of database that comes in handy for such use cases is a key-value database. Redis, for example, is a key-value store. It has shown great performance on data which requires constant updation or modification. In an e-commerce application, Redis can be used to store product search counter. The value of the respective key can be updated every time the product is searched. This indicates the number of times a product has been searched and hence its popularity. Therefore, using polystores in e-commerce is much more efficient than using a single type of database.

4 Design of the Intelligent Data Mapper

An architectural description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. Various models have been proposed in the design of a polyglot persistent system for an e-commerce application. They are mainly based on the primary architecture provided by Mr. Martin Fowler [11]. One of the most crucial steps in the design of this model is the selection of databases. Different databases are optimal at handling different types of data. Also, mapping of data from these dissimilar databases is only possible if they meet the compatibility criteria. An e-commerce application generally holds data such as product details, customer details, payment details and search hits. MongoDb can handle document-type data efficiently. It increases the scalability of the system. Building an e-commerce application with a MongoDb document store has many advantages over using the traditional RDBMS [12]. In an e-commerce application, MongoDb can be used to store product details, customer details and other document-type data. Redis performs better on data which requires constant updation or modification [13]. In an e-commerce application, Redis can be used to store product search counter. The value of the respective key can be updated every time the product is searched. Relational databases perform better when there is a need for aggregate queries [14]. Relational databases can be used in an e-commerce application to store payment details. This can be helpful for business analytics. Keeping the above points in mind, we can choose the respective databases. The next step must be the development of a data mapper that maps data to the appropriate database. Our architectural description depicts how the flow works, inputting the query, decomposition, data mapping and displaying the updated data in the front end of respective database technologies used. It also labels the name of mappers and the database technologies used by us. Here, the input point of the data/query is the data facade (DF). Then, we use object-relational mappers that act as data mappers (DM) and data drivers (DM) to our database. Finally, we can interact with the databases through the respective front end (FE) or a unified front end. We integrate this system as a whole using Python support for the ORMs, and this leads to the formation of a unified data mapper. We use a three way bidirectional architecture for our system. The combination of three ORMs acts as our unified data mapper (Figs. 1 and 2).

5 Results and Analysis

Firstly, we selected an open-source e-commerce data set having multiple tables such as product details, customer details, payment details, search hits. Then, we implemented a set of 11 queries in total on three different systems. These queries mainly consisted of CRUD operations along with some complex integrated queries. In the first system, the data was entirely stored on the MongoDb database. In the second system, the data was entirely stored on a relational database (Sqlite), and in the third

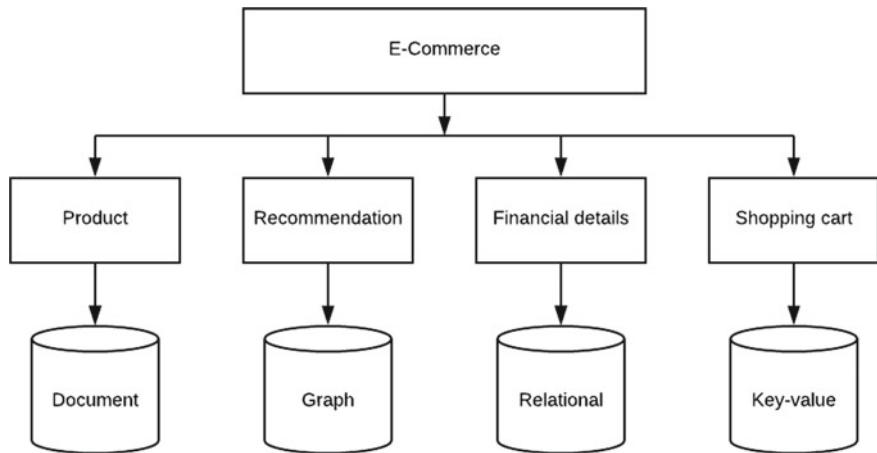


Fig. 1 Polyglot model (e-commerce)

and the final system, the data was stored on our polyglot persistent system with the data mapper. The query execution time of each query in each system was compared. We also implemented this comparison for different data set sizes (number of tuples). The following were the obtained results (Fig. 3).

The results showed the following characteristics: Numerical queries having aggregation functions performed better on a relational database (SQLite). Text search queries performed better on a NoSQL database. This is because of the text indexes available in MongoDB that were used while querying. It is clear that the results of the polyglot system varied according to the data type of the fields. A juxtaposition of our polyglot system with SQLite and MongoDB systems shows that most optimum results were obtained from the polyglot system. This system basically tries to pick out the best results from individual databases and hence provides best query performance.

6 Conclusion and Future Scope

Using this project, we intend to highlight the need for polyglot persistence in order to achieve better scalability and availability and also guaranteeing consistency through transactional systems where consistency was required aside from the obvious advantage of better query performance. Here, we have used an e-commerce as an example and discussed the implementation of a scalable architecture that uses the concept of a polyglot data mapper. The decision of using our own data mapper was taken to ensure flexibility of implementation and meeting performance and extensibility requirements of present day Web applications. The main advantage of this architecture is that it is very simple, and cross-database queries are possible. If implemented

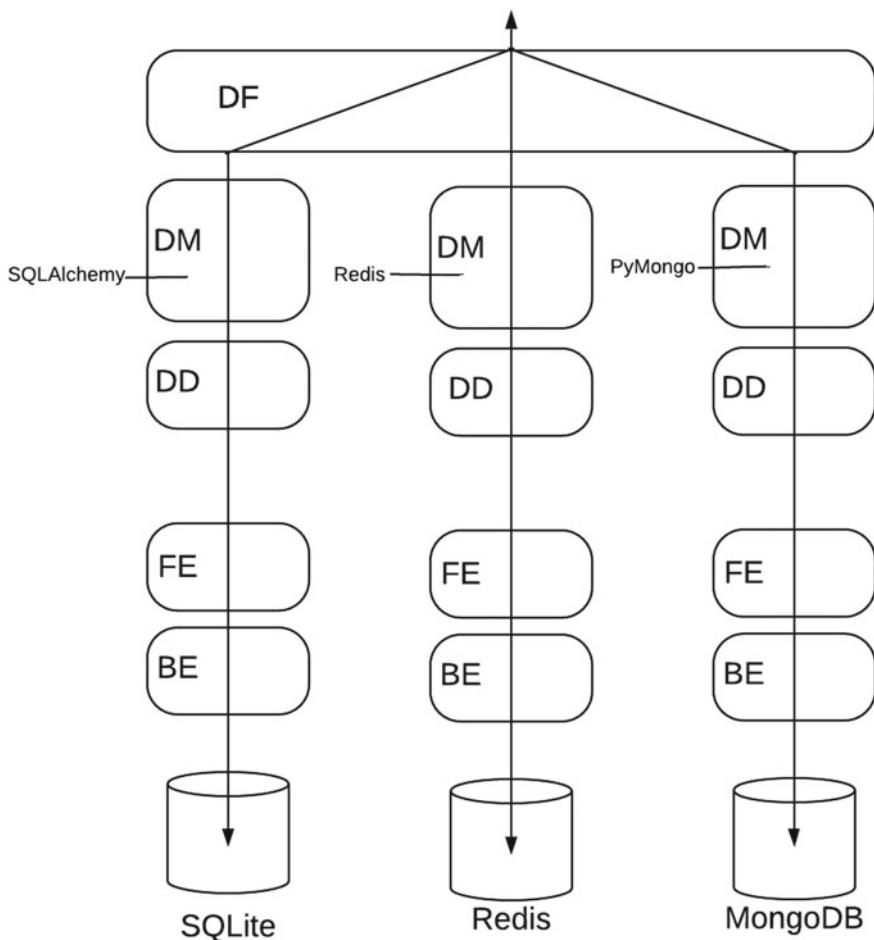
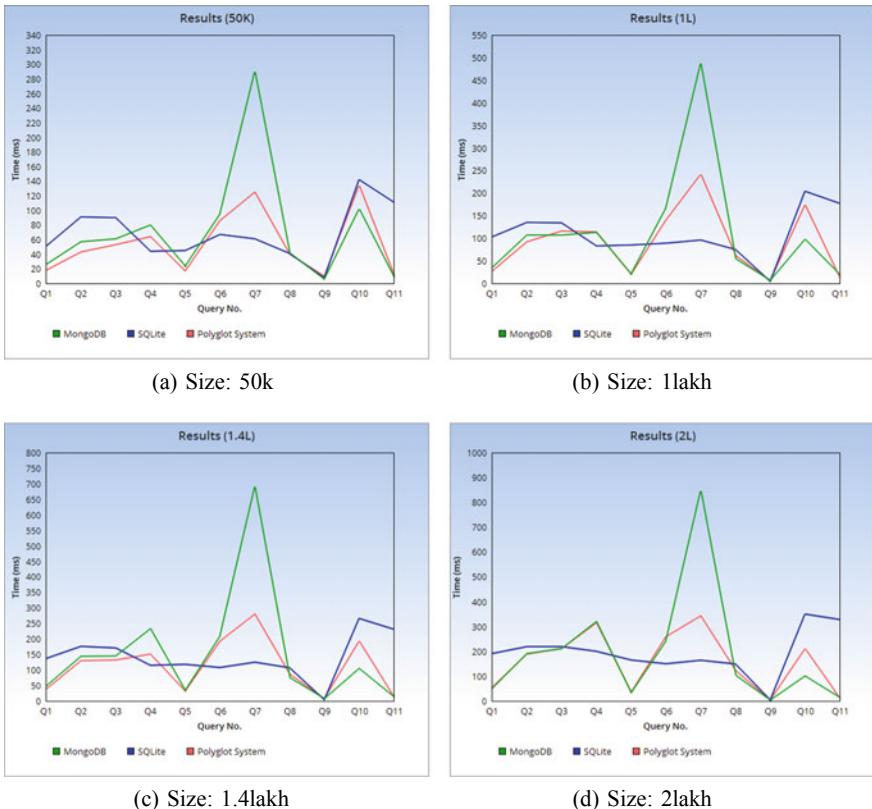


Fig. 2 Architecture

successfully, this idea can be taken on by numerous organizations in the industry that are currently faced by the system that use only traditional RDBMS. This will help them to cope up with the deluge of data that is fed into these systems. Also, further research and testing will result into discovery of new architecture models which can be further useful. If currently incompatible databases products start with providing the necessary support to polyglot system, then the whole scenario of data management will drastically change. Also, with this technology being easily available, the investment and maintenance expenses will also gradually reduce.

**Fig. 3** Results

References

1. Hachem N, Helland P, Stonebraker M, Madden S, Abadi DJ, Harizopoulos S (2007) The end of an architectural era: its time for a complete rewrite. In: Proceedings of the 33rd international conference on very large databases, Vienna, Austria, 23–27 Sept 2007
2. Brewer EA (2000) Towards robust distributed systems (abstract). In: Proceedings of the nineteenth annual ACM symposium on principles of distributed computing, Portland, OR, USA, 16–19 July 2000
3. PolyglotProgramming. Available online: <http://memeagora.blogspot.com/2006/12/polyglot-programming.html>
4. Polyglot Persistence. Available online: <https://martinfowler.com/bliki/PolyglotPersistence.html>
5. Kaur, K, Rani R (2015) Managing data in healthcare information systems: many models, one solution. Computer 48:52–59
6. Prasad S, Avinash SB (2014) Application of polyglot persistence to enhance performance of the energy data management systems. In: Proceedings of the 2014 international conference on advances in electronics, computers, and communications, ICAECC 2014, Bangalore, India, 10–11 Oct 2014

7. Chintan Shah C, Srivastava K, Shekokar N (2016) A novel polyglot data mapper for an E-Commerce business model. In: Proceedings of the 2016 IEEE conference on e-Learning, e-Management and e-Services, IC3e 2016, Langkawi, Malaysia, 10–12 Oct 2016
8. Kolev B, Valduriez P, Bondiombouy C, Jimnez-Peris R, Pau R, Pereira J (2015) CloudMdsQL: querying heterogeneous cloud data stores with a common language. *Distrib Parallel Databases* 34:463–503
9. Gessert F, Ritter N (2016) Scalable data management: NoSQL data stores in research and practice. In: Proceedings of the 2016 IEEE 32nd international conference on data engineering, ICDE 2016, Helsinki, Finland, 16–20 May 2016
10. Tudorica BG, Bucur C (2011) A comparison between several NoSQL databases with comments and notes. In: Proceedings of the RoEduNet IEEE international conference, Iasi, Romania, 23–25 June 2011
11. Sadalage PJ, Fowler M (2012) NoSQLdistilled: a brief guide to the emerging world of polyglot persistence. Pearson Education
12. Ramesh D, Khosla E, Bhukya SN (2016) Inclusion of e-commerce workflow with NoSQL DBMS: MongoDB document store. <https://doi.org/10.1109/IC-CIC.2016.7919652>
13. Paksula M (2010) Persisting Objects in Redis Key-Value Database, University of Helsinki, Department of Computer Science Helsinki, Finland
14. Aboutorabi SH, Rezapour M, Moradi M, Ghadiri N (2015) Performance evaluation of SQL and MongoDB databases for big e-commerce data. <https://doi.org/10.1109/csicsse.2015.7369245>

Chapter 16

Improving Extreme Learning Machine Algorithm Through Optimization Technique



Nilesh Rathod and Sunil B. Wankhade

1 Introduction

With the exponential growth of technical improvements, there is always some kind of basic data that gets produced at a rapid rate, and the parameters of this data such as the size and dimensionality are continually expanding. It, thus, becomes imperative that resourceful machine learning methods are developed which are further used to understand the data properly and discover beneficial knowledge that is required to create understandings from the wealth of information available. Extreme learning machines (ELMs) are a recently occurring widespread structure in the domain of machine learning. ELMs are essentially feed-forward neural networks characterized by an unplanned initialization of hidden layer weights and a fast training algorithm which are joined together. The efficiency of this algorithm is proven to have better results which in turn makes the algorithm very appealing for a larger dataset.

Contrary to the theoretical position on ELMs, practically, the number of samples matters greatly how many samples are available for training. Outliers also impact greatly on the results, and there is an emphasis on the variables used as inputs. Therefore, proper care needs to be taken to prevent overfitting and to obtain a robust and accurate model. Furthermore, because of the size of modern datasets, ELMs benefit from strategies of accelerating their training even though ELMs have efficient training algorithms.

We intend to propose an approach to enhance extreme learning machine. Here, we integrate a multi-objective optimization approach to improve classification accuracy. The optimal parameters include the feature parameters, cost which forms the input

N. Rathod (✉) · S. B. Wankhade

MCT's Rajiv Gandhi Institute of Technology, Andheri (W), Mumbai 053, India
e-mail: nilesh.rathod@mctrgit.ac.in

S. B. Wankhade

e-mail: sunil.wankhade@mctrgit.ac.in

to the ELM. Once the optimal parameters are selected, the random input problem can be avoided along with the improvement in prediction accuracy.

1.1 Background

Artificial neural networks (ANN) are groundbreaking numerical models that are referred to as all-inclusive approximates which demonstrate how any neural system with a solitary hidden layer can estimate any continuous function. This property made ANNs appropriate for the issues of characterization and regression. For example, in supervised learning issues like characterization, the computational model that we look for is the one that symbolizes the preparation set in the most ideal way with the goal that it can effectively order new concealed information. The objective is to fabricate a model that can anticipate the estimation of the objective variable from the input variables Kotsiantis et al. [1].

When utilizing ANN as a classifier, one should consider the quantity of concealed layers, the estimations of weights between the layers, and the choice of the learning algorithm. The execution of the classifier is very much influenced by the mix of the structure of the system and the learning algorithm Mohapatra et al. [2]. Single-hidden layer feed-forward neural networks (SLFN) are the most regularly utilized kind of ANNs. As the name infers, SLFN has just one hidden layer that interfaces the information layer to the yield layer. In the previous decades, feed-forward neural systems have been extensively utilized in numerous fields in light of its undeniable excellences. From one perspective, it could approximate complex nonlinear mappings directly from the information tests. Also, it can offer models which are tough for traditional parametric procedures to deal with. Nonetheless, there exists the reliance between various layers of parameters for which each of the parameters of the feed-forward network should be tuned, which render feed-forward neural systems tedious. Single-concealed layer feed-forward systems (SLFNs), a standout amongst the feed-forward neural systems, have been broadly considered from both hypothetical and application views for their capacities of learning as well as fault tolerance Razavi and Tolson [3]. Nevertheless, most well-known learning calculations for preparing SLFNs are still generally moderate since every one of the parameters of SLFNs should be tuned through iterative techniques, and these calculations may likewise effectively stall out in a local minimum. Generally, SLFNs are trained to utilize gradient descent techniques. For example, backpropagation algorithms to tune loads of the system. In spite of the existing calculations, they face real troubles which incorporate the high reliance on the underlying loads of the system, the likelihood of being caught in nearby minima, and a slow convergence Ding et al. [4], Kaya et al. [5].

Extreme learning machine (ELM) Huang et al. [6] are a quick-learning neural algorithm for SLFNs that was introduced to improve the ability of SLFNs. Unique

in relation to the conventional learning algorithms for neural systems, (e.g. BP calculations), which may confront challenges in physically tuning the control parameters (learning rate, learning ages, and so forth.) as well as local minima, ELM is completely automatic which is actualized without iterative tuning, and in principle, no intercession is required from clients. Besides, the speed of learning of ELM is incredibly quick contrasted with other customary strategies. The learning parameters in the ELM algorithm, of the nodes that are hidden, including input loads and biases, can be arbitrarily assigned, and the yield loads of the system can be systematically controlled by the basic summed up reverse activity. The stage of training can be productively finished through a fixed nonlinear change without a tedious learning process. Additionally, it has been proved that the standard ELM has the ability of universal approximation along with an additive or RBF activation function [7, 8]. Furthermore, the successful application of the ELM to various real-world applications of classification as well as regression problems can be observed Wang et al. [9], Lim et al. [10].

1.2 Comparative Analysis

Compared with other classical learning algorithms in the neural networks such as backpropagation, ELM can accomplish improved performance in a much quicker learning time. Extreme learning machine provides better performance in comparison with support vector machine (SVM) over regression and general classification problems. An ELM offers the advantages of low computational cost, good generalization ability, and ease of implementation that is not the case with backpropagation and support vector machine. However, backpropagation and support vector machine suffer from some problems like local minima, slow convergence rate, and intensive human intervention. These types of disadvantages can be covered in extreme learning machine.

2 Proposed Work

Extreme learning machine has been widely used for prediction based upon the history of data available in various fields. Marjory in the medical field for disease prediction based on the patients' data history has aided medical practitioners in discovering abnormalities in early stages; ELM-based prediction approach is carried out. ELM approach is invariably used for solving classification problems as it provides options for training any sort of datasets. Various approaches were formulated recently for prediction approach based upon the historical data available. But, majority of these approaches has issues in the final prediction made due to inappropriate feature selections and related to the data preprocessing. Extreme learning machine has certain drawbacks which can adversely affect its classification accuracy. Major problem

associated with ELM is that it requires a high number of hidden neurons and leads to ill-conditioned problems due to the random determination of the input weights and hidden biases. Majorly, ELM uses random input parameters which can induce irrelevant classification. However, the algorithm used often needs a large number of hidden units and therefore responds slowly to new observations. The non-analytical parameter determination procedure does improve efficiency of learning, but it also leads to fluctuating performance of ELM on the same problem with different initial parameters. Thus, the ELM algorithm may appear less stable.

Also, some of the challenges related with ELM are the computational cost; as ELM works on random inputs, more cost is required. Another challenge in regression process is singularity and overfitting if the training sample is less.

2.1 Design Issue

The following are some issues related with ELM,

1. It was observed that the classification boundary of the hidden layers' learning parameters may not be optimal since they remain the same during training.
2. ELM is not capable of managing large, high-dimensional data, since it needs more hidden nodes compared to the conventional tuning algorithms.
3. The optimization process may not necessarily find an optimal solution because of a deceptive local minima. Solutions can only be compared relatively and thus have a questionable solution quality.
4. The optimization process may take an extensive period of time to find a reasonable solution, due to the iterative nature of the search, and hence, there is a chance of having long training time.
5. The optimization process has a chance of failing to locate a viable solution or failing to progress by getting stuck due to the presence of flat regions.

3 Methodology

There is a need to improve the structure of extreme learning machine with the help of optimization algorithms so that the problem of a high number of the hidden neurons and random input can be solved. It helps in improving the classification accuracy in medical diagnosis applications.

To optimize the ELM structure, we are using invasive weed optimization (IWO) with a levy flight cuckoo search (CS) algorithm so that the classification accuracy is improved. Figure 1 illustrates the proposed methodology. Initially, the collected data is preprocessed and involves transforming raw data into an understandable format. Then, the features are extracted from this preprocessed data. Then, extreme learning machine is applied to classify the data. The ELM structure is optimized using

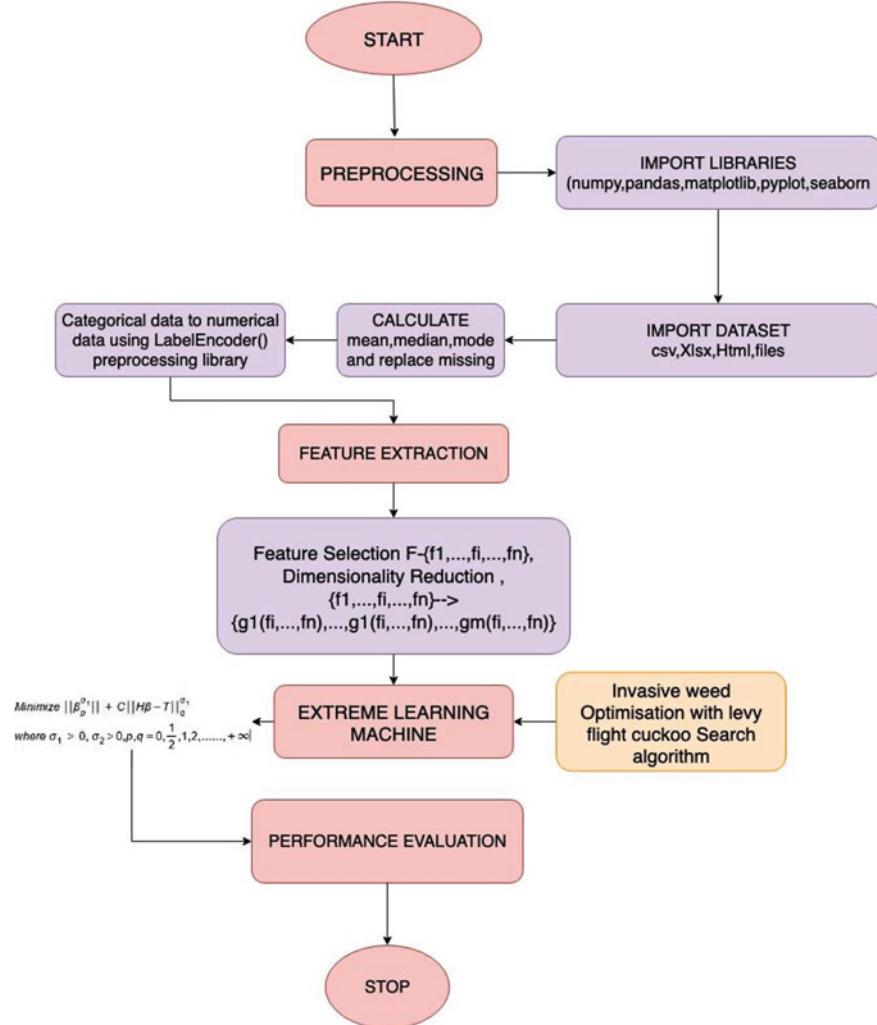


Fig. 1 Process flow of the proposed methodology

invasive weed optimization with a levy flight cuckoo search algorithm. Finally, the performance of the system is evaluated with respect to its accuracy. The multiple objectives considered for optimization so that the ELM structure is improved are the activation function, mean square error, number of nodes and running time.

The optimal parameters such as cost and feature parameters will be the input to the ELM. After the selection of the parameters, the random input problem will be avoided along with an enhancement in the prediction accuracy.

3.1 Proposed Methodology Steps

- Step 1. Initialize positions and head angles with a set of input weights and hidden biases: [11, 12, ..., 1, ..., 21, 22, ..., 2, ..., 1, 2, ..., 1, 2, ...]. These will be randomly initialized within the range of $[-1, 1]$ on D dimensions in the search space.
- Step 2. For each member in the group, the respective output final weights are computed by ELM.
- Step 3. Now, invoke refined IWO with levy flight cuckoo search. Mehrabian et al. [11], Mohapatra et al. [2].
- Step 4. Then, the fitness of each member is evaluated.
- Step 5. Find the producer of the group based on the fitness.
- Step 6. Update the position of each member.
- Step 7. Stopping criteria: Repeat Steps 2–6 until certain conditions are met, along with hard threshold value as the maximum number of iterations. The algorithm returns the optimal final weights with minimal MSE as its solution once it reaches the stopping criteria. Thus, considering both the advantages of both ELM and IWO with Cuckoo search, refined IWO with ELM finds the best optimal weights and bias so that the fitness reaches the minimum to achieve better generalization performance, with minimum number of hidden neurons. In the process of selecting the input weights, the IWO with cuckoo search considers not only the MSE on validation set but also the norm of the output weights. The proposed IWO-based ELM will combine the feature of IWO with cuckoo search into ELM to compute the optimal weights and bias to make the MSE minimal.

3.2 IWO-CS ELM Procedure:

- Step 1: Considering the size of the training data as \times , where refers to the number of samples, and refers to the dimension of features.
- Step 2: Feature extraction and matrix labelling from training data.
- Step 3: Input weight calculation using proposed multi-objective optimization like IWO-CS.
- Step 4: Calculate the hidden layer output based upon the input data used.
- Step 5: Calculate the final output.

The similar steps will be carried out for testing data for the final classification. As mentioned, we enhance the ELM using multi-objective optimization algorithm where IWO and CS will be integrated. IWO functionality will be combined using the levy flight CS functionality. Combining the algorithms refines the optimized output, which is the weight factor. This weight factor is now convoluted with the feature weight, which is the input to the ELM. By optimally selecting the weight function,

the classification accuracy will be improved, and this will be evaluated by comparing with ELM without optimization.

4 Discussion and Conclusion

This paper presents an approach based on using invasive weed optimization (IWO) with levy flight cuckoo search (CS) and extreme learning machines for training single-hidden layer feed-forward networks (SLFN). The IWO-CS-ELM model utilizes the IWO-CS algorithm to optimize the input weights and hidden biases and determines the output weights using MP generalized inverse. The optimal selection of the parameters will avoid the problem of random input and also improve the accuracy of prediction. The performance of IWO-CS-ELM was evaluated with the accuracy. In addition, IWO-CS is also used to utilize a regularized ELM, which shows more consistent performance in comparison with the other training models. The IWO-CS-ELM not only out-performed the other methods by recording better classification accuracy in most of the cases, but also significantly reduced the training time of the mode. IWO-CS-ELM is also more robust and stable than the other techniques and also decreased the size of the network in many cases.

References

1. Kotsiantis S, Zaharakis I, Pintelas P (2007) Supervised machine learning: a review of classification techniques. *Front Artif Intel Appl* 160:3
2. Mohapatra P, Chakravarty S, Dash PK (2015) An improved cuckoo search based extreme learning machine for medical data classification. *Swarm Evol Comput* 24:25–49
3. Razavi S, Tolson BA (2011) A new formulation for feed forward neural networks. *IEEE Trans Neural Netw* 22(10):1588–1598
4. Ding S, Su C, Yu J (2011) An optimizing BP neural network algorithm based on genetic algorithm. *Artif Intell Rev* 36(2):153–162
5. Kaya Y, Kayci L, Tekin R, Faruk Ertuğrul, Ö (2014) Evaluation of texture features for automatic detecting butterfly species using extreme learning machine. *J Exp Theor Artif Intell* 26(2):267–281
6. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feed-forward neural networks. *Neural Netw* 2:985–990
7. Huang GB, Chen L (2007) Convex incremental extreme learning machine. *Neurocomputing* 70(16–18):3056–3062
8. Huang GB, Chen L (2008) Enhanced random search based incremental extreme learning machine. *Neurocomputing* 71(16–18):3460–3468
9. Wang L, Huang Y, Luo X, Wang Z, Luo S (2011) Image deblurring with filters learned by extreme learning machine. *Neurocomputing* 74(16):2464–2474
10. Lim JS, Lee S, Pang HS (2013) Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations. *Neural Comput Appl* 22(3–4):569–576
11. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. *Ecol Inform* 1(4):355–366

Chapter 17

Intrusion Detection System Against Malign Packets—A Comparative Study Between Autoencoder and Ensemble Model



Adit Sadiwala, Kishan Rathore, Yash Shah, Harsh Shah and Kriti Srivastava

1 Introduction

Technological advancements in the last decade have certainly raised concerns about security in systems. With these advancements, network traffic has seen exponentially increased. This Internet traffic is the perfect ground for an attacker to invade systems' privacy. Therefore, data confidentiality, data integrity, and service availability should be given prime importance. Hence, developing robustness in intrusion detection system is the need of the day. The solution provides a huge impact on the development of the robust intrusion detection system which is a key factor in the overall success for all the real-world systems. The detection of the malicious activities is one of the most important strategy of the intrusion detection system.

IDSs are broadly classified into two categories:

- (1) Network intrusion detection system (NIDS).
- (2) Host-based intrusion detection system (HIDS).

NIDs can be placed at points within a network to monitor traffics, whereas HIDs run on individual network devices [1]. However, there is a possibility for the new

A. Sadiwala (✉) · K. Rathore · Y. Shah · H. Shah · K. Srivastava
Dwarkadas Jivanlal Sanghvi College of Engineering, Mumbai, India
e-mail: asadiwala_97@hotmail.com

K. Rathore
e-mail: rathorekishan0@gmail.com

Y. Shah
e-mail: yashketanshah29@gmail.com

H. Shah
e-mail: harshsh31@gmail.com

K. Srivastava
e-mail: kriti.srivastava@djsce.ac.in

attackers who can evade the security of the system as they are new and do not have any signatures in the real world. The signature-based IDS are not capable of detecting new malicious activities, whereas anomaly-based IDSs are capable of capturing unknown malicious activities by measuring the degree of deviation of incoming data streams from what is considered to be the normal data profile. An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner [2, 3]. An autoencoder is a neural network which reconstructs the input provided on its output nodes. The main goal of this architecture is to learn approximations of input. As a result, the model learns useful properties of the input data in order to achieve necessary reconstruction ability. Loss function is the function calculated by the rectified linear unit method which will classify based on dissimilarity between benign data and malicious data. Classifier works in the real-world environment are vulnerable to adversarial drift [4]. Adversarial drift causes change in the distribution of data over period of time. Several methods are used to handle the adversarial drift in the dataset. Due to adversarial drift, the properties of data are changed which further leads to contamination. As iteration of model increases, the chance of adversarial drift within the dataset increases. Hence, our solution is to handle such problems that occur over a period of time.

2 Approach

2.1 Autoencoder

Autoencoder is a type of neural network which aims to map its input to its output. They work by compressing input into a latent representation, and this representation is used to create output.

This neural network comprises of two parts:

1. Encoding layer: This layer compresses its input to the latent space representation which can be represented as $h = f(x)$.
2. Decoding layer: This layer augments latent representation to output layer, which can be represented as $g(h) = r$.

Hence, we can say that main aim of autoencoder is to evaluate the value of r which is close to x . Also, while trying to generate the output, autoencoder acts as a feature selector on latent representation h . This is achieved by creating constraints for the model. We need to ensure that dimensions of latent space should be smaller or equal than input layer. If dimensions of representation h is greater than input, we could lose all the distribution of data (Fig. 1).

Traditionally, autoencoders were used as dimensionality reduction technique or feature selector. Recently, relations between autoencoders and latent variable models, autoencoders are also considered for subspace analysis techniques [6].

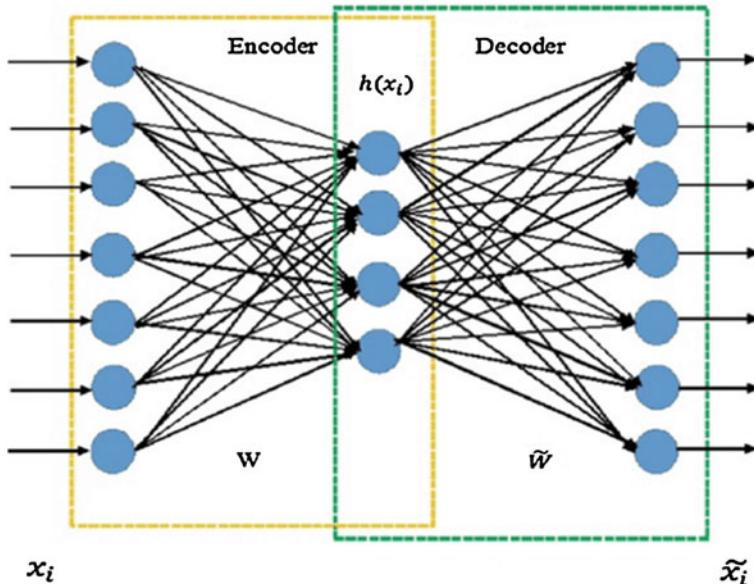


Fig. 1 General autoencoder architecture [5]

2.2 Random Forest Algorithm

Random Forest is a supervised learning algorithm. This algorithm is based on decision tree algorithm. Many decision tree results are combined to get the label for the final class. Basically, algorithm builds an ensemble of decision Tree, mostly trained with bagging algorithm. With help of ensemble model, we get better accuracy and stable prediction when compared to decision tree.

Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Instead of combining decision tree with bagging classifiers, we can use classifier class of random forest. It adds additional randomness to the model, while growing the tree. Instead of searching for the most important feature while splitting a node, it searches for the best feature amongst a random subset of features. This results in a wide diversity that generally results in a better model.

2.3 Robustness in Autoencoder and Random Forest

In previous sections, random forest and traditional autoencoder provided good accuracy, but for IDS, we need to ensure that we get lesser false positive rate. Out of the two, robust autoencoder satisfies this criterion. For checking the robustness, we have made a simulation where after classifying the data, normal instances will be added to training pool for retraining. This simulation happens till all the malicious instances are used in testing dataset. At each iteration, we are comparing the result

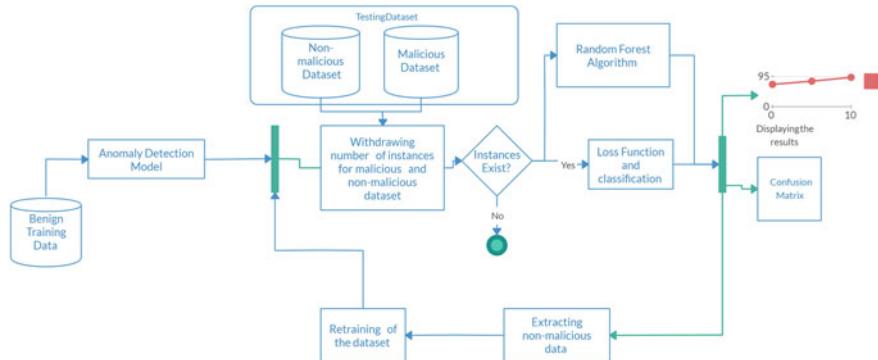


Fig. 2 Robust autoencoder

of autoencoder with random forest using confusion matrix to show the contamination error. As number iterations increases, robust random forest algorithm does not cope up with contamination and the accuracy decreases, but in case of the robust autoencoder, the accuracy remains almost constant even with contamination.

Although our autoencoder has number of false positive higher than random forest, the prediction of attacks is much higher for autoencoder which is 95% as compared to 62% of random forest. The priority in real-world scenario for IDS is to minimize the attacks, but we can afford we re-transmit the package for better security. Our robust autoencoder proves to handle unknown attacks with greater success, and in addition, classification rate of false negative is less (Fig. 2).

3 Experiments

3.1 Data Set

The initial version of the data set is introduced by DARPA in 1998, KDDCUP99 [7] and later it became mainstream dataset for intrusion detection by containing simulated normal and malicious traffics on a typical US Air Force LAN [6]. Though the process of creating the dataset was immensely criticized by McHugh [8] and many other members of the community, NSL-KDD was created in attempt resolving some statistical issues (e.g., data redundancies) by Tavallaei et al. [9]. Although the dataset does cover all the cyberattacks found in modern world, it is still used to evaluate anomaly detection by the research community. For the in-depth explanation of the content of NSL-KDD, we refer readers to the paper published by Tavallaei et al. [9].

Pre-processing: NSL-KDD data set is already divided into training and testing set. Both splits are distributionally different [6]. Testing set contains certain attacks

which were not encountered during the training phase, which is important to check robustness of the algorithm. Since our motive is to train the algorithm for normal and malicious instances, we have replaced all malicious labels as attack and the rest as normal instance. Sampling technique is used to create benign and malicious instances. For robust algorithm, batches are again randomly sampled, so it does not use same distribution for that particular iteration.

Data set contains numeric and nominal values. Autoencoder neural network takes only numeric values as input. To handle those nominal values, one hot encoding is used. One hot encoding helps us to retain all the information from the data instead of converting nominal data into numeric data by ordering. If ordering is used, then data with higher value will get greater preference. Also, the numeric values in the data set had larger variance. Hence, to handle variance, normalization techniques are used to scale the data into specific range.

3.2 System Description

For baseline comparisons, we have implemented traditional autoencoder and random forest algorithm. We have also compared our model with robustness of deep autoencoder proposed by Madani and Vlajic [6]. As described in that paper, desired rate for false positive is 2% which is frequently used as the target rate in the literature of IDS design. That rate is used to define the threshold value. The input and output layers consist of 122 Relu nodes, the representation layer contains 4 Relu nodes, and encoding and decoding layers each contain 8 Relu nodes.

$$R(z) = \max(0, z) \quad (1)$$

Out of all activation functions such as sigmoid and tanh, Relu proved to be better amongst all. Kingma and Ba [10] proposed Adam optimizer and we used their algorithm to stochastically compute the optimal gradient during training of the autoencoder which outperformed other gradient descent methods like RMSProp and AdaGrad. This algorithm is used because of the lack of smoothness in the error surface that had trapped many deterministic optimization methods such as normal gradient descent in different local minima which resulted in a poor performance. Advantages of two different optimization methods are combined together, AdaGrad to deal with sparse gradients and the ability of RMSProp to deal with non-stationary objects.

For classification in robust autoencoder, we need to select a threshold value by looking at reconstruction loss. In case of normal instances, we get similar reconstruction error and for malicious instances produce different values. This value is used for classification in our testing data (Fig. 3).

In random forest algorithm, tuning of parameters is essential. Tuning can take the results from sub-optimal to optimal level. It can drastically change the performance and speed of the algorithm. In previous section, we discussed random forest algorithm as an ensemble model. Therefore, each individual tree can select combination for

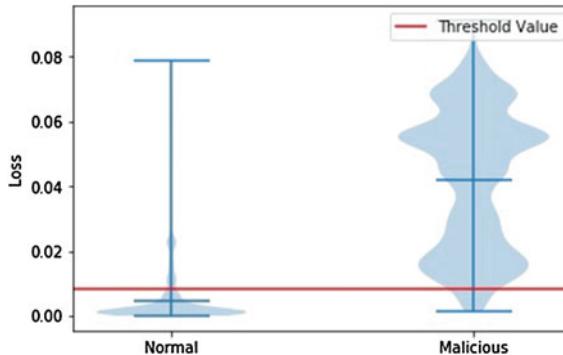


Fig. 3 Threshold for reconstruction loss

features. Max feature parameter helps us to limit the maximum number of feature in an individual tree. In our algorithm, max features are limited to square root of total features. Increasing max features generally improves performance at each node but it tends to take a hit on speed. Also, it can decrease the diversity of individual tree which is the main feature of random forest algorithm. n estimators is taken as 50, which means that minimum 50 individual trees are built before we start maximum voting phase (Fig. 4).

3.3 Result Analysis

In order to study robustness in both the algorithms, we need to first evaluate response of each algorithm on NSL-KDD data set. The accuracy score of the autoencoder model was 90.30%. The accuracy score of the random forest classifier model was 77.38%. Figures 5, 6, 7 and 8 shows the confusion matrix of the predictions.

As we have discussed in Sect. 2.3, our priority is to minimize the attacks. But our aim now is to suppress the false positive rate. This is accomplished by robust autoencoder. The receiver operating characteristic (ROC) curve shows the superiority of the autoencoder in detecting network anomalies compared to random forest. The training instances used to train the random forest and the autoencoder for construction of the ROC curve were empty of any malicious data point. Therefore, it represents non-malign scenario.

For checking the robustness, we made a simulation where after classifying the data the normal instances will be added to training pool for retraining. This simulation happens till all the malicious instances are used in testing data set. At each iteration, we compared the result of autoencoder with random forest using confusion matrix to show the contamination error. This result is important as it shows how much the malicious instances would contaminate the system and the performance of our classifier models. From the above figures, we can see that as the number of iterations

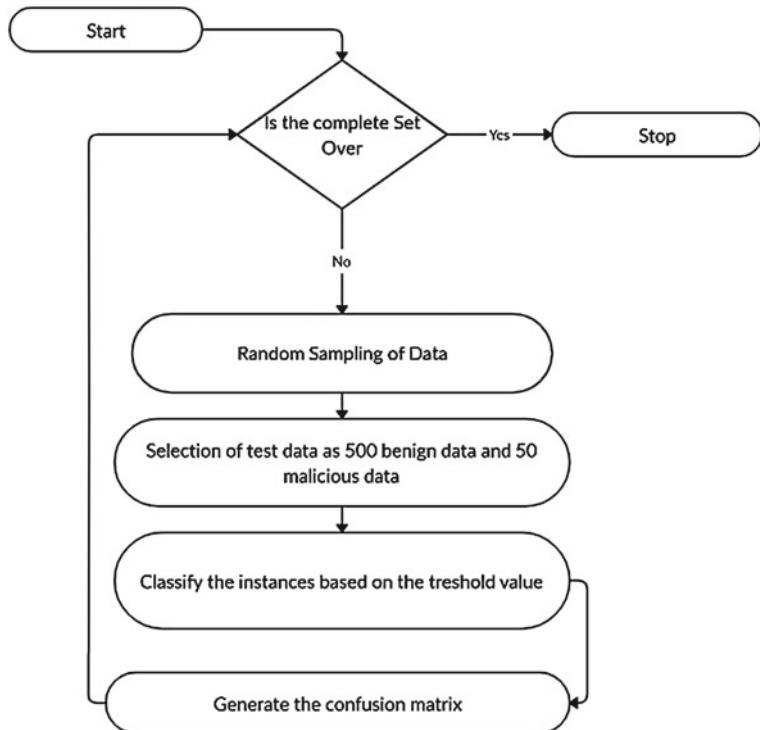


Fig. 4 Algorithm for checking robustness

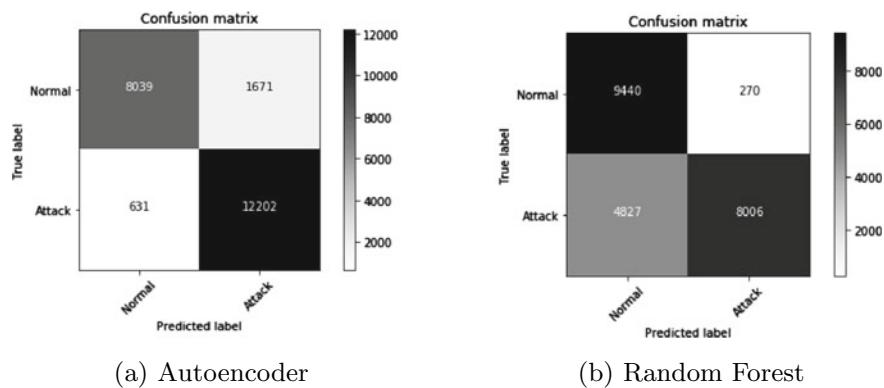


Fig. 5 Confusion matrix

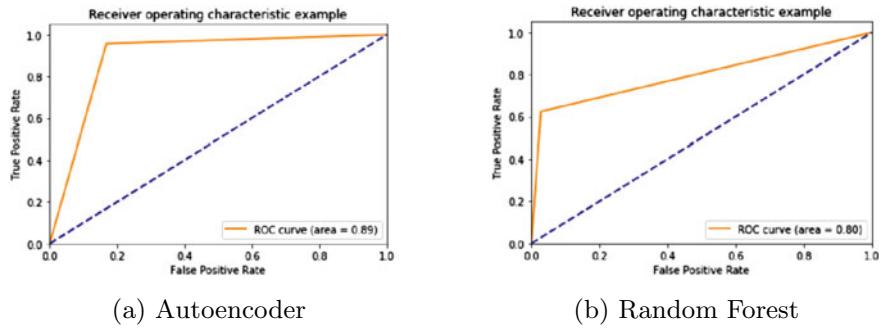


Fig. 6 Receiver operating characteristic

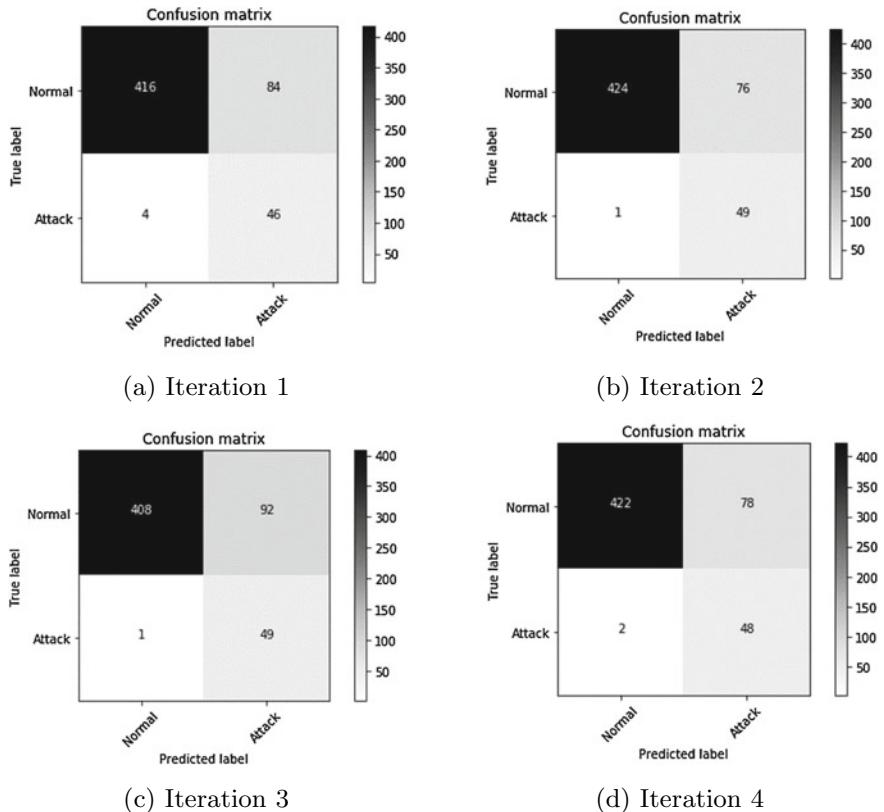


Fig. 7 Confusion matrix for robust autoencoder

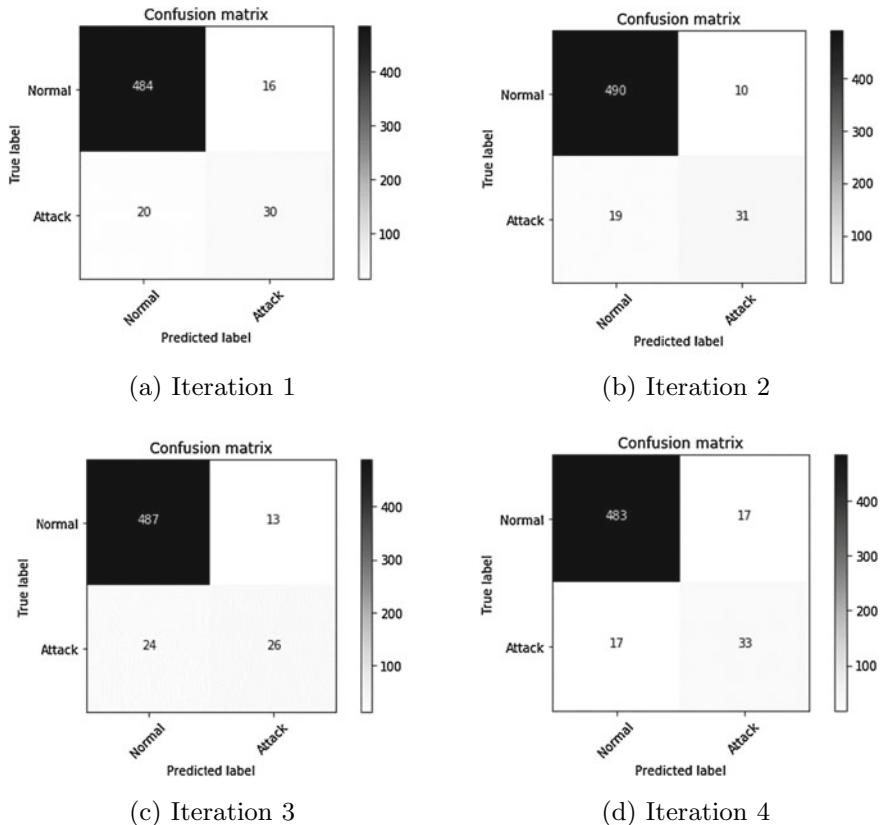


Fig. 8 Confusion matrix for robust random forest algorithm

increases, the random forest does not cope up with contamination and the accuracy decreases, but in case of the autoencoder, the accuracy remains almost constant even with contamination. This shows that the autoencoder is more robust than random forest.

4 Conclusion

In conclusion, autoencoder provides a more robust solution compared to random forest algorithm; moreover, robust autoencoder will eventually determine the number of attacks with great possibilities compared to random forest. The only drawback for autoencoder is that it generates greater false negatives compared to random forest, but we can look that as a positive thing as true negatives are almost classified correctly.

The robustness was evaluated by making a simulation depicted in a real-world scenario. Our autoencoder performed well in this scenario than the random forest which is a ensemble model. Thus, we observed that the autoencoder even though an unsupervised learning model, we can manipulate it to become like supervised learning model. In this project work, we have successfully observed the reconstruction ability of autoencoders which is deep learning for data instances drawn from different data sets. In cybersecurity where normality is a moving target, it has shown that autoencoders can capture new features from learning to reconstruct such data instances. Most anomaly-based IDSs detect normal attacks at the expense of high rate of false alarm. Since most of the detection alarm will be seen and evaluated by human being, high rate of false positive (i.e., false alarm) can quickly annoy the user and make the detection system useless. Thus, adaptiveness to the occurring concept drift can potentially lower such false alarms. Self-adaptiveness is one of the main requirements of developing a modern IDSs which is comfortably done by autoencoders. In our project, we have investigated a naive self-adaptive model and demonstrated that in reality the presence of contaminating instances in retraining data sets is inevitable. Thus, through our proposed novel approach, we have compared the robustness of the autoencoder-based IDS with one of the random forest classifier that is well adapted by the industry by its use in ensemble model which usually attains high number of accuracy. Through this comparative study, the autoencoder IDS maintained a more stable rate of detection even in the presence of contaminations in its training data set. The anomaly-based IDS model has a higher number of accuracy than the signature-based IDS model.

References

1. Deepa J, Kavitha V (2012) A comprehensive survey on approaches to intrusion detection system. Proc Eng 38:2063–2069. ISSN 1877-7058
2. Liou C-Y, Huang J-C, Yang W-C (2008) Modeling word perception using the Elman network. Neurocomputing 71(1618):3150. <https://doi.org/10.1016/j.neucom.2008.04.030>
3. Liou C-Y, Cheng W-C, Liou J-W, Liou D-R (2014) Autoencoder for words. Neurocomputing 139:84–96. <https://doi.org/10.1016/j.neucom.2013.09.055>
4. Sethia TS, Kantardzica M (2018, March 28) Data Mining Lab. University of Louisville, Louisville, USA (Handling Adversarial Concept Drift in Streaming Data)
5. Ahmed HOA, Wong MLD, Nandi AK (2018) Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features. Mech Syst Sign Process 99:459–477. <https://doi.org/10.1016/j.ymssp.2017.06.027>
6. Madani P, Vlajic N (2018) Robustness of deep autoencoder in intrusion detection under adversarial contamination. In: Proceedings of the 5th ACM annual symposium and Bootcamp on hot topics in the science of security, p 1
7. Cup K (1999) Dataset, 72. Available at the following website <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
8. McHugh J (2000) Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory. ACM Trans Inf Syst Secur (TISSEC) 3(4):262–294

9. Tavallae M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD cup 99 data set. In: IEEE symposium on computational intelligence for security and defense applications, 2009. CISDA 2009. IEEE, New York, p 16
10. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)

Chapter 18

Chest Pathology Detection Using Medical Imaging



Devansh Shah, Purav Nisar and Pankaj Sonawane

1 Introduction

The human thoracic cavity is the second **largest hollow space** of the body. The most commonly used image modality to identify thoracic abnormalities is X-ray imaging. X-rays capture the image using ionizing radiation. Specialists use X-ray sweeps to analyze the thoracic depression and an assortment of conditions from pneumonia to fibrosis. In any case, essentially, restorative pictures are perplexing and uproarious. This prompts the need for procedures that diminish challenges in the examination and improve the nature of yield. While trying to diminish manual exertion for medicinal picture examination, the paper proposes a program framework that pursues a three-overlay approach: image pre-handling pursued by pathology recognition and afterward heatmap generation.

D. Shah (✉) · P. Nisar · P. Sonawane
Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai,
India
e-mail: devansh.shah2018@djsce.edu.in

P. Nisar
e-mail: purav.nisar@djsce.edu.in

P. Sonawane
e-mail: pankaj.sonawane@djsce.ac.in

2 Problem Definition

2.1 Problem

In India, the ratio of radiologists to population is highly skewed. According to the latest statistics, there is approximately one radiologist for approx. million people in India. This implies that most of the radiologists are overworked, and this leads to a rise in human errors and a highly uneven demand in supply chain.

2.2 AI in Medical Imaging

Chest X-rays have been a primary yardstick for performing any analytical tests, whereas comprehending X-ray is also a task that finds itself into the significant anomaly. Artificial intelligence with smart computer vision here comes to the picture to aid us in the removal of any possible errors or reduce them to a great extent. The factors such as their affordability, speed, accuracy and precision are important. Chest X-rays can further be used as a medium to analyze patients and diagnose these diseases. Hence, a single modality along with the AI can couple up to give an excellent solution for pre-medical detection and diagnosis. The chest X-ray dataset primarily unveiled by NIH and has more than 90,000 + X-ray along with their respective labels for 14 diseases.

2.3 Proposed Solution

We propose a system which aids the radiologists to detect the probability of the disease/deformity as well as a heatmap which localizes the area of the image most indicative of the pathology. In the first phase, the system works solely with chest X-rays. This paper implements the CheXNet model, which is a 121-layer dense convolutional neural network which is trained on ImageNet, all of which are separately annotated and tagged onto 14 thoracic diseases. The algorithm classifies the following thoracic pathologies. CheXNet outputs a vector ' t ' of discrete taggings indicating the absence/presence of each of the following 14 disease classes and in the presence of the disease, the image which localizes the affected area is displayed:

- (1) Atelectasis
- (2) Cardiomegaly
- (3) Effusion
- (4) Infiltration
- (5) Mass
- (6) Nodule

- (7) Pneumonia
- (8) Pneumothorax
- (9) Consolidation
- (10) Edema
- (11) Emphysema
- (12) Fibrosis
- (13) Pleural thickening
- (14) Hernia.

3 Literature Survey

Image processing using deep learning took off during the last few years since the introduction of the ImageNet dataset and improved processing speed. A number of tasks like object detection and classification have been achieved with accuracy higher than human accuracy. Inspired by the success of convolution neural networks in image processing, they were used in medical image processing as well. A number of applications of DL in field of radiology are described. However, the model was trained from scratch which would lead to a severe increase in processing time.

Most motivated by the accomplishment for convolution neural systems in picture handling, they were utilized in therapeutic image processing too. Various uses of profound learning in radiology are portrayed in these of which some are incorporated. However are not constrained to a grouping of illness utilizing X-rays, CT scans, MRIs, Division of organs, substructures of injuries, (a location which confines them). Then these include influenced territory in the picture, picture enrollment, and picture improvement.

In medicinal image processing, we, by and large, utilize a trained model and afterward influence it to gain proficiency with the particular highlights identified with the job needing to be done. One such generally utilized system is the DenseNet, a kind of convolution neural system. Huang et al. [1] demonstrate DenseNets viability on a few benchmark datasets and contrast diverse best-in-class structures, particularly ResNet and its variations. It likewise portrays the design of DenseNets in detail. Rather than illustration of authentic power from amazingly profound or wide structures, work has been done as of late in the arrangement and discovery front of sicknesses from medicinal pictures of different organs utilizing converts as their essential model.

Esteva et al. [2] depict a cutting-edge calculation for skin malignant growth grouping. They utilized a huge dataset of 129,450 pictures and 757 instructional courses. Their calculation utilizes a Google Inception CNN pre-trained on the ImageNet dataset and tweaked on their skin sore dataset. It accomplishes results at standard.

Of human dermatologists, another great execution of denseness as a benchmark structure for identification is finished by Grewal et al. [3]. They use DenseNet combined with LSTM for the discovery of discharge in cerebrum CT scans.

Pathology	Wang et al. (2017)	Yao et al. (2017)	CheXNet (2017)
Atelectasis	0.716	0.772	0.8094
Cardiomegaly	0.807	0.904	0.9248
Effusion	0.784	0.859	0.8638
Infiltration	0.609	0.695	0.7345
Mass	0.706	0.792	0.8676
Nodule	0.671	0.717	0.7802
Pneumonia	0.633	0.713	0.7680
Pneumothorax	0.806	0.841	0.8887
Consolidation	0.708	0.788	0.7901
Edema	0.835	0.882	0.8878
Emphysema	0.815	0.829	0.9371
Fibrosis	0.769	0.767	0.8047
Pleural Thickening	0.708	0.765	0.8062
Hernia	0.767	0.914	0.9164

Fig. 1 Relative comparison of CheXNet

After the arrival of the vast explained ChestX-14 database, investigation in the fields of location, forecast, and confinement of thoracic sicknesses has expanded exponentially.

Islam et al. [4] recognize variations from the norm and limit the influenced territory in Chest X-rays utilizing CNN. This uses the beforehand accessible biggest dataset of chest X-beams, the Indiana dataset, which comprises 7824 chest X-rays. It analyzes the effectiveness of differently trained CNN model—AlexNet.

ResNet, VGG, and their blends. This paper utilizes a similar assessment measurement for correlation as characterized in this paper for discovery—area under the curve (AUC), sensitivity, and explicitness (Fig. 1).

Zhou et al. [5] utilize a pitifully directed versatile DenseNet for classifying thoracic diseases and identifying variations from the norm. It utilizes a feebly administered system in light of the fact that not every one of the pictures in the dataset is physically commented on for portrayal of influenced territory.

4 System Design

4.1 Design Architecture

Raw data of chest X-rays is given as input to the system. The first stage is pre-processing where size reduction, noise removal, and data augmentation are done. Later on, the model is trained on this training dataset using the deep CNN model. Using this model, F1 score and AUC are calculated for the test dataset and the

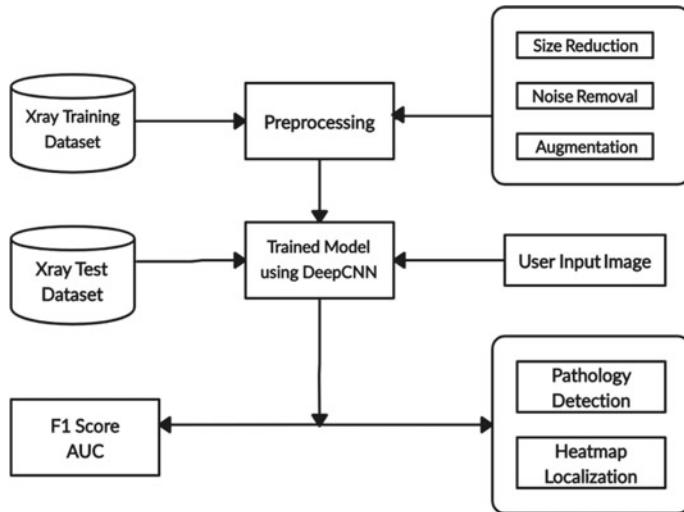


Fig. 2 Model's design architecture diagram

corresponding pathology is detected. Localization using a heatmap is also an output of the system (Fig. 2).

5 Methodology

5.1 Training Phase

For training the model, first, the dataset is split explicitly into a training dataset and testing dataset.

The next step involves the creation of a text file that consists of the names of the pathologies along with a hot vector for the representation of the corresponding disease.

Example: images_011/00027736_001.png

Its respective pathology vector: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The vector consists of 14 bits wherein the value 0 represents absence of the disease and the value 1 represents presence of the same.

Later, the Descent 169 is loaded with its trained weights.

Once the initialization is done, a transform sequence is generated to be applied to each image. This sequence comprises three steps:

- Resizing the image to dimensions of 256×256 ;
- Cropping them into 224×224 to obtain a downsampled dataset;

- Horizontal flipping is done to double the availability of the data items.

Once the pre-processing is done, the train.txt file is read lines, and a list of image names and image labels is created. The data is then loaded using a batch size of 16. A list is instantiated which contains RGB matrices of 16 images. After applying transformation sequence on these images, a vector of size $16 \times 3 \times 224 \times 224$ is obtained.

For the first run, the model had been initialized with its trained weights. But for all following iterations, a previous checkpoint (model trained with weights) is loaded (values of state direct and optimizer) into the working memory.

It compares the achieved result with the expected result and backpropagates the loss. The following variables are stored in a dictionary so as to fine-tune a model which could be used during testing: ‘best_loss’, ‘epoch’, ‘state_dict’, and ‘optimizer’.

Adam Optimization Algorithm is used to update iteration-based network weights of the model on the training data. It is also computationally efficient, and that is why it is preferred for this system.

Hyperparams: learning rate—0.0001, and betas (0.91, 0.999) are the parameters used.

Binary cross-entropy loss. It is a sigmoid activation plus a cross-entropy loss function which is independent for each vector component. The maximum number of training epochs is 100 (Figs. 3, 4 and 5).

Confusion Matrix

The *confusion matrix* (or *error matrix*) is one way to summarize the performance of a classifier for binary classification tasks. This 2×2 matrix comprises segments listing the number of occurrences as absolute or relative ‘actual’ versus ‘predicted’ (Fig. 6).

5.2 Testing Phase

The above image shows the input of CheXpert provided to the system. This input is obtained from NIH, Stanford’s universal dataset of chest X-rays.

Again the transformation sequence is applied to the testing dataset images. Once the trained model is loaded (the one achieved with minimum loss during training), the images from the testing dataset are evaluated using this model.

We get the probability of the presence of each disease. The prediction is made on the basis of the probabilities obtained, and the three diseases with the highest probabilities are shown as the output of the prediction. The accuracy of the system is obtained by comparing the outputs obtained from the model for each set of inputs with the desired set of output as labeled in the dataset (Fig. 7).

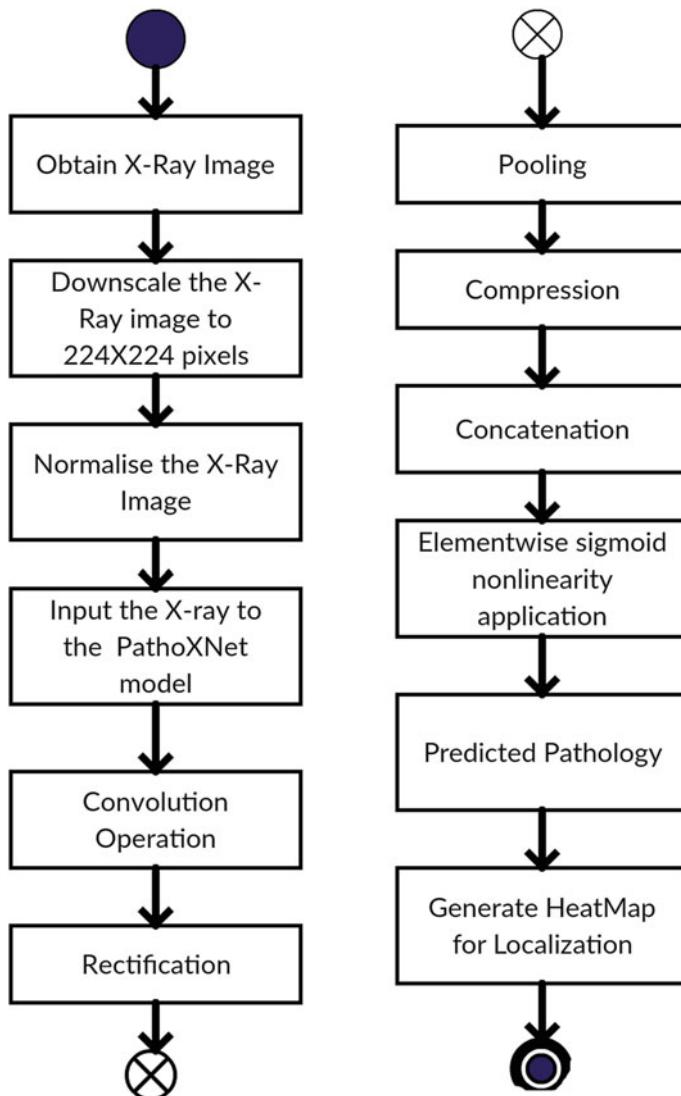


Fig. 3 Activity flow diagram

6 Conclusion

Pathology detection is a specialized area in the field of biomedical sciences. These systems are concerned with providing relevant and accurate detections of pathogenic diseases existing within the human body in response to the X-ray inputs.

Fig. 4 Original image**Fig. 5** Downsampled image**Fig. 6** Confusion matrix

		Predicted class	
		<i>P</i>	<i>N</i>
<i>P</i>	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

After pre-processing of the input data, the model is trained using deep CNN such that it becomes capable of classifying and detecting the diseases from the raw X-ray inputs provided by the end-users/patients. It then generates the final output where the



Fig. 7 Chest X-ray testing

presence/absence of the diseases is evaluated, and it localizes the area of the affected region.

This report provides an overview of chest pathology detection using medical imaging and its system architecture with respect to the components that were covered and the approaches that were followed. After extensive research into multifarious models, it provided an overview of a proposed system with an analytical discussion of the limitation and use-cases affecting it. This report also provides complete knowledge for understanding the working of the system, the scope, and the various outcomes of the model depending on the image input.

6.1 Future Scope

This system can be extended to other imaging modalities like CT scan, MRI, etc. It can also be used to study more details of the anatomical and thoracic structure of the chest like blood vessels, tissues, etc.

The system was designed to focus only on abnormality detection from chest X-rays. But the whole concept can be extended by adding additional functionalities which would enable the system to accept images (X-rays/CT scans, etc.) of other organs as input and identify the existence of any abnormalities. It would become very useful and purposeful if these functions were brought under a single system.

References

1. Huang G et al (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). 22
2. Esteva A et al (2017) Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542(7639):115
3. Grewal M et al (2018) RADNet: radiologist level accuracy using deep learning for hemorrhage detection in CT scans. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). IEEE
4. Islam MT et al (2017) Abnormality detection and localization in chest X-rays using deep convolutional neural networks. arXiv preprint [arXiv:1705.09850](https://arxiv.org/abs/1705.09850)
5. Zhou B, Li Y, Wang J (2018) A weakly supervised adaptive DenseNet for classifying thoracic diseases and identifying abnormalities. arXiv preprint [arXiv:1807.01257](https://arxiv.org/abs/1807.01257)

Chapter 19

Disorder Detection in Tomato Plant Using Deep Learning



Saiqa Khan and Meera Narvekar

1 Introduction

Plants are an integral aspect of human life. The agricultural landmass is more than just being a feeding source in today's world. Indian economy highly depends on agriculture. Smart farming is very important for describing the ongoing challenges occurring in agricultural production due to rise in uncertain climatic conditions. Therefore, disease detection plays an important role in agricultural field. Tomato is by far the most consuming crop in the world. As per the report published from the Food and Agriculture Organization of the United Nations around the world, it says that approximately 170 kt of tomatoes was produced in the year 2014 [6]. However, this count has gone up substantially to have approximately 170.8 million tons in the year 2017. China is on top for tomato production as it delivers 31% of the total production, while India accounts for 18.7% of the total production [7]. Tomatoes are more susceptible to pests and diseases that occur in the fields. The diseases and pests affect not only the fruit but also the other parts of the plant that is the roots, the stems and the leaves of the tomato plants. However, leaves are considered as easily observable part for disease identification. Therefore, current research focusses on detection based on patterns observed on leaves. Tomato is impacted through various disorders, namely early blight, late blight, etc.

For detecting diseases in tomatoes, potatoes, banana and grapes and other vegetable and fruits, machine learning and computer vision techniques have been applied

S. Khan (✉)

Department of Computer Engineering, M. H. Saboo Siddik College of Engineering,
Mumbai, India

e-mail: saiqa.comp@gmail.com

M. Narvekar

Department of Computer Engineering, D. J. Sanghvi Engineering College
Mumbai, Mumbai, India

e-mail: meera.narvekar@djsce.ac.in

in recent times. Motivated by development in computer vision, many authors have done substantial work to produce significant results in image classification domain. This paper proposes a new system based on CNN for tomato plant disease detection. There is a need to have more diversified dataset that should be capable enough to consider all possible cases.

This is an era where deep learning and neural networks are combined to make deep neural networks. They are advanced and modern technique for image classification. They are applied in many diverse fields of application. Neural networks yield a mapping between an input (diseased plant) and an output (disease). Neurons are basically mathematical functions that accept numerical input and provide numerical output. Deep neural networks are comprised of input layer, a number of processing layers and an output layer. The basic task of network training is to fine-tune hyperparameters in order to improve training process. Figure 1a shows the production of tomatoes worldwide showing that China is the largest contributor followed by India and USA, and Fig. 1b shows type of disorders for tomatoes. Generally, there are two factors that can affect tomato plant: living (biotic) and non-living (abiotic) agents. Different living agents include fungi, insects, viruses and bacteria. Non-living agent includes various environmental effects such as rapid insufficient nutrients, temperature change, excess moisture, poor soil pH and high humidity conditions. Figure 1b shows diseases of tomato plant considered; they are:

1. Early Blight: Spots of approximately 1/4 to 1/2 in. in diameter appear on the leaves of tomato plant. Concentric circles with tan centers and appearance of yellow halos around the edges.
2. Late Blight: Spots usually appear on the edges of tips of foliage in pale green color and then turn to brown in color and then to black. Fuzzy mole appears on the back side of leaves in humid conditions.

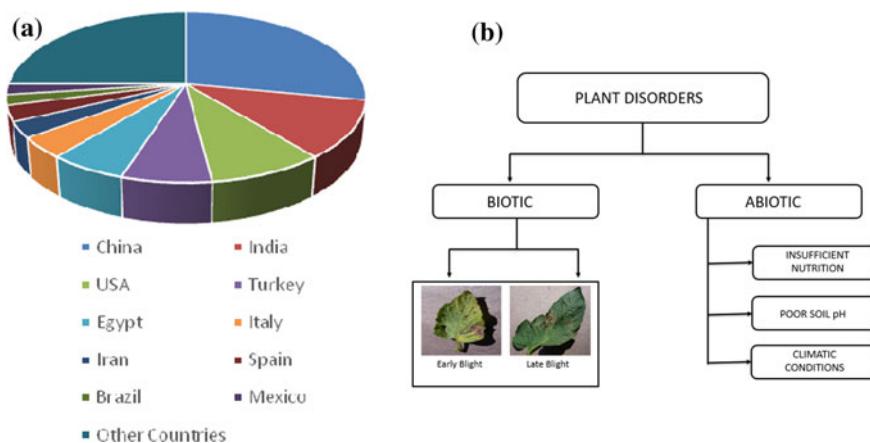


Fig. 1 **a** Tomato production worldwide and **b** tomato disorders

Section 2 discusses the related study of the proposed system. Section 3 presents materials and methods adopted in this work. Section 4 deals with results and discussion. Section 5 describes conclusion and future work.

2 Related Work and Motivation

Deep learning techniques have shown immense improvement in the field of agriculture that includes disease detection in plants since 2014 (Table 1). Table 1 shows study experiments of different plants and their species with promising results. Convolution neural network based significant studies are presented by Srivastava [8], Sabrol and Kumar [9].

Mohanty [1], Amara et al. [11], Fuentes et al. [10], Oppenheim and Shani [13], Liu et al. [14], Lu et al. [16].

Parameters used in the table are described as follows:

1. Deep Learning Architecture: The convolution neural network architectures include special kind of multilayered neural networks, designed to identify visual patterns directly from pixel images from minimal preprocessing.
2. Accuracy: The correct amount of classification is achieved by a CNN model. There are two types of accuracy [17].
 - Training Accuracy: The accuracy of the model on the dataset was constructed.
 - Testing Accuracy: The accuracy of the model on an instantaneous sample.

3 Materials and Methods

We have analyzed 13,548 images of tomato plant leaves from different sources that are segregated using three class labels such as Tomato_Early_Blight, Tomato_Late_Blight and Tomato_Healthy. Proposed system focusses on two major tomato leaf diseases, namely early blight and late blight. Using these labels, we predict the disease by testing an unseen image of the plant leaf. Figure 2 shows the distribution of the dataset of our system. The above table (Table 2) describes statistics of the dataset collected from different sources that includes plant village [1], Internet data [18] and real-world images from Tansa Farm captured in an uncontrolled environment.

For all the experiments, two different categories of dataset are considered: 1. color image dataset and 2. segmented image dataset. Initially, execution is carried out on color image dataset; then, experiment is finally executed on the dataset where segmented data was used for execution. Segmentation consists of images any without background information that might disturb results while training. Segmentation is

Table 1 Related work using deep learning for plant disease detection

S. No.	Year	Work cited	Plant	Disease	Deep learning (DL) architecture	DL algorithm	Accuracy
1.	2016	[9]	Tomato	Bacterial leaf spot, fungal septoria leaf spot, bacterial canker, fungal late blight, tomato leaf curl	Adaptive neuro-fuzzy classification model	Hybrid learning algorithm, multilayer feed-forward back-propagation network	Best classification accuracy is obtained by feed-forward back-propagation classifier of 87.2%
2.	2016	[1]	Tomato	26 different diseases	CNN (AlexNet, GoogleLeNet)	^a	99%
3.	2017	[10]	Tomato	Powdery mildew, canker, gray mold	Faster R-CNN, SSD, RFCN	VGG-16, ResNet-50, ResNet-101, ResNet-152	80% (mean accuracy of entire system performance)
4.	2017	[11]	Banana	Black Sigatoka, banana speckle	CNN (LeNet)	Convolution pooling and classification	92–99%
5.	2017	[12]	Tomato	Tomato leaf diseases	CNN (AlexNet, GoogleLeNet)	Steps 1. Pre-training, 2. training, 3. disease classification, 4. symptom detection	99%
6.	2017	[13]	Potato	Black dot disease, common scab, black scurf, silver surf	CNN (VGG)	^a	96%
7.	2018	[14]	Apple	Alternaria leaf, spot mosaic, rust	CNN (AlexNet)	^a	98%
8.	2018	[15]	Different plant species	Anthracnose, southern corn leaf blight, Phaeosphaeria leaf spot, Physoderma brown spot	CNN (GoogleLeNet)	^a	81%

^aNot mentioned

carried out by a script that automates the removal of extra background information. Our methodology mainly consists of the following steps:

Data Preprocessing. For our work, dataset contains images of healthy and diseased tomato leaves. Each image has three color channels, namely red (R), green (G) and blue (B). Experiments are carried out where our approach will be tested on two different sets of images: 1. color images and 2. segmented images. In proposed

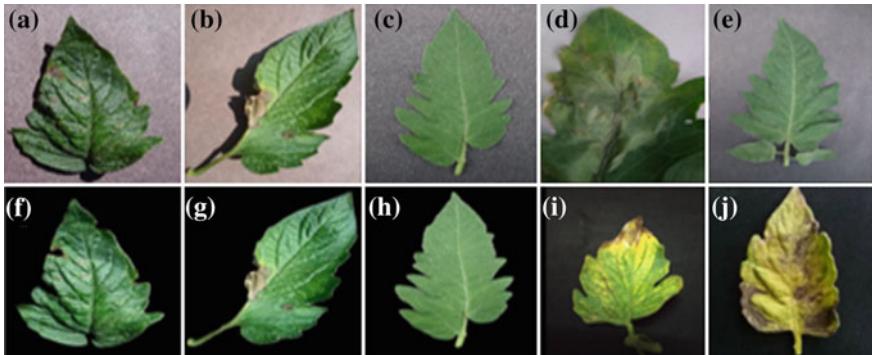


Fig. 2 Sample images from the plant village dataset, Internet dataset and real-world dataset used in various experimental configurations (a–e), non-segmented images (f–j) and segmented images

system, the images are resized to 256×256 pixels, and also we perform image augmentation that artificially increases the training dataset size and enhances the model's performance by using various augmentation techniques like random rotation, shifts, shear and flips, etc.

Feature extraction. An integral part of CNN is feature extraction. It is through feature extraction that high-level features are extracted from the available input images to create effective neural network solutions. Our model consists of a series of convolution and pooling layers [19]. Feature extraction generates:

Convolution Map. The main elementary unit in CNN architecture is convolution maps. Convolution map is used to get features from the input image. In this system, convolution is applied on the input data (infected/healthy tomato leaf image) using convolution filter to produce feature maps (in our system 32 dimensions). This is done by sliding the kernel matrix (3×3) over the input image matrix (256×256). At each location of the input matrix, we perform element-wise matrix multiplication (i.e., depending upon the strides). After the matrix multiplication is performed, we add the results. This sum then goes into the feature map. A single convolution map is not enough to extract features of the tomato leaf. Multiple convolutions using different filters are applied on image to get different feature maps. As a result, we perform multiple convolutions on the input image each using different filters and each resulting in unique feature maps. All the feature maps are then stacked together, and the result obtained is the final output of the convolution layer [20].

Nonlinearity. A powerful neural network is constructed using nonlinearity. By using nonlinearity, the network achieves its true potential. The results of the convolution layer are passed through the activation function. ReLU activation function is useful in a manner that it facilitates activation operation. Here, we have used ReLU because the training time of ReLU is several times faster than its counterpart conventional tanh [20].

Pooling. Pooling is the most important layer after CNN. Pooling downsamples the feature maps by keeping the important information. Pooling downsamples each

Table 2 Dataset used for training and testing

Source dataset	Early blight non-segmented	Early blight segmented	Late blight non-segmented	Late blight segmented	Healthy non-segmented	Healthy segmented	Total
Plant village dataset [1]	1000	1000	1909	1909	1591	1591	9000
Internet dataset [18]	–	1059	590	2057	592	–	4298
Real-world dataset (Tansu Farm)	60	60	65	65	–	–	250

feature map independently. Downsampling of feature maps is done by reducing the dimensionality. This results in reducing training time and limiting the effect of overfitting problem by reducing the number of parameters. This system uses max pooling. Max pooling takes the maximum value from the feature map (depending upon the strides) in the window [20].

Fully Connected Layer. Last layer in CNN architecture is fully connected layer. Significance of fully connected layer is that CNN does not know what a noise is, and it considers it as a feature. The job of fully connected layer is to get understanding/learning of features produced by convolution and pooling layers [20].

- *Training*

The CNN network is then trained on 120 epochs using back-propagation technique.

- *Dropout*

To reduce overfitting issue, dropout layers are introduced. It results in 2% accuracy increment. Dropout layer is implemented in order to address the overfitting problem in the training data [20].

Classification Model. Here, the classification model is used to compute the score of the classes using softmax activation function. The fully connected layers supply a connection to all the feature maps generated from the previous layers [13, 2]. The proposed classification model does a job of giving accurate detection of two destructive fungal diseases. Thus, the proposed system can serve as a decision support tool for farmers for the detection of diseases in tomato plant.

As shown in Fig. 3, CNN has three parts: 1. convolution, 2. pooling and 3. fully connected layers [11]. Dataset is then applied to the sequential CNN model that consists of 26 CNN layers. Compared to traditional feature extraction, CNN convolution and pooling layers do the functioning of feature extraction. Proposed model is composed of 6 convolution layers ($32, 3 \times 3 \times 3$) and 3 pooling layers (with a pool size of 2×2). Additionally, three dropout layers are added to provide accuracy boost to our network. Convolution is used to extract features from an image. Convolution does the task of extracting features from an input image, and pooling layer aids in dimensionality reduction [21]. The activation function that the proposed system uses is ReLU [20] and then softmax. Softmax acts as a classifier at the end of the neural

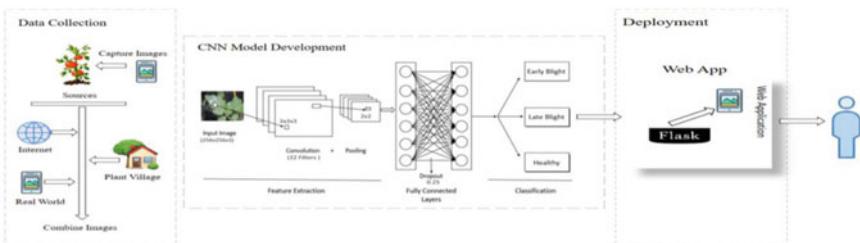


Fig. 3 Detailed block diagram of proposed system

network. Softmax activation function is used in the end for fully connected layers to produce final output classes.

4 Results and Discussion

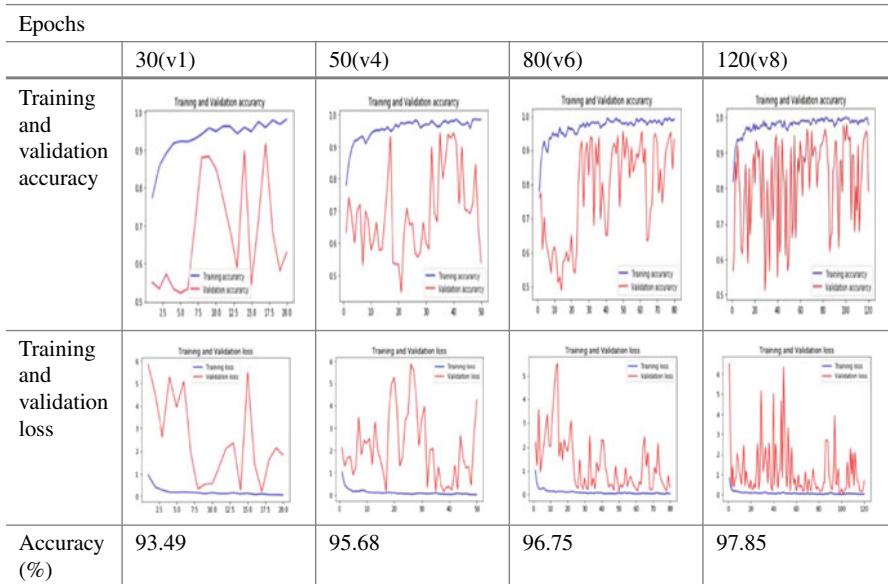
Trained models are tested on the validation set by using GPU provided by Kaggle. Results constitute two categories of dataset as discussed earlier. Therefore, the overall accuracy we obtained on our dataset varied from 86.66 to 97.25%. Table 4 shows the accuracy and validation loss across all our experimental dispositions. The setup consists of different configurations made to run for a total of 120 epochs each. Overfitting issue has been dealt by considering a worst train-test ratio, and 20% data is trained for 80% testing. For rest 80% of the data, the model achieves an overall accuracy of 97.25%. Two categories help to analyze variations in results while keeping other configurations same. Model performs best in case of the color image version of the dataset. The table (Table 3) shown below compares our model based on varying parameters, versions and varying test set-to-train set ratio:

1. Accuracy: The amount of correct classifications/the total amount of classifications.
2. Validation accuracy: The accuracy of a model on unseen data.
3. Validation loss: The value of cost function for your cross-validation data.

All experiments are run for a different number of epochs, where one epoch is nothing but a degree to know number of iterations required to complete one full cycle of the entire dataset. It is worth to note that 120 epochs are a better choice for convergence in all carried out experiments as shown in Table 4 for all variations. The table also presents the train and test accuracy at various epochs as shown by the graphs. It also describes the overall accuracy when the model is implemented at different epochs.

Table 3 Accuracy, validation accuracy and validation loss for the corresponding experimental configuration

Trained data (%)	Test data (%)	Accuracy	Color dataset		Accuracy	Segmented dataset	
			Validation accuracy	Validation loss		Validation accuracy	Validation loss
80	20	0.9725	0.7756	0.7618	0.9287	0.7818	0.1837
60	40	0.9597	0.7068	0.1109	0.9061	0.7183	0.2372
50	50	0.9644	0.7006	0.1011	0.8950	0.7448	0.2640
40	60	0.9648	0.6284	0.9886	0.8828	0.7120	0.2867
20	80	0.9640	0.6989	0.1095	0.9134	0.6636	0.2578

Table 4 Results for different variations

5 Conclusion and Future Work

Agriculture suffers from various problems; plant diseases contribute as the major devastating factor. In this paper, we have proposed a system that focusses on disorder detection of a tomato plant. The proposed system classifies tomato plant diseases such as early blight and late blight using convolution neural network; CNN trains the system in order to classify the diseases of a tomato plant.

It is evident from Table 5 that the proposed system has better accuracy with varying train-to-test ratio. Our system achieves an overall accuracy of **97.25%** with train-to-test ratio of 80:20. Thus, the proposed system can act as tool to help farmers in the early detection of diseases in tomato plant. We intend to extend this proposed model for other crops and more diseases. We would also like to target the severity estimation of the identified diseases, as it will be helpful for the farmers in deciding how to stop the disease.

Table 5 Final model performance with various epochs

Train data (%)	Test data (%)	Proposed system
80	20	0.9725
60	40	0.9597
50	50	0.9644
40	60	0.9648
20	80	0.9640

Acknowledgements We would like to show our gratitude to Mr. Swapnil Dekhane, Research Officer at Tansa Farm, Bhiwandi, for assisting in data collection process.

References

1. Mohanty SP, Hughes DP, Salathe M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7. Article 1419
2. Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng IEEE* 22(10):1345e1359
3. Cruz A, Luvisi A, De Bellis L, Ampatzidis Y (2017) X-FIDO: an effective application for detecting olive quick decline syndrome with deep learning and data fusion. *Front Plant Sci* 8
4. Deep Learning—Part 4: Convolutional Neural Network. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>. Last accessed 2017/11/08
5. The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3). <https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>. Last accessed 2019/11/25
6. Home. In: Food and Agriculture Organization of the United Nations. <http://www.fao.org/home/en/>. Accessed 25 Nov 2019
7. Jacobs IS, Bean CP (1963) Fine particles, thin films, and exchange anisotropy: (effects of finite dimensions and interfaces on the basic properties of ferromagnets). Research Information Section. The Knolls, Schenectady, NY
8. Srivastava S, Boyat S, Sadistap S (2014) A novel vision sensing system for tomato quality detection. *Int J Food Sci* 2014:1–11
9. Sabrol H, Kumar S (2016) Fuzzy and neural network based tomato plant disease classification using natural outdoor images. *Indian J Sci Technol* 9(44)
10. Fuentes A et al (2017) A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* (Basel, Switzerland)
11. Amara J, Bouaziz B, Albergawry A (2017) A deep learning based approach for banana leaf diseases classification. In: Lecture Notes in Informatics (LNI). Gesellschaft für Informatik, Bonn, Germany, pp 79e88
12. Brahim M, Boukhalfa K, Moussaoui A (2017) Deep learning for tomato diseases: classification and symptoms visualization. *Appl Artif Intell J*
13. Oppenheim D, Shani G (2017) Potato disease classification using convolution neural networks. Advances in animal biosciences: precision agriculture. In: 11th European conference on precision agriculture (ECPA 2017), John McIntyre Centre, Edinburgh, UK
14. Liu B, Zhang Y, He D, Li Y (2017) Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* 10:11
15. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145(2018):311–318
16. Lu Y, Yi S, Zeng N, Liu Y, Zhang Y (2017) Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267:378–384
17. Brosnan T, Sun D-W (2004) Improving quality inspection of food products by computer vision—a review. *J Food Eng* 61:3–16. [https://doi.org/10.1016/s0260-8774\(03\)00183-3](https://doi.org/10.1016/s0260-8774(03)00183-3)
18. TM (2019) dataset.zip, Google Docs. <https://drive.google.com/file/d/1DVy0LyUUfJciyo7BUFm1sHKSrdTVJgfF/view>. Last accessed 2019/03/10
19. Validation loss increases while validation accuracy is still improving. <https://github.com/keras-team/keras/issues/3755>. Last accessed 2019/11/21

20. Aydogdu MF (2017) Comparison of three different CNN architectures for age classification. In: 2017 IEEE 11th international conference on semantic computing (ICSC)
21. Durmus H, Gunes EO, Kirci M (2017) Disease detection on the leaves of the tomato plants by using deep learning. In: 6th international conference on agro-geoinformatics. <https://doi.org/10.1109/agro-geoinformatics.2017.8047016>

Chapter 20

Improving Security of IoT Networks Using Machine Learning-Based Intrusion Detection System



Smita Sanjay Ambarkar and Narendra M. Shekokar

1 Introduction

IoT is a ubiquitous and revolutionary technology; this contemporary technique has the potential to contrive the application to become more fast, interactive, accessible, and reliable. Gradually, IoT apprehends the whole business world and provides contentment to end user. However, the challenges such as constrained environment, abandoned deployment and heterogeneous network architecture make the implementation of IoT network onerous. IoT applications will be propitious if devices in the IoT network communicate securely, i.e., security measures incorporated while developing the IoT system. IoT networks shown in Fig. 1 consist of interconnected sensor nodes (devices) that use RPL as routing protocol to communicate with 6LoWPAN border router (6LBR). 6LBR is acted as a gateway between sensor nodes and the Internet or more precisely cloud. Nodes in IoT networks are not having any predefined topology; RPL is responsible to form the node topology. RPL creates destination-oriented directed acyclic graph (DODAG) [1] where 6LBR is a root node having a unique IPv6 address. All the data routed through 6LBR; any attack on a border router may result in the collapse of entire IoT networks. Along with the rapid growth of IoT applications, hasty increase in attacks is also observed [2, 3]. Hence to prominently secure the IoT network, it is obligatory to identify the attacks and prevent them too. IDS provides a pertinent solution to this detection and prevention mechanism. IDS monitors the network traffic and reports the network administrator if any malicious

S. S. Ambarkar (✉) · N. M. Shekokar
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: smita.ambarkar27@gmail.com

N. M. Shekokar
e-mail: narendra.shekokar@djsce.ac.in

S. S. Ambarkar
LTCOE, Navi Mumbai, India

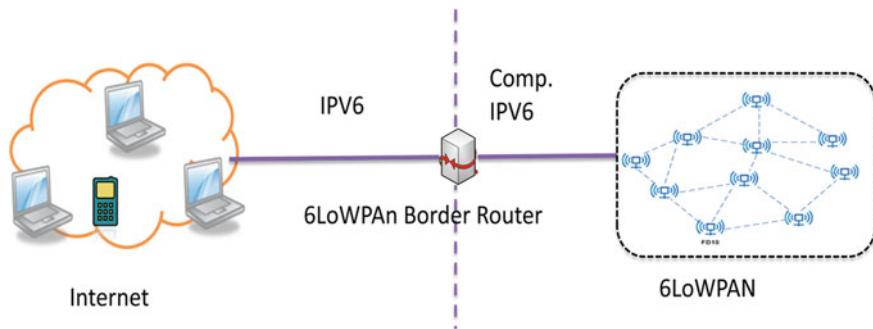


Fig. 1 IoT network

or suspicious activity observed. This is the main concern of this paper. IDS deployed on any host is termed as host-based IDS [4] and on the network called network-based IDS [5], respectively. Similarly, IDS can be located centrally on one center node or distributed on various nodes. IDS is adept in detecting internal as well as external intrusion, and it mainly categorizes into the following three types [6].

Signature-based: It compares the performance and behavior of node with predefined rules of attack also called a signature [7]. If the attack signature is mismatched, network administrator will get an alert. The drawback of the system is that new attacks are difficult to detect, and the false-positive ratio is too high.

Anomaly-based: The normal behavior of the system or host is observed, and if any deviation of the system from normal behavior occurs, it will be reported as an anomaly. The strength of these IDS is that it can identify the novel attacks. It also suffers from the false positive. Machine learning techniques are used effectively to implement these IDS to invent a zero-day attack.

Hybrid: Hybrid technique is a combination of signature and anomaly-based IDS, which can effectively combine the benefits of both the approaches. Hence, Hybrid-IDS with distributed deployment is more apt for IoT applications.

The paper mainly discusses the following contributions of our research work.

1. IoT applications generate huge data. Processing and analysis of this huge data is a big challenge. Machine learning algorithms perform the analysis of IoT data. This paper helps to understand which machine learning algorithm is more pertinent for IoT data. This research work finds IoT application, data characteristics, and properties of machine learning algorithms are mainly three driving sources to choose an appropriate algorithm.
2. Every attack bears different characteristics and behavior; hence, the paper discusses various RPL routing attacks.
3. The existing literature for intrusion detection is analyzed in detail; the emphasis is given on discussion of pivotal characteristics, detection method, a technique used for feature selection, the capability of attack detection.
4. Future directions of machine learning provided for further research in the intrusion detection mechanism.

The rest of this paper organized as follows. Section 2 reviewed the related articles. Machine learning algorithms used for the implementation of IDS are described in Sect. 3. The performance of machine learning algorithms for particular routing attacks is discussed in Sect. 4 and Sect. 5 gives recommendations for future scope. Finally the conclusion is presented in Sect. 6.

2 Related Work

With the escalation of IoT networks, inflation of data generation leads to apprehension about the security. IDS is a promising solution for mitigating security threats and attacks. Processing IoT data and extracting the features from this stream IoT data for implementation of IDS is a NP-hard problem [8]. There are various contributions for implementing IDS; this paper performs systematic research on the above-said problem. The taxonomy of research is given in Fig. 2. At first, the research covers the study of the conventional data mining approach, which will help to find the solution of feature extraction. The data mining models are proposed in [9] which are used for processing IoT stream data (i) a multilayer model (ii) a distributed data mining model (iii) a grid-based data mining (iv) data mining model for IoT from a multi-technology integration perspective. These models describe the framework of future Internet model. Data mining technologies like clustering, classification, frequent pattern mining from the perspective of IoT infrastructure and services are

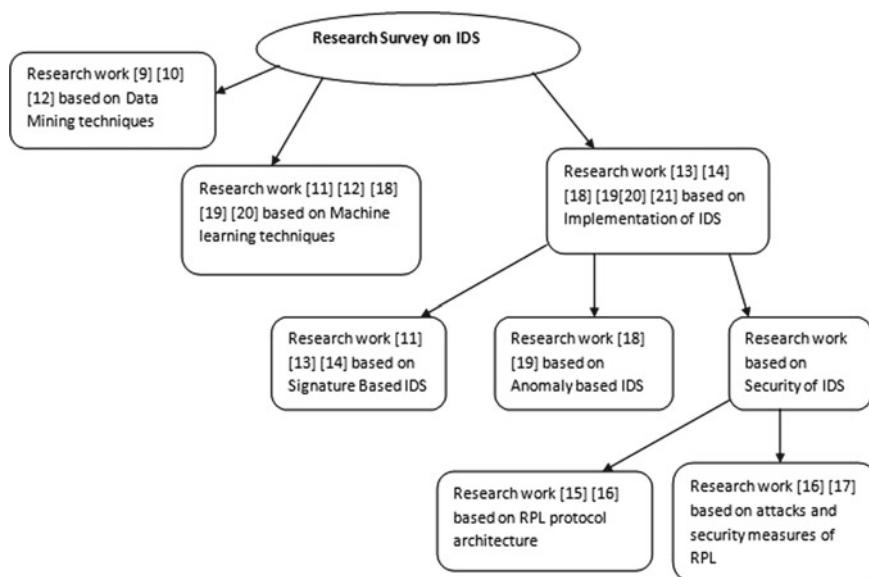


Fig. 2 Research survey on IDS for IoT network

reviewed in [10]. Further, the diverse research survey is performed for implementing IDS for IoT. The survey covers the analysis of signature-based and anomaly-based IDS. The comparative analysis of machine learning and data mining approaches to improve the security of the Internet by using signature and anomaly detection is discussed in [11].

In spite of limitations of signature-based IDS as discussed in Sect. 1, it achieved a huge commercial success. Supervised machine learning algorithms are used to implement signature-based IDS in an effective way. Three categories of machine learning algorithms naive Bayes, J48 also called C4.5, and random forest are surveyed in [12] along with a detail description of a single classifier. The proposal states in a supervised learning classifier required labeled data for training while unsupervised learning based on hypothesis and the labeled dataset is not required for training the classifier. Instances can be labeled by experts in reinforcement learning.

SVELTE proposed by [13] is a first lightweight IDS which was deployed on border router which can detect sinkhole attack with 90% true-positive rate, but this implementation is failed to detect denial-of-service attack as in SVELTE attack, information is transmitted from network to IDS agent; hence once network fails, it is hard to detect denial-of-service attack. A fog layer-based framework to detect an anomaly-based intrusion is proposed in [14]. This IDS is deployed at a fog layer which is nearest to the sensor network; hence, the traffic at the cloud can be reduced. It uses the random forest and extra tree machine learning algorithm, which is a decision-free family algorithm.

RFC 6550 [15] describes the architecture and security of the RPL protocol. Routing attacks are addressed in [16] as RPL is susceptible to many attacks [15, 16]. RPL rank property notions [16] will help to get details of rank-specific attacks. RPL form destination-oriented directed acyclic graph (DODAG) based on the rank property [16]. RPL protocol may fall prey to increase rank attack where a malicious node may satisfy DIO message and joins the network, thereby increasing the rank of the child node. The exactly reverse process happens at the time of decreased rank attack where malicious node modifies its own rank. Other attacks like a sinkhole, sniffing, identity, DOS, DDOS, and selective forward also hamper the security of RPL protocol. According to authors of [17], every category of attack behaves differently from another category of attacks; hence, unique features and characteristics are selected for detecting a particular category of attack. Authors propose a signature and anomaly detection model using c4.5 and one-class SVM algorithms, respectively. IDS-based fuzzy SVM is proposed by authors [18] to improve the accuracy of the classification engine; authors compare the accuracy with multi-class SVM. In another research [19] authors developed SVM with a least square method, which removes insignificant features and improves the detection accuracy. From comprehensive research, backpropagation artificial neural network (BP-ANN) [18], naive Bayes classifier, decision tree (DT) C4.5 [19], multi-class support vector machine (SVM) [20], and random forest classification can be effectively used to implement IDS system.

3 Machine Learning Techniques for IDS Implementation

Through extensive research on the intrusion detection system, it observed that the success of the implementation of this system in a real-time operational environment is very less. Machine learning provides strong support for the implementation of IDS if it applies using care. Machine learning requires understanding the context in which it operates the semantics of the detection process, dataset used, and attacks propagated. Machine learning algorithms are experts in finding similarities than discovering the identities, which are not belonging to that system. Therefore, it is a challenging task to apply machine learning algorithms for anomaly detection techniques [21].

This section presents the detailed working of machine learning algorithms along with their characteristics and features.

Every machine learning algorithm has two aspects of the working, one is training and another is testing as shown in Fig. 3. Sample data from the input dataset is trained using a statistical method to learn the behavior of the network; then, the data is tested to find the abnormalities of the system.

Support vector machine (SVM):

SVM is a supervised learning algorithm. It needs labeled data, and it effectively classifies multidimensional data with maximum margin using the hyperplane. SVM is categorized as linear and nonlinear. Linear SVM uses linear algebra to form linear kernel. Suppose x be the input and y_i the support vectors can be calculated using Eq. 1,

$$f(x) = b(0) + \sum_{i=0}^k y_i. \quad (1)$$

Nonlinear SVM is used when data is not linearly separable. One-class and multi-class SVM is available for implementing signature and anomaly-based detection. Multi-class SVM is a supervised learning algorithm, while one class is an unsupervised learning algorithm. Reference [20] uses one-class SVM for anomaly detection. SVM is very effective for multidimensional IoT data classification, but as it demands

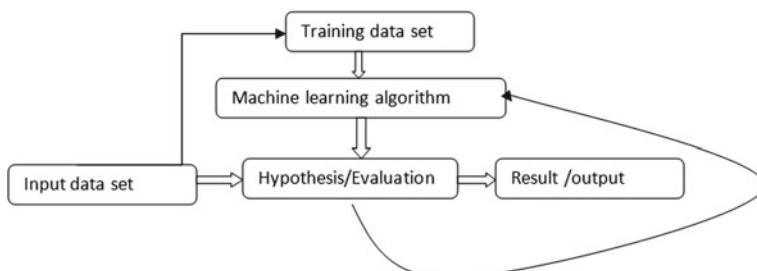


Fig. 3 Basics of machine learning algorithm

high computational resources, lightweight implementation of SVM is required for resource constraint IoT network.

The working principle of SVM includes the kernel selection. Various kernels are proposed in the literature like linear function, radial basis function (RBF), sigmoid function, and polynomial function. These specified kernels are treating all features equally. Instead, not all features are equally important, considering them equally for applying classification may hamper the accuracy of the algorithm. This conventional working of SVM is not suitable for IoT applications as IoT uses heterogeneous and dynamic network architectures. Therefore, for implementing lightweight SVM algorithm, different weights for different features are considered. If the feature does not appear at all, consider weight 0 and if any feature appears for more number of times, consider it for higher weight. Hence, Eq. 1 is modified by considering weights as $k(W_{xi}, W_x)$ shown in Eq. 2.

$$f(x) = b(0) + \sum_{i=0}^k k(W_{xi}, W_x) y_i \quad (2)$$

Naive Bayes Classifier:

Naive Bayes classifier gets its foundation from the Bayesian network. It contains the family of classifiers based on probabilistic estimation as shown in Eq. 3, where x is the feature variable and Y is a class variable.

$$P(y|X) = P(X|y)P(y)/P(X) \quad (3)$$

The attribute values are independent of each other; hence, it is called as a naive Bayes classifier. Naïve Bayes classifier is used when there is a discrete output value of the target function. Three naïve Bayes algorithms are available. (i) Gaussian naïve Bayes, it uses the sample Gaussian distribution; hence, data values can be continuous. (ii) Bernoulli naïve Bayes is used for the predictor which can take only two values; for example, alphabet is there in word or not, and (iii) multinomial naïve Bayes is the same as Bernoulli naïve Bayes; only difference is that it can be used to find the frequency of occurrences. It is a simpler classifier with the belief that features are independent of each other, but this assumption will not be true for IoT dataset; hence naïve Bayes classifier fails to detect different types of attacks.

Decision Tree:

The decision tree is a machine learning algorithm that uses a tree-like model to arrive at decision. If-else rules are used to represent the tree. The decision tree performs well for discrete as well as a continuous value. Internal node, edge, and leaf are the basic parts of the decision tree. The root node can be the best predictor/classifier by using greedy approach; ID3, C4.5, CART, and LMT tree are some of the decision tree algorithms. A decision tree is strapping against an error-prone dataset, and it performs well if some attributes are missing from the dataset. A decision tree can

solve the classification problem well, if attribute and value pairs are given and if the target functions have a discrete output value.

Random Forests:

Random forest algorithm is a classification and regression algorithm and trains a large number of decision trees. The final predicted decision is depending on the decision of individual trees. The random forest algorithm performs better with a low correlation value between individual trees. Predictions that are more accurate are produced from uncorrelated models. Authors of [22] specified that for intrusion detection, random forest performs with better accuracy with the following expedient consideration

- (i) The feature set must contain some feature by using which the prediction model will perform better than random guessing.
- (ii) There must be low co-relations between the prediction states by individual trees.

ANN:

ANN is an artificial neural network motivated by the function of the nervous system. It consists of an input layer, a hidden layer, and output layer. The input layer feeds with the input dataset, the hidden layer consists of processing elements, and then, output is generated from the output layer. If the output is not matched with input, the error is calculated and fed into the input layer again which is called backpropagation. Multilayer perceptron neural network is trained by using a backpropagation algorithm. The gradient descent method is used to adjust the weights between input and hidden layer until the predefined threshold value is not achieved. Fast network convergence and unsupervised learning are the features used to achieve IDS implementation. IDS implementation requires optimized data; this data representation is a challenging task in ANN.

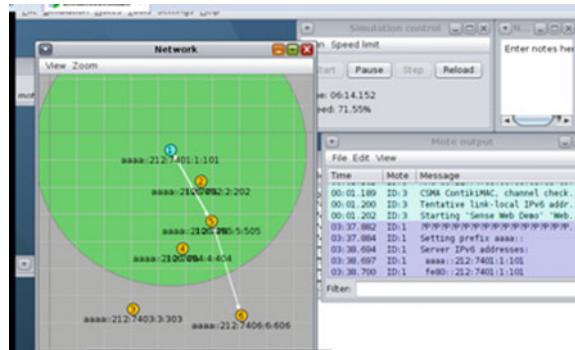
4 Performance of Machine Learning Algorithms Against Routing RPL Protocol Attacks

IoT is an IPV6 network with 6LoWPAN support. Security providers and attackers both use the advanced features of the IPV6 network. Figure 4 shows the implementation of the 6LoWPAN network where the border router is acting as a sink node, and in the IoT network, it is always available to forward the data packets to the Internet. However, this border router is prone to the following attacks unless and until the security measures are not deployed on it.

Selective Forwarding Attack:

In this attack, the malicious node selectively forwards the packet and disturbs the routing paths. This is used to perform a denial-of-service attack (DOS) when an attacker will forward only RPL control messages, and hence, the rest of the other messages

Fig. 4 Implementation of the border router



is dropped. To counter this attack, the nodes in 6LoWPAN networks should choose the paths dynamically. IDS uses source routing of RPL to verify path availability in DODAG. Signature-based IDS is implemented by authors of [23] they used SVM and authors of [24] uses decision tree and naïve Bayes to detect DOS and selective forward attack. DOS is detected by naive Bayes with a detection rate of 96.65 and 97.24% by a decision tree, whereas the implementation of SVM achieves a detection rate of 91.6%.

Hello Flood Attacks:

Destination acyclic graph information object (DIO) is used to launch hello flood attack. DIO messages are used to advertise DODAGs information to the new node. DIO message in the form of hello is the first message sent from the attacker node to introduce itself as new or neighbor. Flooding the network with hello messages can lead to user to root (U2R) attack [25]. Authors in [26] combined the three soft computing techniques SVM, ANN, and MARS. MARS is multivariate adaptive regression splines which is a regression technique based on the novel mathematical process for function approximation and curve drawing. It is used to find the relationship between dependent and independent variables $x = f(y)$. This method detects the attack with a 76% detection rate, but it is not acceptable.

Wormhole Attacks:

This is root to local R2L [25] attack. In 6LoWPAN networks, when two nodes connected by out-of-band connection to transfer critical messages at a faster rate than a normal connection called wormhole. When the attacker created this wormhole, it will transfer malicious packet in the network at a very faster rate. This attack is more dangerous when combined with the sinkhole attack. Authors of [26] claim 100% detection of remote to local (R2L) attacks by combining SVM and MARS with ANN, but the authors did not specify the implementation details. It observed that multiple classifiers will give a better result for detecting the user to root (U2R) and R2L attacks.

Table 1 An analysis of IDS techniques

Research paper	ML algorithms	Dataset used	Type of IDS and dataset used	Attacks	Detection rate% (DT)
[17]	SVM, decision tree C24.5	NSL_KDD dataset	Hybrid (it uses signature at the first stage and anomaly at the second stage)	DoS, R2L, clone ID	99.9
[23]	SVM	KDD99 dataset	Signature-based	DoS	91.6
[24]	Decision tree (DT) and naïve Bayes	KDD99 dataset	Signature-based	DoS	97.24 (decision tree) 96.65 (naïve Bayes)
[26]	SVM, MARS with ANN	KDD99 dataset	Not specified	DoS, R2L, U2R	99.9, 98.3, 76
[13]	6Mapper	Not specified	Signature-based	Sinkhole attack	90
[27]	Dynamic clustering and naïve Bayes	KDD	Signature-based	Sinkhole attack	92%

Sybil Attack and Clone ID Attack:

“Single node with multiple identities” is the Sybil attack. The malicious nodes have various logical identifications at the same time at multiple places. In a similar line, clone ID attack will take place when the attacker copies the valid identity of node onto another malicious node; both are R2L attacks. The integration of decision tree C4.5 and SVM [17] implements the two-stage hybrid classification technique. The first stage is anomaly detection, and the second stage is misuse detection. The input is given to the first stage that uses the SVM algorithm; then, the normal traffic is separated from attack traffic. In the second stage, only attack traffic is given as input and using ANN, attacks like R2L, DoS are detected with the detection rate of 99.9% on the NSL_KDD dataset (Table 1).

5 Recommendations for Future

In an IoT network, cloud plays a vital role. Border router routes the sensor data toward the cloud, and then, the data is available for the application. IDS needs a classifier to classify the attacks and abnormal behavior of the network. The implementation

of the IDS classifier requires training. Therefore, the cloud data is used as a sample dataset to train the classifier of IDS. IDS then deployed on the border router of the IoT networks. Undoubtedly, IDS protects IoT network, but the cloud security is at risk as the attacked packets may get forwarded to the cloud, and also, cloud space will rapidly get exhausted with the mechanism of data storage. Hence, the author of this paper recommends using edge computing or fog computing to deploy the IDS. Edge computing refers to the technique where edge devices connected near to sensor networks are for IDS deployment.

It observed that the IDS implantation of the existing research mostly is done on KDD, DARPA, and KDD99 datasets. KDD and KDD99 are actually not the IoT datasets. Hence, it is required to implement IDS on the real-time IoT dataset. Real-time IoT dataset needs an extensive study of the dataset to select and extract features from it.

IDS needs to be implemented in such a manner that it will automatically update itself to unknown attacks, discover, and detect a zero-day attack.

6 Conclusion

Brisk increase in intrusion is terrifying for IoT network users. Security is at high risk; hence, there is a need to come up with a solution that could defeat not only the existing attacks but also the new signature of it. Although researchers have worked extensively, still there is a long way to go for detecting every intrusion in IoT network. Our research is a step to contribute toward understanding the approaches, which used for the implementation of the intrusion detection system. Various machine learning algorithms discussed in our paper along with the detection rate. The RPL routing attacks are discussed with their descriptions. The analysis is performed for the machine learning algorithms based on the attack detection ratio. With the best of our knowledge, SVM and decision tree perform better for huge stream IoT data.

References

1. Winter T (2012) RPL: IPv6 routing protocol for low-power and lossy networks. IETF
2. Deogirikar J, Vidhate A (2017) Security attacks in IoT: a survey. In: Proceedings of IEEE international conference I-SMAC (IoT social, mobile, analytics cloud) (I-SMAC), Feb 2017, pp 32–37
3. Tomić I, McCann JA (2017) A survey of potential security issues in existing wireless sensor network protocols. *IEEE Internet Things J* 4(6):1910–1923
4. Torkaman, A, Javadzadeh, G, Bahrololum M (2013) A hybrid intelligent HIDS model using two-layer genetic algorithm and neural network. In: Proceedings of IEEE 5th conference information and knowledge technology (IKT), pp 92–96
5. Klippel MDD, Elovici Y, Dolev S (2008) Optimization of NIDS placement for protection of intercommunicating critical infrastructures. In: Proceedings of IEEE international conference on intelligence and security informatics, Taipei, Taiwan, pp 191–203

6. Pathan A-SK (2014) The state of the art in intrusion prevention and detection. CRC Press, Boca Raton, FL
7. Callegari C, Vaton S, Pagano M (2010) A new statistical method for detecting network anomalies in TCP traffic. *Eur Trans Telecommun* 21(7):575–588
8. Montazeri M, Montazeri M, Naji HR, Faraahi A (2013) A novel memetic feature selection algorithm. The 5th conference on information and knowledge technology, Shiraz, pp 295–300. <https://doi.org/10.1109/ikt.2013.6620082>
9. Bin S, Yuan L, Xiaoyi W (2010) Research on data mining models for the internet of things. In: 2010 international conference on image analysis and signal processing. IEEE, pp 127–132
10. Tsai C-W, Lai C-F, Chiang M-C, Yang LT (2014) Data mining for internet of things: a survey. *IEEE Commun Surv Tutor* 16(1):77–97
11. Ahmad B, Jian W, Anwar Ali Z (2018) Role of machine learning and data mining in internet security: standing state with future directions. *J Comput Netw Commun* 2018:10. <https://doi.org/10.1155/2018/6383145>. Article ID 6383145
12. Haq NF et al (2015) Application of machine learning approaches in intrusion detection system: a survey. *Int J Adv Res Artif Intell* 4(3):9–18
13. Raza S, Wallgren L, Voigt T (2013) SVELTE: real-time intrusion detection in the internet of things. *Ad Hoc Netw* 11(8):2661–2674 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870513001005>
14. Alrashdi I, Alqazzaz I, Aloufi E, Alharthi R, Zohdy M, Ming H (2019) AD-IoT: anomaly detection of IoT cyberattacks in smart city using machine learning. In: 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), Las Vegas, NV, USA, pp 0305–0310. <https://doi.org/10.1109/ccwc.2019.8666450>
15. Vasseur J et al (2011) RPL: the IP routing protocol designed for low power and lossy networks
16. Sahay R, Geethakumari G, Modugu K (2018) Attack graph—based vulnerability assessment of rank property in RPL-6LOWPAN in IoT. In: 2018 IEEE 4th world forum on internet of things (WF-IoT), Singapore, pp 308–313. <https://doi.org/10.1109/wf-iot.2018.8355171>
17. Kim G, Lee S, Kim S (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Exp Syst Appl* 41(4):1690–1700
18. Shaohua T, Hongle D, Naiqi W, Wei Z, Jiangyi S (2010) A cooperative network intrusion detection based on fuzzy SVMs. *J Netw* 5:475–483
19. Amiri F, Mohammad RY, Caro L, Azadeh S, Nasser Y (2011) Mutual information—based feature selection for intrusion detection system. *J Netw Comput Appl* 34:1184–1199
20. Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC (2000) Support vector method for novelty detection. In: Proceedings of advances in neural information processing systems, Denver, CO, USA, pp 582–588
21. Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE symposium on security and privacy, Berkeley/Oakland, CA, pp 305–316. <https://doi.org/10.1109/SP.2010.25>
22. Almseidin M, Alzubi M, Kovacs S, Alkasassbeh M (2017) Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th international symposium on intelligent systems and informatics (SISY), Subotica, pp 000277–000282. <https://doi.org/10.1109/sisy.2017.8080566>
23. Kim DS, Park JS (2003) Network-based intrusion detection with support vector machines. In: Kahng HK (ed) Information networking. ICOIN 2003(LNCS 2662). Springer, Heidelberg, Germany, pp 747–756
24. Amor NB, Benferhat S, Elouedi Z (2004) Naive Bayes versus decision trees in intrusion detection systems. In: Proceedings of the symposium on applied computing, Nicosia, Cyprus, 2004, pp 420–424
25. El Mourabit Y, Toumanari A, Bouriden A, Moussaid NE (2015) Intrusion detection techniques in wireless sensor network using data mining algorithms: comparative evaluation based on attacks detection. *Int J Adv Comput Sci Appl (IJACSA)*, 6(9). <http://dx.doi.org/10.14569/IJACSA.2015.060922>

26. Mukkamala S, Sung AH, Abraham A (2005) Intrusion detection using an ensemble of intelligent paradigms. *J Netw Comput Appl* 28(2):167–182
27. Cervantes C, Poplade D, Nogueira M, Santos A (2015) Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for internet of things. In: 2015 IFIP/IEEE international symposium on integrated network management (IM), Ottawa, ON, pp 606–611. <https://doi.org/10.1109/inm.2015.7140344>

Chapter 21

Sapling Health Monitoring System



Shubham Mishra, Nishanth Shastry and Tarun Tiwari

1 Introduction

In today's world, due to increase in population demand for food has increased to a large extent. Increasing urbanization is making condition more worse. According to reports, in 1960, 39.7% of land around the world was under agriculture, while in 2016, 39.7% decreased to 35.2% (a decrease of 22.92 million km²). On the other hand, crop failure happens. In most cases, crop failure occurs due to infection in crops. We are struggling to meet current demands, whereas according to United Nations report, till 2050 demand would increase by approximately 70%. Modern problems require modern solutions. We need modern technologies to cope up with the demand by increasing the yield and decrease chances of crop failures. Combination of traditional methods and modern technologies can lead to modernization in agriculture. A rover automated using image processing collects data about moisture, pH of soil time to time and stores it in form of .csv format. Every time data is collected a report of data is sent to user through email. Data stored in .csv format is sent to a cloud-based local system, where a report is generated which shows the condition of crops over a period of time. In case of disease detection, same rover can click the leaf snap and send it to the cloud-based local system. If crops are suffering from any disease, the shape and pattern of leaves change depending on the type of disease. So disease can be predicted by analysis of leaf. The local system contains a deep learning-based model which is trained using a dataset containing approximately 55,000 pictures of

S. Mishra · N. Shastry (✉) · T. Tiwari
MVJ College of Engineering, Bangalore, India
e-mail: n18shastry@gmail.com

S. Mishra
e-mail: smishra.shubhammishra@gmail.com

T. Tiwari
e-mail: tarun12191998@gmail.com

different cash crops. The model makes a prediction with 96.1% of prediction. The accuracy of system increases with increase in time.

2 Related Work

Gondchawar [1] proposed an IoT-based smart agriculture. The paper aimed at a system based on IoT and automation, where rover loaded with sensors is used to collect the real-time data. Along with data collection, the rover also performs spraying, weeding, etc. The rover helps in maintaining temperature, humidity and other factors that help in healthy growth of crops. The system provides concepts on automation in agriculture.

Mohanty [2] trained a model that can detect disease in plants based on images of their leaves by applying deep learning algorithms on a public dataset “PlantVillage Dataset.”

Khirade [3] proposed plant disease detection using image processing. The system achieves the aim using concepts of segmentation. After preprocessing of image, segmentation and feature extraction are applied on the image, and then classification and detection are done.

Rajalakshmi [4] proposed a system for monitoring the agriculture field using sensors. The system also displays the data collected of web server in JSON format. Some of the sensors used are soil-moisture sensors, humidity sensor and light sensors. The following system explained topics like data visualization, data collection and data analysis.

Thangadurai [5] came up with an algorithm that can detect disease in plant by cross-validating the cumulative histograms of captured image and infected plant's image to make prediction about the health of plants.

Jhuria [6] trained a model that detects diseases in plants using ANN algorithm.

The system concludes that models based on color and morphology are more accurate than models trained on textures.

Prathibha [7] in his work proposed a system that can give alert during critical situation based on data collected on temperature, humidity and other factors.

All the above models and systems were considered and analyzed while building our system.

3 Proposed Methodology

3.1 Working Cycle of the Device

The first step in the implementation would be to create a virtual map of the particular plantation and marking the plant tags and initiating a new database. Based on the

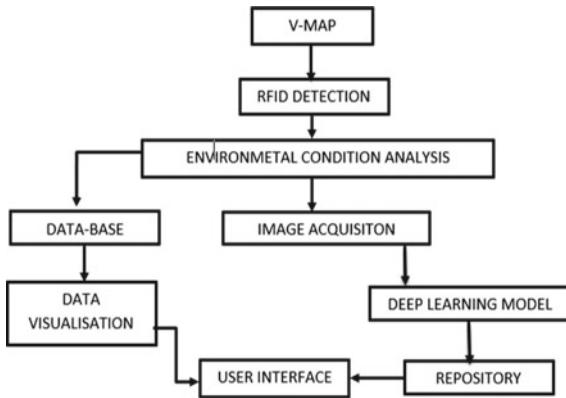


Fig. 1 Block diagram of work cycle

map, the rover scouts the area at regular intervals checking for vital parameters like soil-moisture level, pH levels, light reception, etc. The rover moves around the field and identifies the planted sections and appends the data into the database on the region identified using RFID system. The user is able to view, control and intervene into the system through an interface on smartphone or a personal computer. The rover is equipped with camera which captures picture of each plant and checks for any unusual patterns or symptoms visible and reports the same to the user (Fig. 1).

4 Specifications of the Prototype Device

4.1 Hardware Specification

Arduino Mega: Microcontroller used in rover to control the motion of rover as the wheel and motors of rover are connected to rover.

Raspberry Pi 3b: Microprocessor that collects the data and stores it and sends it to a local system. Raspberry Pi and system are connected over cloud.

Pi-Camera: Compatible camera for Raspberry Pi that takes the video and passes it on to processor for processing.

Moisture sensor, pH sensor: Sensors that collects the readings of moisture and pH of soil in real time.

Servo-Motors: These motors are operated through the Arduino control board. It provides an arm like motion for sensor, providing it an easy to get readings.

4.2 Software Specification

Python: A multifunctional language which was used in implementing image processing, model of deep learning, data storing and analysis, creating and plotting graphs.

C++: It is primarily used for configuring the Arduino microcontroller board, to provide logic for rover to work on.

HTML and CSS: Following languages are used in creating a web page to provide a user-friendly way to show the report of crops.

5 Working

The following techniques are used in the beta model of this implementation for testing and development (Fig. 2).

5.1 Data Acquisition

The rover is equipped with range of sensors from soil-moisture sensor to pH level sensor which collects the data from various regions of the plantation. RFID tags are setup to mark different areas or plant types. This process can be made totally autonomous where the system will decide the location for sampling and collect the necessary information and load it on to the database for further processing.

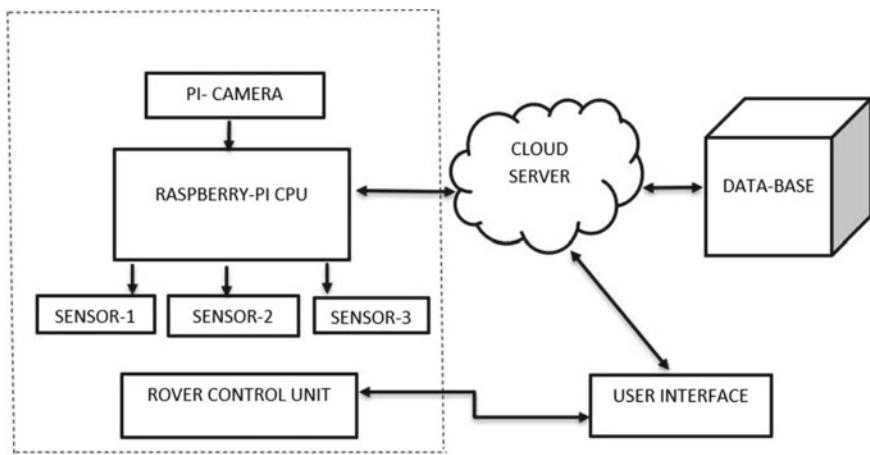


Fig. 2 Block diagram of the system architecture

Alternative to this the user can define the rover some particular locations which need to be checked through his interfacing device.

Once data is collected, it is stored in .csv format using pandas library. Using matplotlib library of python, the data collected is converted in form of bar graph and with the help of smtp library the bar graph sheet is sent to the user. Simultaneously, the .csv file is sent to local system through clouds, where the report is generated in the form of graphs and displayed to users on web page or the user interface device using a mobile application.

5.2 Disease Detection

We propose an AI application using deep learning and transfer learning technique. The “PlantVillage Dataset” [spMohanty] is used for training the model. This dataset contains an open-access repository of images on plant health to enable the development of plant disease diagnostic systems. The dataset contains 54,309 images. The images span 14 crop species: apple, blueberry, cherry, grape, orange, peach, bell pepper, potato, raspberry, soybean, squash, strawberry and tomato. There was 80%–20% split for training and testing, respectively.

5.2.1 Data Augmentation

The term data augmentation refers to methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables.

Image

Data augmentation can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images. Data augmentation such as random rotation, resized crop, random horizontal flip and center crop was applied on the dataset. This is to ensure that the model classifies the images regardless of orientation.

$$\int_{\mathcal{M}(Y_{\text{aug}})=Y_{\text{obs}}} p(Y_{\text{aug}}|\theta) \mu(dY_{\text{aug}}) = p(Y_{\text{obs}}|\theta). \quad (1)$$

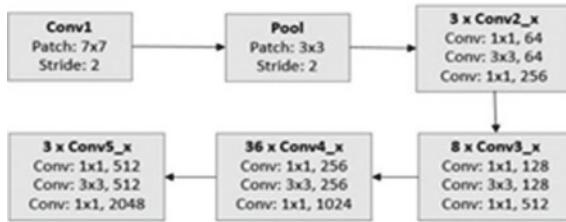


Fig. 3 Block diagram of architecture of Resnet-152

5.2.2 Neural Network Architecture

The proposed idea is to use a pre-trained model to get the image features and build and train a new feed-forward classifier using those features. The pre-trained model that was used is Microsoft's Residual Networks architecture: Resnet-152. This is one of those models used in COCO 2015 competitions, which won the first place in: ImageNet classification, ImageNet detection, ImageNet localization, COCO detection and COCO segmentation (Fig. 3).

The rationale behind using a pre-trained model is because it saves time and this kind of model was trained on a large dataset to solve a problem similar to the one addressed here, i.e., image classification. After installing the pre-trained model for the image classification, the original classifier was removed, and a new classifier was added, to help in identifying plant diseases. Finally, the model was fine-tuned by freezing some parameters.

5.2.3 Adam Optimizer

Adam is an adaptive learning rate optimization algorithm that is been designed specifically for training deep neural networks. It can be looked at as a combination of RMSprop and stochastic gradient descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum.

$$w_t = w_{t-1} - \eta \left(\frac{\hat{m}}{\sqrt{\hat{v}_t} + \epsilon} + \lambda w_{t-1} \right) \quad (2)$$

5.2.4 Hyperparameters

In order to improve the performance of the proposed model, different hyperparameters were tried (such as learning rate, optimizer and batch size) during the training.

Hand-tuning hyperparameters can be expensive since a training job may take many hours, if not days, to finish. This leads to the demand of doing systematic search. The proposed was run with 10 epochs and used Adam optimizer with a learning rate of 0.001.

5.2.5 Training and Validation Accuracy

There was an 80–20% split between training and testing data. The model was tested and predicted the diseases with an accuracy of 0.961 (Figs. 4 and 5).

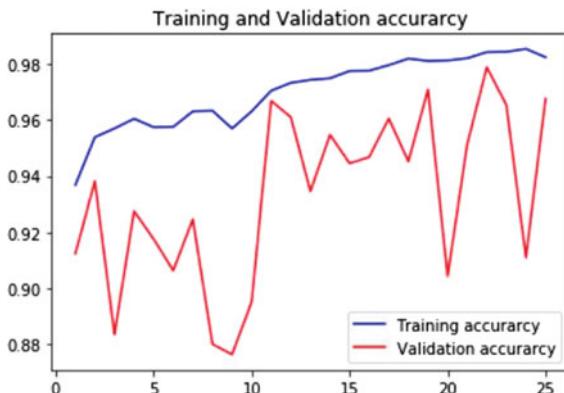


Fig. 4 Training and validation accuracy

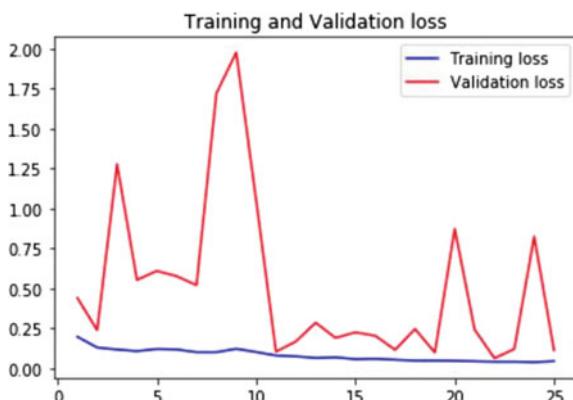


Fig. 5 Training and validation loss

6 Prototype Setup

The prototype is constructed with an aluminum body, and all the electronics are housed inside the body to protect it from environmental hazards. The sensors are mounted on to the exterior of the rover for collection of data. The soil-moisture sensor is attached onto a servo motor for mobility and ease of use. All terrain wheels are used which are powered by brushed DC motor. The whole system is powered by a 12 V on-board battery. The motor system is run using L298N motor drivers operated via an Arduino Mega 2560 microcontroller board (Figs. 6, 7 and 8).

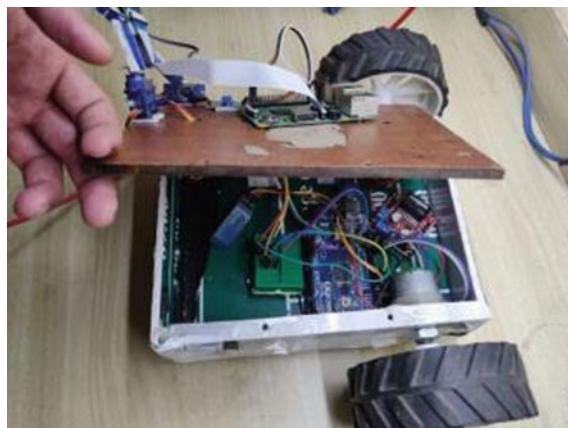


Fig. 6 Inner view

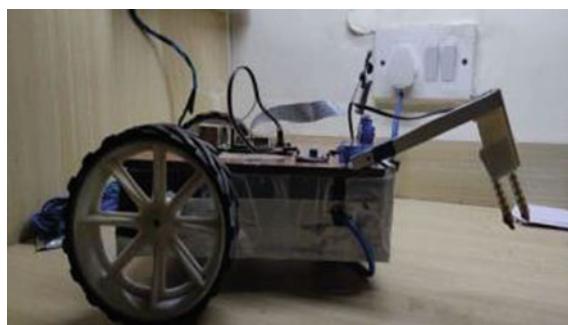


Fig. 7 Side view



Fig. 8 Top view

7 Results

The acquired data is processed on a cloud platform and sent back to the user in a comprehensive visual representation via the user interface (Figs. 9, 10, 11, 12).

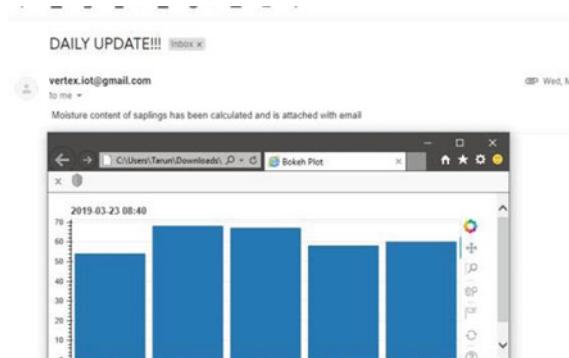


Fig. 9 Report sent through email automatically

A	B	C	D	E
Field Area	19-03-2019 17:27	19-03-2019 17:29	19-03-2019 17:33	19-03-2019 17:37
e501	87	74	87	98
e502	57	68	65	25
e503	68	65	85	65
e504	92	69	69	85
e505	21	20	42	25

Fig. 10 Data stored automatically using python (pandas)

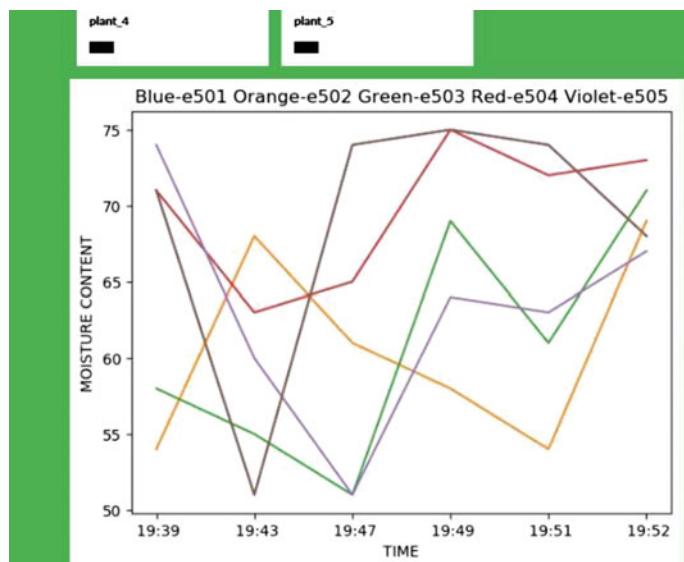


Fig. 11 Web page displaying information about all region at same time

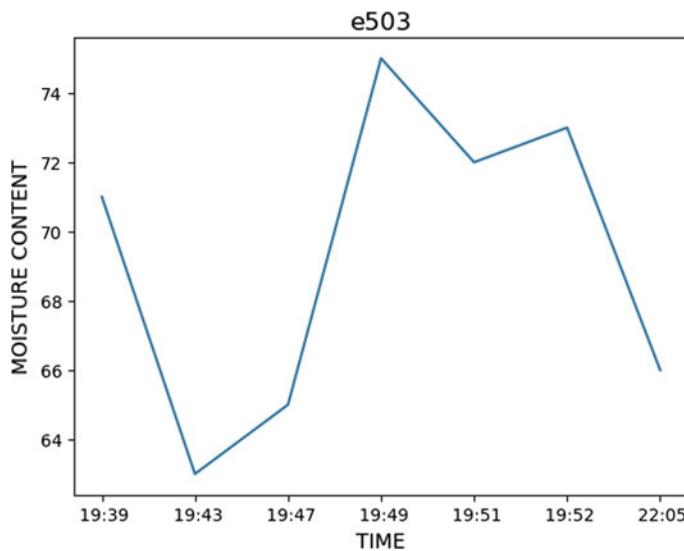


Fig. 12 Graphical representation of data collected from e503 (extracted from webpage)

8 Conclusion and Future Scope

The aim is to make system more accurate in terms of data collection by connecting more sensors to the rover to analyze more factors on which crop growth depends. The data collection will be made secure and reliable by implementing suitable communication protocols with necessary encryption. Since the main target of project is farmers the system is designed to be cheap, user friendly and reliable. A user-friendly web or app interface is to be developed for easy access to the data, manual control and maintenance of the system.

References

1. Gondchawar N, Kawitkar RS (2016) IoT based smart agriculture. *Int J Adv Res Comput Commun Eng* 5
2. Mohanty SP, Hughes DP, Salathe M Using deep learning for image-based plant disease detection
3. Khirade SD, Patil AB (2015) Plant disease detection using image processing. In: 2015 international conference on computing communication control and automation
4. Rajalakshmi P, Mahalakshmi SD (2016) IoT based crop-field monitoring and irrigation automation. In: 10th international conference on intelligent systems and controls, Jan 2016
5. Thangadurai K, Padmavathi K (2014) Computer vision image enhancement for plant leaves disease detection. In: 2014 world congress on computing and communication technologies
6. Jhuria M, Kumar A, Borse R (2013) Image Processing for smart farming: detection of disease and fruit grading. In: Proceedings of the 2013 IEEE second international conference on image information processing (ICIIP-2013)
7. Prathibha SR, Hongal A, Jyothi MP (2017) IOT based monitoring system in smart agriculture. In: 2017 international conference on recent advances in electronics and communication technology (ICRAECT)

Chapter 22

Proposed Infrastructure for Census Enumeration and Internet Voting Application in Digital India with Multichain Blockchain



Vivek Tirodkar and Sonali Patil

1 Introduction

India is a vast democracy with population of ~1.4 billion; maintaining data of 1.4 billion population is a huge task. Digitization of system can help to manage huge data. India is transforming into a digitally empowered society and knowledge economy under Digital India program. But still digitization of census and voting system has not been done yet which can solve problem associated with current system.

Census is carried out every 10 years; current census enumeration process involves data collection on paper, then collecting and entering on computers, segregation of data, and its analysis and data processing make it lengthier and more complicated. By the time the data is published, it becomes out of date and cannot be used for any policy matter. The census organization should be armed with necessary authority to have access to households and canvass the prescribed questionnaires and to expect the people to answer truthfully with assurance of secrecy of the information collected by law [1]. In this method, custody of data needs to be taken care. The answers ascertained at the census can be used only for statistical purposes in which the individual data gets submerged. With huge geographical area and population along with diversity of people living together, it is impossible for census enumerator to know all languages, but digitization of collection process can make it available in different languages. So that census enumeration will become easier. The online process will make it more simple by directly entering data on blockchain node via Internet, so that it can be retrieved from any other node and available for further processing. The cost of purchasing or hiring the equipment is quite high and also

V. Tirodkar (✉) · S. Patil
K. J. Somaiya College of Engineering, Mumbai 400077, India
e-mail: vivek.tirodkar@somaiya.edu

S. Patil
e-mail: sonalipatil@somaiya.edu

requires trained manpower to work on large data again with the electronic equipment [1]. The proposed infrastructure will make it possible to automate data processing from data available on blockchain node. It will reduce the unnecessary cost and manpower and will produce quick result. In India, there are several ways to identify an individual such as Aadhaar, passport, and driving license. With the possibilities of the blockchain technology, all such identities can be consolidated and only one identification can be used for the endless applications [2].

Paper and electronic ballot systems have been adopted by various democratic countries, while Internet-based voting is still in the development stage because of lack of security and cyber-security threats [3, 4]. The USA and Russia explored Internet voting in early 2000s and decided not to use the system due to various security and privacy concerns. Election Commission of India has also expressed its concern about the security issues related to Internet voting systems [3]. India had adopted electronic voting system using Electronic Voting Machine (EVM) in 2003 to avoid several drawbacks of paper ballot-based voting systems. Electronic voting system has many issues like physical security, transportation, and storage and aging problem of EVM. Also, possible hidden loopholes because of proprietary hardware and software, as well as lack of publicly accessible test procedure and certification authority reports, reduce public confidence over EVM [5]. Currently in 2019 Indian election, EVM facilities were available with voter-verified paper audit trail (VVPAT) to improve public confidence over EVM with vote tamper proofing evidence [6, 7]. However, election percentage of voting was less because of voter restriction about voting through their eligible constituency. Alternative method such as e-postal ballot and proxy voting available for voting from other places rather than voting physically from registered(eligible) constituency for service voter working at different constituency and NRI voter, this method has possible drawbacks. Delays in reception of marked votes through e-postal ballot after the date of counting of votes consider the ballot meaningless also nominated person for proxy voting may caste different choice of votes than original voter. The large population of service voters and their families, NRI voters and domestic migrant voters is sufficient enough to alter election results. India also does not support absentee voting for people who are not able to reach or not want to vote in their respective constituency because of migration or other reason. At the last, previous election data and geographical division of India for election are also necessary to understand, to propose, and to check requirement, feasibility of infrastructure. The maximum size of the Lok Sabha as outlined in the Constitution of India is 552; hence, maximum of 550 constituencies can be divided on the basis of their population and two Anglo-Indians are nominated by President [6, 8]. As per 2019 general election, 900 million people were eligible to vote. The ECI deployed a total of 1.74 million voter-verified paper audit trail (VVPAT) units and 3.96 million electronic voting machines (EVM) in 1,035,918 polling stations [9]. From 2014 election, first time “none of the above” option is included and Non-Resident Indians are allowed to vote [9, 10]. For Vidhan Sabha election, India is divided into 4139 constituencies [11].

Considering cost and management issues of current process of census data collection and e-voting system, secure online system must need to be established under

e-Kranti program of India to make it more flexible. Also, it will eliminate geological restrictions secure online system under ekranti program will eliminate geological restriction for voters temporary relocated (i.e. voter need to go back to register constituency for voting). If geological restriction not enforce on voter then they can vote from any voting station present in any constituency from part of India. Client-server model for online voting system with centralized database server cannot be used for voting in India. Because it cannot handle load of number of polling station and population available in India. It will not possible to run voting on secure channel which causes wide open attack surface for hackers. The use of conventional structural database like MySQL has various limitations of handling huge data and also has known issue. Cloud computing technology to store result voting over cloud can be cost-effective solution. But cloud storage and security are in hand of cloud provider questionable to trust and privacy issue. Cloud technology or platform for online voting system cannot be used because of unawareness about cloud infrastructure and security measures taken by them, also cannot control sensitive data storage space, hidden loophole of share storage and forensics difficulty. Sensitive application like voting and census enumeration cannot trade off cost over security and privacy [12].

Security associated with blockchain technology is better than other models like client-server model, cloud-based technology, or involvement of mobile-based services. Multichain blockchain platform is a private blockchain platform, selected because of various advantages over another public blockchains available, also suitable to build infrastructure and application requirement to achieve optimal security [13]. Existing infrastructure based on Internet is not suitable to accommodate blockchain technology limitation. Also, Internet-based voting system problems [3] can be solved by instead of implementation of system over wide open infrastructure, attack surface can be reduced by implementing system over isolated infrastructure. Keeping security and privacy in mind, building a required isolated infrastructure for online system is challenging. Isolated infrastructure is long-term planning process but best solution. Only possibility is that allowing voter to vote from any nearest constituency's voting station instead of eligible constituency's voting station. Because certain authority can take care of security measures at polling station and also can govern the process. Practical infrastructure and application requirement for online voting system and census enumeration process with private multichain blockchain have been addressed in this paper.

2 Blockchain-Based System Implementation Requirement

Considering technical blockchain limitation for application design [14] and blockchain-based voting system concerns [15], only private blockchain is suitable for sensitive application than public or consortium blockchain. Multichain blockchain, a private blockchain, is suitable because of its various advantages over public blockchain, and security can solve blockchain limitation. Integrated management of user permission, permission associated with transaction to provide control for

specific nodes over that transaction, makes it secure. Multichain blockchain implements immutable key-value-based, time series and notarization database with help of concept of stream. The append only database is unstructured type of database where it stores each entry in array forms next entry will get append to previous entries continuously. Append only data based don't have predefine schemas or table to store data. Also, same key mapping to similar data, with help of proper application design, can provide structure data retrieval benefits [13]. Only limitation of bandwidth about blockchain requires precaution in Web application design to reduce unwanted data and isolated infrastructure along with exiting Internet infrastructure for efficient use of bandwidth which is proposed in following topic.

3 Limitation of Current Internet-Based Infrastructure for Census and Online Voting Application with Multichain Blockchain

Consider each multichain node as either each voting station or one node of multichain can used to store data of one constituency/administrative division of state legislative assembly. This node will be used to collect and store census data of people in that constituency. India has around 10.5 lakh voting station, and most of them are temporarily available during election period only. Idea of considering each multichain node as voting station will be impossible. So other option is that consider number of multichain nodes equal to 4689 smallest administrative division which include 550 Lok Sabha and 4139 Vidhan Sabha constituencies in India. All voting station present in one area of constituency will only connect and communicate with blockchain node assign for that constituency node. They cannot connect to any other node because each node requires rpc username and password to connect as per application design. An average 254 polling stations per node will communicate in election period. One possibility can be run 4689 blockchain nodes on existing Internet with each node for one administrative constituency. But for peer-to-peer communication and data sharing of 4689 nodes, it will consume most of existing Internet bandwidth. As shown in Fig. 1, it will require same bandwidth channel through Internet Service Provider (ISP) for communication with every other node, communication from various voting stations, devices, and authorized places within constituency. Wide open Internet-based infrastructure makes it possible to wide open attack surface. It will be impossible to implement blockchain-based online voting and census process over existing Internet.

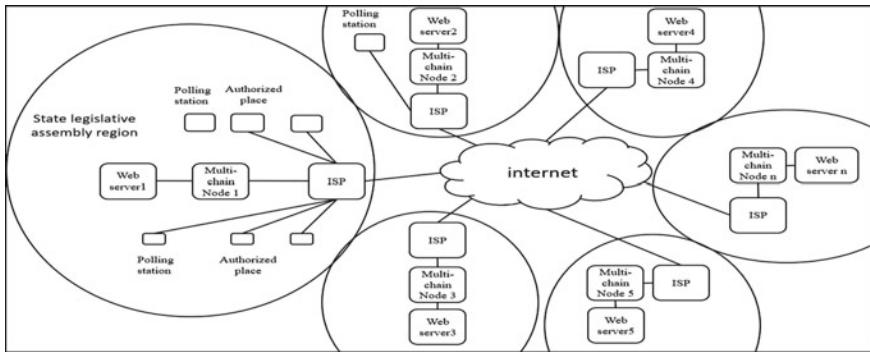


Fig. 1 Blockchain implementation over existing Internet for voting and census in India

4 Proposed Infrastructure for Census and Internet Voting Application with Multichain Blockchain

As shown in Fig. 2, the proposed infrastructure is isolated infrastructure that must build and govern by Indian government, where 4689 fix station one for each constituency including 550 Lok Sabha and 4139 Vidhan Sabha across 29 state and 7 union territories. In India, approximately eight constituencies of Vidhan Sabha cover one Lok Sabha constituency region. All these stations assign for Vidhan Sabha must connect with nearest constituency station via fiber-optic cable to build isolated peer-to-peer infrastructure as well as high availability of communication channel. This

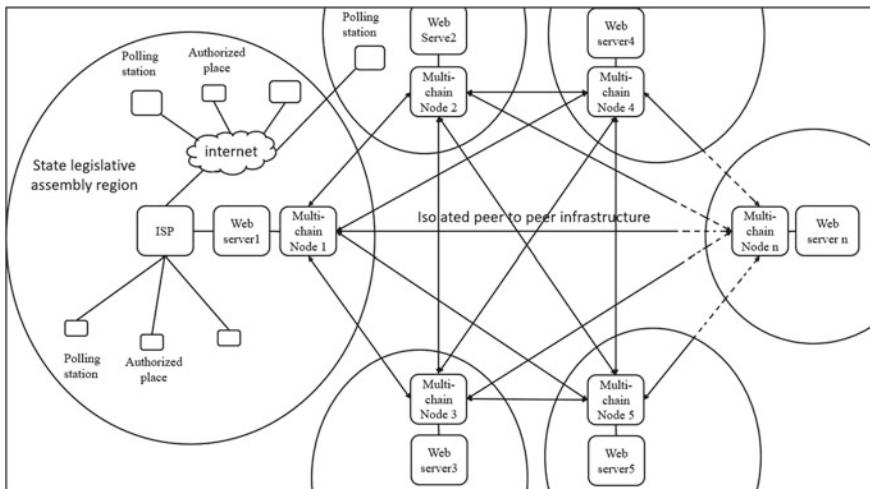


Fig. 2 Proposed infrastructure with multichain blockchain for census enumeration and online voting system

Vidhan Sabha station also must connect to Lok Sabha station where node assigned for Lok Sabha will grant permission of mine and will take care of mining of all Vidhan Sabha node present within that constituency. Every constituency station requires one multichain node for that constituency as well as Web server for connecting over Internet, which also require same number of IP address. Each voting station can access this node with Web application from different voting station or authorized places from India. These authorized places will be used to collect census information or it will be possible to collect data from mobile device provide to authorize official over blockchain node with help of Internet. Each multichain node will collect data and store it on multichain stream. Data collected from each node is replicated on every single node of blockchain. Hence, it will be accessed from anywhere around India and it will be helpful in disaster recovery.

Figure 2 is infrastructure proposed with multichain blockchain, where every blockchain node will communicate with all surrounding constituency's blockchain node in isolated peer-to-peer manner. Data entry and result display can be achieved with Web application running over Web server with help of Internet. Another advantage of this infrastructure such as local ISP can route voting station traffic and traffic of authorized places directly to node belong to that constituency instead of routing over Internet. Hence will ensure privacy of data channel and minimize attack surface. To build this infrastructure, it will require huge effort on constituency level and it is ambitious and long-term process, but once created, definitely reduce overall cost of individual project of India in future.

5 Web Application Development Consideration for Census Enumeration and Online Voting System with Multichain Blockchain

Application design plays an important role in building security and managing huge data. Proposed infrastructure allows authorize official to connect and enter data directly on multichain node global stream created. Data of people living in particular constituency can be enter over each node of multichain assign for that constituency using Web application from authorized places or mobile device connecting to node in that constituency. Different 50 streams will require to store different type of census data and election-related information. As per census, 50 streams are created to store information about Aadhaar, date of registration, PAN number, voter ID, nationality, birth date, voter type, first name, father name, mother name, surname, spouse name, room number, building number, building name, street name, pin code, voting state, voting district, voting subdistrict, voting town, relation with head of department, sex, age at last birth date, marital state, age at marriage, religion, cast, mother tongue, language known, education, current education, disability status, worker state, worker type, worker activity, other worker activity, job requirement, mode of transport, birth-place, last resident place, duration at last, migration cause, children born married

women, children survive, previous year childbirth for recently married women, area cultivated, tenancy of area cultivated, death of people. Data of each person will be mapped with its own unique Aadhaar number as key to store data over stream. That data can be retrieved from any node with key associated with that person. Mapping all data with unique Aadhaar number as key provides data consolidation benefit. Answer of all above data has different characteristics; hence, the Web application must design in such a way that answer associated with particular question must be specific with fix possible values. It will help to reduce unwanted data entering over blockchain as well as fix answer pattern will make it easier to analyze and process the data and that data can also be used for other government operation such as user authentication for online voting application. Some of voting details like voting card number and candidate eligible ward number, eligible constituency detail, must need to collect along with census data gathering and will serve as election data for 10-year period from census. That data will be replicated on each node and can be available on any node. Data is signed by publisher node private key, time stamp, and hash so that once recorded can't be modified later.

Figure 3 shows working flow of census process. Web application must develop for census enumeration in such a way that it will allow to fill data and automate analysis and generate census report with information available on blockchain stream in this way process becomes faster and easier. Multichain stream can be accessed with JSON RPC which requires JSON username and password and port allows on blockchain server node for JSON RPC to store and access data over multichain. Web application design must take care of authentication measures and session where without knowing credentials nobody can access information associated with Web

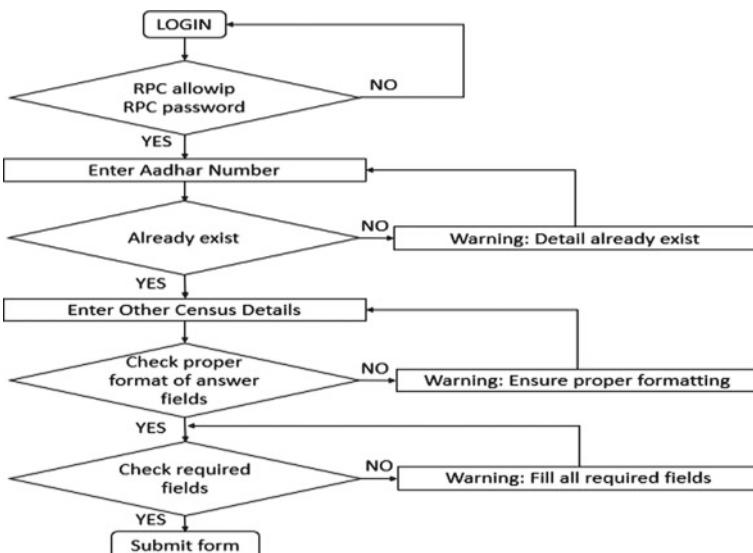


Fig. 3 Flow of Web application for census

site. Also, Web application must design in such a way that it will record the IP of client and check that IP is allowed to access the JSON RPC call will make it more secure. Only permitted IP can access this Web site with proper credential.

Proposed infrastructure allows voter to vote from any nearest voting station belong to any nearest constituency rather than eligible constituency. Voter will verify their identity with data available on constituency node. Since data of each person is mapped with its own unique Aadhaar number, voter requires its Aadhaar number to enter. If voter's age is greater than 18, then only allowed to proceed for further voting. Voter will verify other parameters like birth date, voter ID, and ward number of polling station. After verification from local node constituency, if voter belongs to same constituency, then voter will see list of candidates standing for that constituency or if voter belongs to another constituency, then voter will redirect to the Web server of eligible constituency containing candidate list from other constituency. For voting purpose, voter will use concept of local cryptographic address key pair and asset. Each multichain node can create any number of local public and private key pair. Admin of that node will create address equal to number of candidates standing for each candidate. Admin will create two more addresses; one is for constituency address creating token equal to eligible voter belong to that constituency, and another is for none of the above option. Admin from each constituency node will generate constituency tokens with name defining constituency and election detail. Quantity must be equal to eligible voter in their constituency for election. As voter selects option listed from available candidate to vote, the token is sent from constituency address to candidate address or none of the above address. This process will require one global stream name voting status which store election ID as key and Aadhaar number as data to track election status globally. This stream data is replicated at each node. When candidate enters Aadhaar again for voting, multichain node will check his previous voting status for particular election ID as key; hence, candidate once voted cannot vote again from any voting station and avoid multiple voting, which is a fraud. Also, it will store Aadhaar of candidate over stream under same election ID, which will maintain anonymity of voter. Timestamp of that data entry will useful for analysis of block in which transaction happened to verify detail about token deposited in intended candidate address. Finally, all tokens associated with candidate address will give voting winner and token available in constituency address will provide percentage of voting. Figure 4 shows working flow of Internet voting system.

5.1 Test Case for Census and Voting Application

Web application designed as per security requirement, three multichain nodes are created for analysis purpose and census data of 50 users divided among three constituencies is randomly prepared to store over blockchain. Consider three candidates standing for voting through each constituency. Data is entered over stream through that constituency node as per application flow of census. Application is used to design

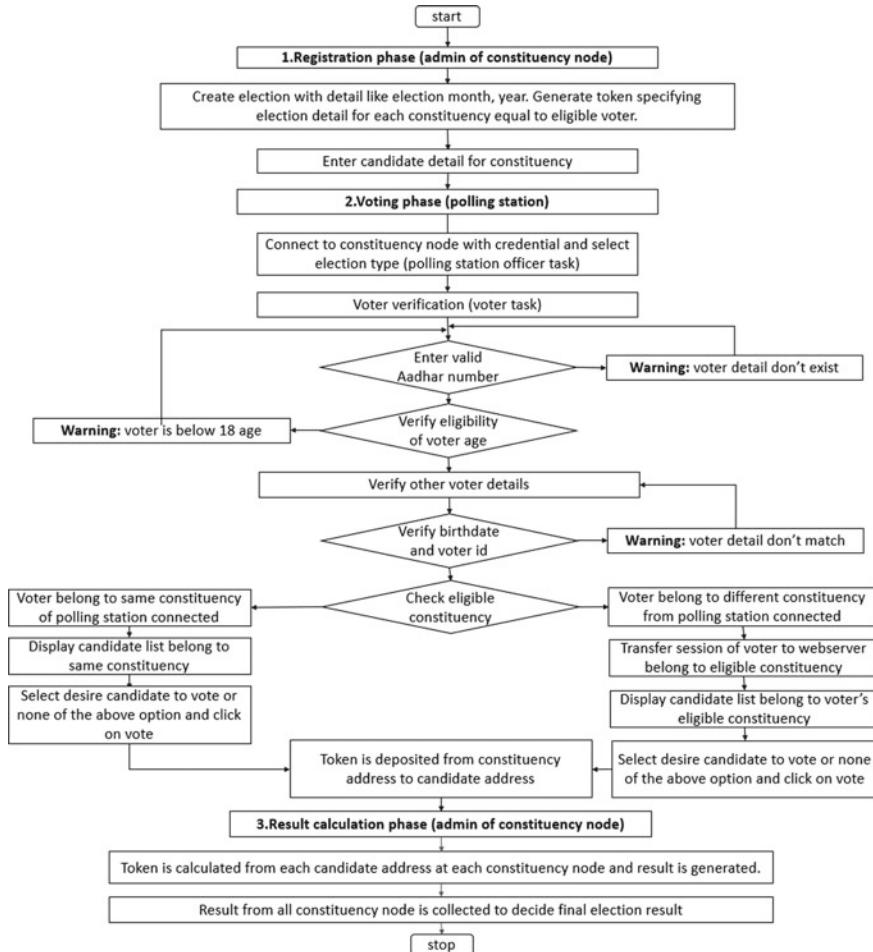


Fig. 4 Flow of Web application for Internet voting system

census report creation with local node, and report is checked with all nodes. Publisher signature and timestamp will also verify for data of different node over stream. Voting is also carried out as per voting application flow. After election admin can check election result by selecting election ID for local constituency. Result from all nodes was collected and decides winning party and winning candidate. Also token in constituency address decides voting percentage. Block analysis is done to find out real-time size requirement of all transaction and storage requirement of each node.

5.2 Observation, Analysis, and Discussion

In multichain, every activity is considered as transaction and block are mined for every transaction received. Average data size of null data transaction is 264 bytes, and ten rounds of empty mine blocks are set for this test case. Miner continuously mines if transaction receives in that interval of ten empty mining rounds else if no transaction receives after 10 empty rounds then it will stop mining. Ten empty rounds will take $10 * 264$ bytes = 2640 bytes mean 2 kb. From block analysis, it is found that size of block associated with census data entry transaction is 12,402 bytes equal to approximately 12 kb for one user data. From observation of all 50 user input data average size require to store one user data is found to be 12–13 kb. The size of block associated with transaction of creating stream is found to be 400–700 bytes. Average block size required for granting permission to address is 500 bytes; hence, granting initial permission for connecting 4689 nodes will take $4689 * 500$ bytes = 2.344 MB of block size. Average block size required for creating stream is 500 bytes. Creating 50 streams to store different data of census form required $50 * 500$ bytes = 25,000 bytes means 25 kb. After submission of census form, average block size required for storing data over multichain node is 12,710 bytes hence for storing data of 150 crore people $1,500,000,000 * 12,710$ bytes = 19 TB which is practically possible with current storage technology available (Table 1).

Maximum block size of 8 MB is defined for test case, which can be decided as per number of enumerators accessing each node for filling data simultaneously. With 12 kb of average size of individual user data, it is possible to store data of more than 6 lakh people simultaneously in one block. But processing such large data will require miner node require high-processing power device. Also, Web server requirement will be sufficient enough to handle large traffic. Block size of less than 1 MB is sufficient for application to connect and store data up to 1 lakh people simultaneously in one block.

Table 1 Required size analysis for transaction from one node (constituency)

Type of transaction	Average size of block for one transaction (bytes)	Number of quantities require for application	Total size requires for each node
Stream create	500	50	25 kb
Data entry of census form	12,710	150 crores	19 TB
Assigning permission to node address	500	4689	2.344 MB
Total required size of disk for each node			20 TB

Analysis is done considering voter count equal to 150 crore people. From analysis of block recorded voting transaction, average transaction size required for recording vote cast by one voter is 810 bytes. Total storage size required for each multichain node to record transaction of vote cast by 150 crore people is equal to $1,500,000,000 * 810 = 1,215,000,000,000$ bytes means 1 TB.

Problem with redirection of candidate list for eligible constituency from different constituency must be solved with redirecting Web traffic from one node to another node over internal isolated infrastructure rather than over Internet. Because accessing Web application of different constituency required session credential like user name and password and port to connect with another constituency node through Web server and also Web server must allow access to that voting station address for communication.

6 Conclusion and Future Scope

Digitization is truly necessary to solve various problems of current census and voting system in India. Building a secure online system is possible with private blockchain. Private blockchain is not private unless each node of blockchain is connected via private infrastructure. Bandwidth limitation of blockchain can only be minimized with isolated infrastructure, and data validation and formatting of data require best output result. NoSQL-type database can only handle large data if stored in systematic and structured manner. Mapping all user data to Aadhaar will provide data consolidation benefit as well as structured data storage on NoSQL-type database.

With proper application design, it is found that approximately 21 TB storage is required for each multichain blockchain node for storing 1.5 billion people data so practical implementation will be possible. Fix answer option for common type of census question will provide data processing advantage and faster result calculation for census result. Multichain blockchain is best suitable to build proposed infrastructure. For building secure infrastructure, planning security of each and every device is required with help of all available security methods and component. Hardening, vulnerability analysis, and patching mechanism for all components must be planned while infrastructure design.

Creating isolated infrastructure for private blockchain by creating node at each election constituency and connecting them with fiber-optic cable is time-consuming and costly process, but once infrastructure is established, then it will reduce overall cost for each election and census process. Also, it will overcome limitation of current process of census enumeration and voting system in India.

No system is completely secure. Hence, following measures must be taken while establishing constituency center. If separate servers are used for different purpose, then it will reduce processing burden on devices like one server requires for running multichain node and another server for running Web server to divide load over single machine. Scaling of device will become easier if needed in future. There is also requirement of firewall, IPS, IDS to monitor, control, and allow Web traffic. Hardening of each server is required to reduce unnecessary service and its possible vulnerability. Vulnerability analysis and patching mechanism must be present for each and every component including firewall, router, operating system, blockchain, application. Also, backup requirement and power required of each device must be taken into consideration.

References

1. Census of India: Religion. http://www.censusindia.gov.in/Data_Products/Library/Ind-ian_perceptive_link/Census_Operation_link/censusoperation.htm. Last accessed 2019/7/9
2. Yadav V, Batham S, Jain M, Sharma S (2014) An approach to electronic voting system using UIDAI. In: 2014 international conference on electronics and communication systems (ICECS), IEEE, pp 1–4. <https://doi.org/10.1109/ecs.2014.6892510>
3. Singh V, Pasupuleti H, ChandraBabu N (2017) Analysis of internet voting in India. In: 2017 international conference on innovations in information, embedded and communication systems (ICIIECS), IEEE, pp 1–6. <https://doi.org/10.1109/iciiecs.2017.8276137>
4. Election Commission of India. Wikipedia. https://en.wikipedia.org/wiki/Election_Commission_of_India. Last accessed 2019/7/9
5. Balzarotti D, Banks G, Cova M, Felmetserg V, Kemmerer R, Robertson W, Valeur F, Vigna G (2010) An experience in testing the security of real-world electronic voting systems. IEEE Trans softw Eng 36(4):453–473. <https://doi.org/10.1109/tse.2009.53>
6. List of Constituencies of the Lok Sabha. Wikipedia. https://en.wikipedia.org/wiki/List_of_constituencies_of_the_Lok_Sabha. Last accessed 2019/7/9
7. Electronic Voting in India. Wikipedia. https://en.wikipedia.org/wiki/Electronic_voting_in_India. Last accessed 2019/7/9
8. 16th Lok Sabha. Wikipedia. https://en.wikipedia.org/wiki/16th_Lok_Sabha. Last accessed 2019/7/9
9. Lok Sabha Election. General elections in india. <http://www.elections.in/indian-general-election/>. Last accessed 2019/7/9
10. Indian General Election. Wikipedia. https://en.wikipedia.org/wiki/2014_Indian_general_election. Last accessed 2019/7/9
11. Vidhan Sabha. Wikipedia. https://en.wikipedia.org/wiki/Vidhan_Sabha. Last accessed 2019/7/9
12. Matharu G, Mishra A, Chhikara P (2014) CIEVS: a cloud-based framework to modernize the indian election voting system. In: 2014 IEEE international conference on computational intelligence and computing research, IEEE, pp 1–6. <https://doi.org/10.1109/iccc.2014.7238454>
13. Greenspan G (2019) Multichain private blockchain, white paper. <http://www.multichain.com/download/MultiChain-White-Paper.pdf>. Last accessed 2019/7/9

14. Dai F, Shi Y, Meng N, Wei L, Ye Z (2017) From bitcoin to cybersecurity: a comparative study of blockchain application and security issues. In: 2017 4th international conference on systems and informatics (ICSAI), IEEE, pp 975–979. <https://doi.org/10.1109/icSAI.2017.8248427>
15. Rifa H, Rahardjo B (2017) Blockchain based e-voting recording system design. In: 2017 11th international conference on telecommunication systems services and applications (TSSA), IEEE, pp 1–6. <https://doi.org/10.1109/tssa.2017.8272896>

Chapter 23

A Cost-Efficient and Time Saving Exercise Posture Monitoring System



Sarvesh Virkud, Aditya Mehta, Necil Dabre and Jignesh Sisodia

1 Introduction

In this monotonous life, people tend to keep aside their fitness and neglect their health and well being. With the advancement in convenience due to technology the amount of physical activities in ones life have deteriorated to none. This has lead to the cause of obesity and many health related diseases. Researchers have put forth that a good physical shape plays a major role in boosting one confidence and self-esteem.

Considering that, In this fast moving world due to the unavailability of time maintaining a fit lifestyle is a challenge. Also, not being able to afford a personal trainer is one of the major reason one not being able to adapt to a healthy lifestyle. All of the fitness solutions available today include books, mobile applications and online videos helps one understand what to do but unfortunately not how to do it. Also, lack of real time monitoring to correct one's physical form during an exercise leads to serious injuries in the long run.

The system put forth makes sure that the posture of the user performing the exercise is precise. We achieve this by plotting points on all the joints of a person to form a skeleton-like figure. This figure is then used for comparison with the reference frame and the different angles formed between its key points are compared with the

S. Virkud (✉) · A. Mehta · N. Dabre · J. Sisodia
Sardar Patel Institute of Technology, Mumbai 400058, India
e-mail: virkud.sarvesh@gmail.com

A. Mehta
e-mail: mehtaaditya030@gmail.com

N. Dabre
e-mail: dabre.necil@gmail.com

J. Sisodia
e-mail: jsisodia@spit.ac.in

angles formed between the reference frame and the confidence score is shown to the user. Any anomaly, if present, is reported to the user.

2 Related Works

There are multiple works discussing Real-time posture tracking [1] by extracting skeletal joints by making use of Kinect [2–9] or inertial sensors [10]. They present a virtual trainer for performing exercise at home by making use of the Kinect sensor [2]. The proposed model provides the user with real-time feedback via two forms. First, it provides visual assistance to the user by showing a 3D view of the exercise posture from the database that is classified by the RF classifier. Secondly, it provides the user with a confidence score in real-time using which the user can assess his/her own posture by comparing performance accuracy. The main contributions of the paper are as follows:

- They present a interactive system for regular workout routines with real-time assistance that is able to provide the user with feedback for correction of errors in his posture and assessing himself.
- Then, a database of correct exercise postures has been created which stores the key skeleton points and angles between them. Random Forest (RF) classifier is used to perform classification on the exercise being performed by the user and to perform error detection. The confidence score for each exercise from RFC is extracted for the purpose of self-assessment.

A virtual trainer using kinect is created which remotely monitors the posture of a person while exercising in real-time. It showcases the user a visual aid on how to perform the exercise. If the posture of the user matches to the posture of the training data provided the confidence score is high otherwise less. They make use of the kinect sensor for depth sensing and posture tracking.

A system for posture correction for people undergoing rehabilitation activities by making use of deep neural networks [11]. By making use of Kinect sensors, the paper found out the joints of the body and using the joints, the formed a skeleton. The model is trained and the trained model is superimposed on the users skeleton to identify the anomalies. The paper proposes a new system that corrects the rehabilitating patients posture. However, no consideration was given to the accuracy measure and the difference in the users body physique from the person who trained the model.

A system to measure physical fitness by making use of Machine Vision and Human Posture Recognition [12]. There are several key techniques in the design of the test system proposed by them, one of them is using wearable sensors to detect motion of the human body while exercising, and another is a machine vision-based approach. They also perform pre-processing in the form of skeletal smoothing and human posture recognition. A machine-vision based measurement system is created to collect data related to human physical fitness using a non-contact method. They also make use of multiple digital filtering techniques to improve the test accuracy.

A system is put forth where it tracks the Yoga Posture by detecting skeletal joints using Kinect [13]. Posture detection takes place in 3 phases: Data collection, Data processing and Detection of the poses of yoga. Data collection step involves acquiring depth, color and skeletal points. The step after this involves the pre-processing of data for achieving our desired goals. In this paper, their major focus is to use the skeletal points information and angles between those points for recognition of the yoga pose.

They have proposed a model to detect 3 major yoga poses by detection of human skeleton joints using Kinect. They have recognized the yoga poses considering accuracy greater than 97% for every angle in between different parts of body. Their system can also be used to detect other poses of yoga by providing the reference model of each pose which consists of the key skeletal joints. Therefore, their finding is trying to replace a trainer while performing Yoga. A system to interpret sign languages is also put forth for deaf people by making use of Leap Motion Sensors [14]. A system which recognizes human activity based on 3-d Posture data [15].

3 Methodology

3.1 MobileNetV1 Architecture

It is a model provided by Google specifically for Mobile Vision and embedded vision uses. That is for devices having limited resources like battery and computing power. MobileNetV1 makes use of Depth wise Separable Convolution for reducing the size of the model and its complexity. Reduced model size means a smaller number of parameters to take into consideration and less complexity means fewer mathematical operation.

Depth wise separable convolution comprises of a depth wise convolution and a point wise convolution. Depth wise Convolution is the spatial convolution based on the number of channels. If we have 4 channels then, we have 4 $DK \times DK$ spatial convolution. Pointwise convolution is used for the purpose of changing the dimensions. In Fig. 1, We can see the internal layers of the MobileNet Architecture.

3.2 Skeleton Data Acquisition and Processing

In this section, we are going to design a model which recognizes a particular posture and compares it with the desired posture. The deviation from the correct posture is noted and the user is notified about the changes to improve the posture. The skeleton is extracted using a web camera and PoseNet API of tensorflow coupled with the MobileNet architecture.

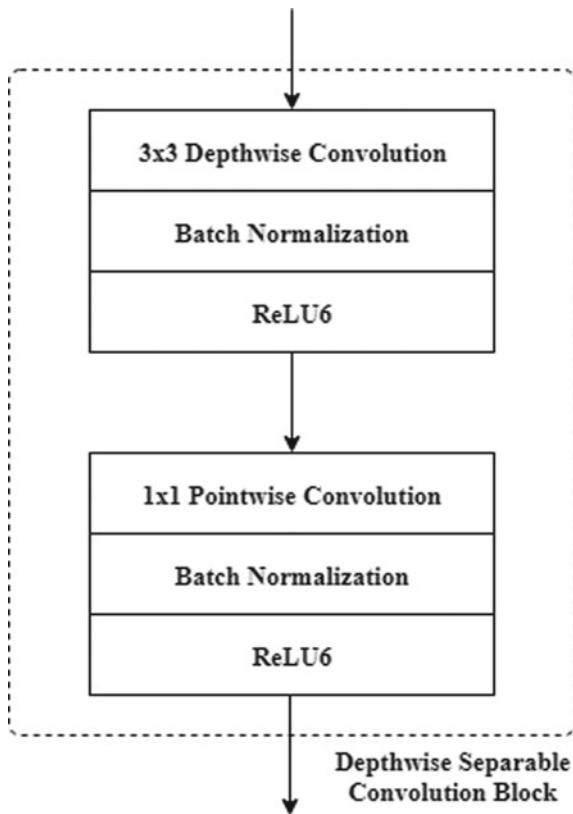


Fig. 1 MobileNetV1 layers

Skeleton Key Points Extraction

PoseNet is a machine learning model designed using Tensorflow.js to extract human body joints using the CNN algorithm. Using the MobileNet architecture, the live feed of the person is given as the input to the PoseNet model. Using CNN algorithm and MobileNet architecture, it extracts the 17 keypoints: eyes, nose, ears, shoulders, elbows, palms, hips, knees and feet. Basically, we give the PoseNet model 4 inputs such as:

- Input Image: Continuous frames from the real-time video feed of the person performing the exercise.
- Image Scale factor: Default value is 0.5. Scaling is required as the size of the model performing the exercise posture tracking may vary from the user.
- Flip Horizontal: Default value is False. It is used for flipping the frames in a web camera, ie. Left side to right side and vice versa.
- Output Stride: Affects the accuracy & speed of the model. It can be 32, 16 or 8. Higher the value, faster speeds are achieved but with a dip on the accuracy.

The output of this process will be the confidence scores identifying each body joint. All the keypoints extracted and given as input to the next phase can be seen in Fig. 2.

ID	Part	ID	Part	ID	Part
5	Left Shoulder	9	Left Wrist	13	Left Knee
6	Right Shoulder	10	Right Wrist	14	Right Knee
7	Left Elbow	11	Left Hip	15	Left Ankle
8	Right Elbow	12	Right Hip	16	Right Ankle

Fig. 2 Extracted keypoints

Formation of Human Skeleton

The output of the above process is in the form of heatmaps and offset vectors. The heatmap produces an approximate version of the skeleton. Offset Vectors improves the accuracy and help in creating the skeleton by joining the vectors in the direction to which they point for the body joints.

Extraction using the Skeleton

The major advantage of using PoseNet API is that it does the scaling purposes on its own. So, even if we provide the posture of professional body builder for training purposes, it would scale down for the user having his/her own different physique. Each keypoint has its own confidence score and the x and y positions. Using the x and y positions of the keypoints, the angle between two different joints with respect to each other can be found out. We use euclidean distance to find the distance between two particular key points of the body as shown in (1). Here, ax and ay are x and y co-ordinates for a keypoint respectively. Similarly, bx and by are x and y co-ordinates of the other keypoint.

$$\sqrt{(ax - bx)^2 + (ay - by)^2} \quad (1)$$

To find the angle between 3 keypoints, three distances $d12$, $d23$ and $d32$ are calculated using (1) where $d12$ denotes the distance between keypoint 1 & 2 and so on. Then the actual angle is extracted using (2). This extracted angle is then compared with the angle of the correct posture to be matched with a consideration 10° as deviation.

$$\arccos \left(\frac{d23^2 + d12^2 - d13^2}{2 * d23 * d12} \right) * \frac{180}{\pi} \quad (2)$$

Comparing and Displaying the Results

The models posture is evaluated by performing the above procedures and the target value of the angle between the joints involved in the respective exercise is extracted and displayed. Then, the live feed from the web camera of the user performing the exercise is provided as the input to the above procedure and the actual value of the angle between the joints involved in the respective exercise is extracted and displayed. The skeleton of the person is displayed and if the posture of the user matches with training model, the skeleton turns from black colour to green and we use a tick sound so that the user can know he is doing correctly without even looking at the screen. A deviation angle of 10° is taken into consideration from the expected angle. After evaluation of the starting posture, the user moves forward in evaluating the

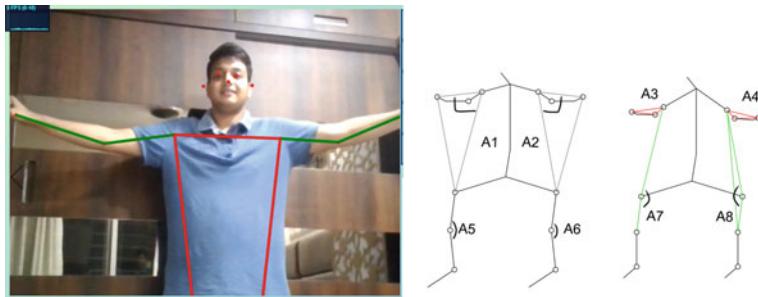


Fig. 3 Real-time skeleton points extraction and comparison with the expected posture

final posture. The same procedure is applied and thus a repetition of the exercise is evaluated. In Fig. 3, the image on the left side indicates correct posture. The image on the right depicts the angles taken into consideration for comparing the user's posture with the reference frames [2]. The architecture for extraction of skeleton points and angle between points can be seen in Fig. 4.

4 Results

All the previous existing models provide posture detection by making use of a Kinect sensor. We have implemented that by making use of a web camera, that is without the need of a depth sensor. The extracted key points from a given frame are directly compared to a real-time video stream. The extracted key points were used to form a skeleton connecting the key points. Also, the expected skeleton points were displayed to the user and the points of the user's current posture were also displayed to the user in real-time so that the user may adjust his/her posture to achieve an ideal posture. In the future, the video of the trainer performing the exercise should be used for extracting the key skeleton points and then compared with the key points extracted from the real-time video feed.

In Fig. 5, the average confidence score of users with different body types performing bicep curl is shown varying the stride of the model. In Fig. 6, the average confidence score of users with different body types performing lateral raise is shown varying the stride of the model. It can be seen that the accuracy of the system varies based on the physical parameters of the user performing the exercise. As the stride is increased, the accuracy of the system decreases.

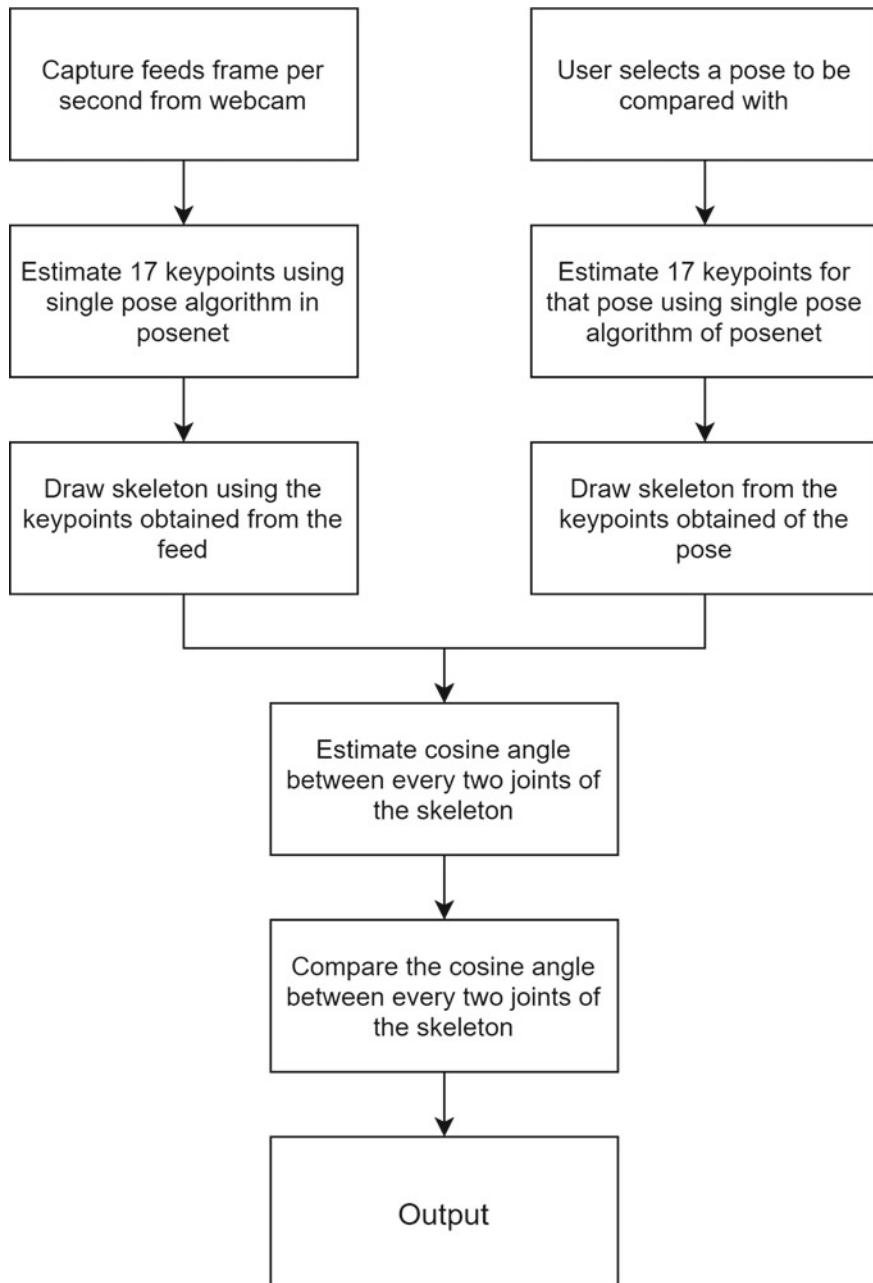


Fig. 4 Architecture for Skeleton points and angle extraction

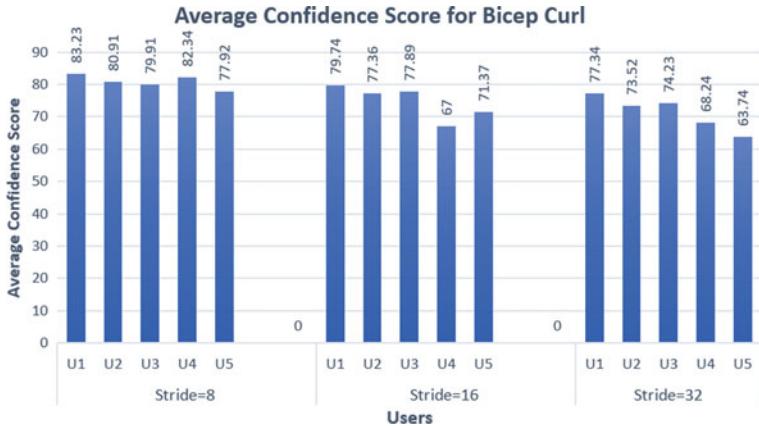


Fig. 5 Confidence scores of various users performing Bicep curl

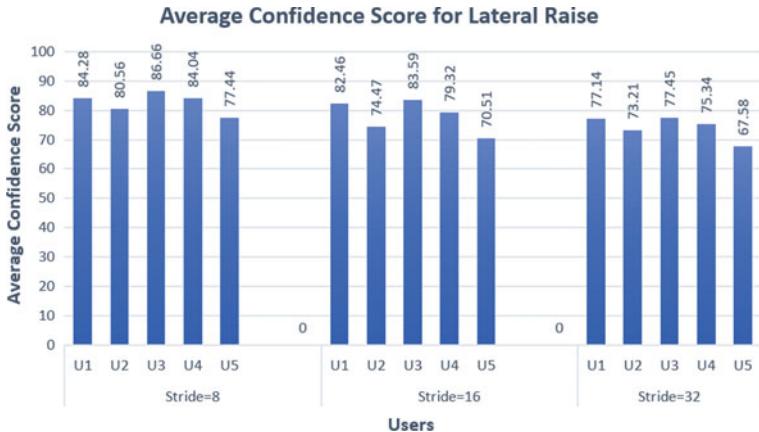


Fig. 6 Confidence scores of various users performing lateral raise

5 Conclusion

In this, we design an application using Tensorflow.js and Posenet API that extracts the skeleton of the user performing the exercise through web camera by making use of CNN coupled with MobileNet architecture. We compare the angle between the joints of the extracted skeleton performing a particular exercise with that of the trained model. If the posture is correct, the skeleton of the joints is displayed in green colour and also plays tick sound and the incorrect posture is displayed and black colour. This procedure is applied for evaluation of the initial and final posture of the exercise thus obtaining a complete monitoring application.

References

1. Fan Z, Antoine C, Yiannis D (April 2019) Probabilistic real-time user posture tracking for personalized robot-assisted dressing. *IEEE Trans Robot* 11
2. Kumar P, Saini R, Yadava M, Roy PP, Dogra DP, Balasubramanian R (2017) Virtual trainer with real-time feedback using kinect sensor. In: 2017 IEEE Region 10 Symposium (TENSYMP), 19 October 2017
3. Sarraee E, Singh S, Joshi A, Betke M (2017) PostureCheck: posture modeling for exercise assessment using the Microsoft Kinect. In: 2017 international conference on virtual rehabilitation (ICVR), 15 August 2017
4. Jin X, Yao Y, Jiang Q, Huang X, Zhang J, Zhang X, Zhang K (2015) Virtual personal trainer via the kinect sensor. In: 2015 IEEE 16th international conference on communication technology (ICCT), October 2015
5. Zhang Z, Liu Y, Li A, Wang M (2014) A novel method for user-defined human posture recognition using Kinect. In: 2014 7th International Congress on image and signal processing, October 2014
6. Le T-L, Nguyen M-Q, Nguyen T-T-M (2013) Human posture recognition using human skeleton provided by Kinect. In: 2013 international conference on computing, management and telecommunications (ComManTel), January 2013
7. Trejo EW, Yuan P (2018) Recognition of Yoga poses through an interactive system with Kinect based on confidence value. In: 2018 3rd international conference on advanced robotics and mechatronics (ICARM), July 2018
8. Rallis I, Langis A, Georgoulas I, Voulodimos A, Doulamis N, Doulamis A (2018) An embodied learning game using kinect and labanotation for analysis and visualization of dance kinesiology. In: 2018 10th international conference on virtual worlds and games for serious applications (VS-Games), September 2018
9. Zhao W, Lun R (2016) A kinect-based system for promoting healthier living at home. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC), October 2016
10. Arsenault DL, Whitehead AD (2014) Quaternion based gesture recognition using worn inertial sensors in a motion tracking system. In: 2014 IEEE games media entertainment, October 2014
11. Han S-H, Kim HG, Choi H-J (2017) Rehabilitation posture correction using deep neural network. In: 2017 IEEE international conference on big data and smart computing (BigComp), 20 March 2017
12. Cheng X, He M, Duan W (2018) Machine vision based physical fitness measurement with human posture recognition and Skeletal data smoothing. In: 2017 international conference on orange technologies (ICOT), 12 April 2018
13. Islam MU, Mahmud H, Ashraf FB, Hossain I, Hasan MK (2018) Yoga posture recognition by detecting human joint points in real time using Microsoft kinect. In: 2017 IEEE Region 10 humanitarian technology conference (R10-HTC), 12 February 2018
14. Behera SK, Kumar P, Dogra DP, Roy PP (2017) Fast signature spotting in continuous air writing. In: 2017 Fifteenth IAPR international conference on machine vision applications (MVA), May 2017
15. Gaglio S, Re GL, Morana M (December 2014) Human activity recognition process using 3-d posture data. *IEEE Trans Human-Mach Syst*

Chapter 24

Sarcasm Detection on Twitter Data: Generative Versus Discriminative Model



Ashwini M. Joshi and Sameer S. Prabhune

1 Introduction

Wikipedia [1] defines sarcasm as a use of irony to pretend or convey disregard for something. Sarcasm is dependent on context, prior knowledge of the situation and on the tone of voice. This makes it fundamentally difficult to analyse, not just for machines but also for humans [2]. Despite this, sarcasm is a common occurrence in social media.

The Naive Bayes classifier and a Recurrent Neural Network are used to identify sarcasm in tweets, as it represents generative and discriminative models, respectively. The Naive Bayes classifier is the simplest model in Natural Language Processing to perform classification. RNN is an effective ML tool that simulates the function of a brain artificially. RNN is a simulation of the working of the human brain using network units known as neurons, which maintain internal state. RNNs make predictions based on the state maintained by neurons. It gives weights to the input data based on its importance and predicts the output. Based on the error, it updates its model parameters to predict better, until the best possible accuracy is reached. The inputs to the RNN are vectors generated by Word2vec, which is a tool for learning vector representations of words.

A. M. Joshi (✉) · S. S. Prabhune
Department of CSE, SGBAU, Amaravati, Maharashtra, India
e-mail: ashwinimjoshi@pes.edu

S. S. Prabhune
e-mail: ssprabhune@gmail.com

2 Procedure

2.1 Data Collection and Pre-processing

Data sets of sarcastic tweets were available for download from the Internet. They were collections of tweets that had '#sarcasm' or '#sarcastic' in them. The fact that it needed to be explicitly mentioned that the tweets are sarcastic made them either context-dependent or difficult to classify as sarcastic [3]. A more apt hashtag for sarcastic tweets was '#not'. Since this was a custom requirement, no data set was available. A Python library called Tweepy was used for accessing the Twitter API and collecting tweets. A filter was applied for tweets containing '#not'. Non-sarcastic tweets were collected the same way, without applying any filters. Basic pre-processing such as removal of URLs, usernames, hashtags and punctuation was done.

2.2 Classification Using Naive Bayes

The Naive Bayes classifier was implemented as a binary classifier. It was trained with a data set containing both sarcastic and non-sarcastic tweets. Later, it was tested with random tweets to measure its accuracy. The job of the classifier was to predict whether the tweet was sarcastic or not.

2.3 Vector Representation of Words

All the words in the training data set are converted into appropriate vector representations based on the Word2vec model. The set of all the words present in the training data set is called as vocabulary.

2.4 Classification Using RNN

A Recurrent Neural Network was implemented based on Karpathy [4]. Forward propagation was performed with random values as the model parameters. Back-propagation through time was implemented on the RNN to train the network and enable the RNN to learn and maintain state information about the data set. Stochastic Gradient Descent was used to update the model parameters in each iteration of back-propagation. Forward propagation was performed again to analyse the effectiveness.

3 Literature Survey

Sarcasm can be broadly classified into two types, i.e. context-dependent and linguistic [5]. Detection of context-dependent sarcasm is difficult, not just for computers but humans as well, because it needs knowledge of the situation in which the remark was made, background knowledge of the person's opinions, etc. Detection of linguistic sarcasm is not as tedious because it depends on the lexical structure of a sentence, along with the parts and figures of speech used. Essentially, this has a structure and hence can be learned as a pattern for a mathematical model.

A system for sarcasm detection in social media text is automated using six algorithms which are capable of detecting and analysing different types of sarcasm proposed in [6]. These algorithms use pragmatic, lexical, contextual and hyperbolic features of text to recognize sarcasm. In the contextual feature, we mainly focus on situation, temporal, historical and topical context of the text. The results of proposed approaches were compared with advanced technique.

A rule-based approach to detect sarcasm expressed due to numbers is presented in [7, 15]. This approach compares numerical magnitudes with those seen in similar contexts in a training data set. Since 'similar context' is key here, two variants of this approach are considered in order to match the context.

Machine learning-based approach and its variant that take different features as input for learning can also be used for detecting sarcasm. Deep learning-based approaches to numerical sarcasm detection on social media that does not require extensive manual feature engineering can also be used [7].

4 Methodology and Implementation

4.1 Naïve Bayes

The most simple classification tool in Natural Language Processing is Naïve Bayes Classifier. As per Bayes' theorem, if it is given that event x has occurred, then the probability of occurrence of an independent event c is given by the formula (1)

$$P(c|x) = \frac{P(c) * P(x|c)}{P(x)} \quad (1)$$

This classifier makes assumptions about all the features of the input being independent, which accounts for the presence of the word 'naïve' [8].

4.2 Word2vec

Word2vec [9] is a very effective tool for converting words to vector representations, while maintaining the semantics of the context the word appears in. The mathematical formula behind this is given in (2) [10]:

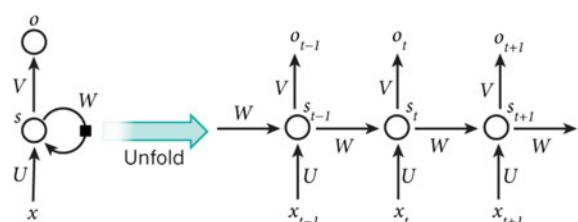
$$p(w_j|w_I) = \frac{\exp(\mathbf{v}'_{w_O}^T \mathbf{v}_{w_I})}{\sum_{j'=1}^V \exp(\mathbf{v}'_{w'_j}^T \mathbf{v}_{w_I})} \quad (2)$$

where w_O is a neighbouring word and w_I is the word whose vector is being generated. Each word has an outer vector V'_{w_O} and an inner vector V_{w_I} . W is the vocabulary of all words present in the data set. Based on these predictions, the vector representations of each word in the vocabulary are made. The vectors for each word have the same length. These are given as inputs to the RNN. This appears to be the most appropriate way of converting words to vectors because in a sarcastic comment, the semantic structure and word orientation needs to be maintained.

4.3 Recurrent Neural Networks

A Recurrent Neural Network [11] is a broader category of artificial neural networks (ANN). ANNs are a family of machine learning models that are inspired by the human brain. An RNN has a special feature of retaining information when used on sequential data. When a sequential input is given to an RNN, each unit gets some feedback from the previous unit in the sequence. This means that the RNN is capable of storing information from the previous time. This is useful because the current unit of the input is dependent on the previous unit. This feature of an RNN is called unfolding. As shown in Fig. 1 [12], an RNN consists of an input layer, one or more hidden layers and an output layer. Each layer consists of a set of units which are called neurons. The vector representation of the word given as the input at index t in the sequence is called x_t . Weights are assigned to each input unit randomly. Weights correspond to the amount of importance that particular input unit has in the application. The weight matrix used is U . The weighted input is passed to the hidden layer, which applies a

Fig. 1 Unfolding an RNN



mathematical function on the input and produces an output. Commonly, nonlinear functions such as tanh (hyperbolic tangent function) and sigmoid functions are used and produce S_t . S_t is again weighted based on importance of units using the matrix V and passed to the output layer. The output layer now has a vector, O_t , where each term in the vector represents a probability of that word belonging to some class. The softmax function is applied on the output to produce an even probability distribution among all the output classes. There is a weighted matrix W , between the hidden layer as well, which corresponds to how much past information is to be maintained and how much can be discarded. Error is measured with respect to expected probability for each class. Now, error is minimized by performing back-propagation through time, where the model parameters are tweaked in such a way that error decreases. Sometimes, the output vector is not required at all units because only the overall gist of the sentence is required, not just of each word. In such cases, the error obtained at all these units is assumed to be zero.

4.4 Stochastic Gradient Descent

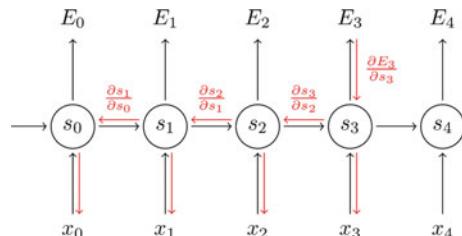
Gradient Descent is an algorithm which updates the model parameters of the system, i.e. the weight matrices of the system based on the error and a learning rate [13]. The expression for Gradient Descent is as follows:

$$\theta = \theta - \alpha \nabla_{\theta} E[J(\theta)] \quad (3)$$

Here, θ is a model parameter that needs to be updated. The LHS of the equation represents the updated value, while the RHS has the old value of θ . ∇_{θ} is the gradient of $E[J(\theta)]$ with respect to θ . $E[J(\theta)]$ is the error in the model. Here, evaluation is done over the full training set. This is also known as batch Gradient Descent, where the batch is the training data set. On the other hand, Stochastic Gradient Descent performs this update for each element in the training data set. As compared to Batch Gradient Descent, Stochastic Gradient Descent usually converges faster.

Figure 2 shows the working of Stochastic Gradient Descent in a Recurrent Neural Network [14]. At s_3 , it can be seen that the error back-propagation occurs to the input layer and to the hidden layer of the previous unit in the sequence.

Fig. 2 Stochastic Gradient Descent Recurrent Neural Networks



5 Results and Discussion

The accuracy obtained with the Naive Bayes classifier was about 73.76%. In contrast, the Recurrent Neural Network performed better. When two units were taken in the hidden layer, an accuracy of about 80.23% was obtained, with four units, around 85.25% was obtained, whereas with eight hidden units, the accuracy decreased to around 83.18%. In Stochastic Gradient Descent, the number of epochs (iterations) was about 60 times the size of the data set. These observations show that the Recurrent Neural Network has more representational power than a Naive Bayes classifier. The RNN turns out to be more effective and learns much better than the Naive Bayes classifier. The comparison of Naïve Bayes and RNN with respect to various features, as per our observation, is shown in Table 1 [16].

Another disadvantage of the Naive Bayes classifier is the fact that it assumes all words to be independent of each other, which is not the case in real-world data. This assumption clearly hinders its performance in detection of linguistic sarcasm.

The main advantage of an RNN in this case is that it can store information from the beginning of the sentence till the end, in the form of the weight matrix between the hidden layers. On the other hand, the Naive Bayes classifier predicts based on likelihood, which is not as relevant to sarcasm as it may be in other applications. Few (if any), words have the property that they may occur only in a sarcastic context. In contrast to this, the problem of identifying the polarity of a statement makes use of likelihood as many words may occur only in a certain context.

Table 1 Comparison between Naïve Bayes and RNN with various features

Feature	Naïve Bayes	RNN
Based on	Bayes' theorem	Artificial neural networks
Simplicity	Very simple	Advance
Performance	Good	High
Accuracy	Good	Very good
Memory requirement	Low	High
Other applications	Spam detection Document Classification Recommender System	Language modelling Text generation
Result accuracy over a period of time	Variable	Consistent
Time required for training	Less	High

6 Conclusion

The comparison of Naïve Bayes (NB) and RNN with respect to various features, as per our observation, is shown in Table 1.

Another disadvantage of the Naive Bayes classifier is the fact that it assumes all words to be independent of each other, which is not the case in real-world data. This assumption clearly hinders its performance in detection of linguistic sarcasm. Thus it can be concluded that RNN is better performing than NB for sarcasm detection.

6.1 Future Work

This work can be extended further to incorporate detection of context-dependent sarcasm as well. This can be done by using techniques that help in training the system to learn the context in which the tweet was made. Related tweets can be collected, and the context can be modelled. This context would then help determine whether the tweet in the question is sarcastic or not. Detection of sarcasm would help user in better decision-making.

References

1. Wikipedia.org, “Sarcasm,” [Online] Available: <https://en.wikipedia.org/wiki/Sarcasm>
2. Maynard D, Greenwood MA Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis [Online]. Available: <https://gate.ac.uk/sale/lrec2014/arcomem/sarcasm.pdf>
3. Rajadesingan A, Zafarani R, Liu H Sarcasm detection on twitter: a behavioral modeling approach [Online]. Available https://repository.asu.edu/attachments/140784/content/Rajadesingan_asu_0010N_14283.pdf
4. Karpathy A The unreasonable effectiveness of recurrent neural networks [Online]. Available <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
5. Ptáček T, Habernal I, Hong J Sarcasm detection on czech and english Twitter [Online]. Available <https://aclweb.org/anthology/C/C14/C14-1022.pdf>
6. Sarcasm-Detection-in-Twitter-Data -A-Supervised-Approach—<https://www.researchgate.net>
7. Sarcasm detection on twitter data, Soumya brata Maity Kumarjit Ghosh Nishan Singh Sekhon Rohit Kumar Jaiswal, http://rcciiit.org/students_projects/projects/cse/2018/GR6.pdf abcd
8. Wikipedia.org, Naive Bayes classifier [Online]. https://en.wikipedia.org/wiki/Naive_Bayes_classifier
9. Deeplearning4j.org, Word2Vec [Online]. Available: <http://deeplearning4j.org/Word2vec>
10. Wikipedia.org, Word2vec [Online]. Available: <https://en.wikipedia.org/wiki/Word2vec>
11. Mikolov T, Kombrink S, Deoras A, Burget L, Cernocky JH, RNNLM—recurrent neural network language modeling toolkit. Available: IEEE automatic speech recognition and understanding workshop
12. Britz D Recurrent neural networks tutorial, Part 1—introduction to RNNS [Online]. Available: <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorialpart-1-introduction-to-rnns/>

13. UFLDL tutorial, Optimization: stochastic gradient descent [Online]. Available <http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/>
14. Britz D <http://www.wildml.com/2015/10/recurrent-neuralnetworks-tutorial-part-3-backpropagation-through-time-and-vanishinggradients/>
15. Mahendran A, Duraiswamy A, Reddy A, Gonsalves C (2013) Opinion mining for text classification. *Int J Sci Eng Technol* 2(6):589–594
16. Gupte Amit, Joshi Sourabh, Gadgil Pratik, Kadam Akshay (2014) Comparative study of classification algorithms used in sentiment analysis. *Int J Comput Sci Inf Technol* 5(5):6261–6264

Chapter 25

Network Intrusion Detection System Using Machine Learning Approach



Mrunal Teli, Riya Singh, Minal Kyada and Ramchandra Mangrulkar

1 Introduction

A network setup is generally seen everywhere in every corner of the world. It is recommended for any organisation to continuously monitor their entire network. In today's era, there have been historical security attacks faced by major tech giants. Security is broad illusion trying to protecting wider network. With the increasing demands of the mankind, technology is bound to take over. With the increasing technology usage, there is an increasing need to protect the product, in use.

An intrusion detection system (IDS) is a software application that monitors network or system activities for malicious activities and unauthorised access to devices. IDS comes in a variety of 'flavours' and approaches the goal of detecting suspicious traffic in different ways. There are network-based (NIDS) and host-based (HIDS) intrusion detection systems. There is IDS that detects based on comparing traffic patterns against a baseline and looking for anomalies. There is IDS that simply monitors and alerts and there is IDS that performs an action or actions in response to a detected threat. We will cover each of these briefly.

Network intrusion detection systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally,

M. Teli · R. Singh · M. Kyada (✉) · R. Mangrulkar

University of Mumbai, Dwarkadas J. Sanghvi College of Engineering, VileParle, Mumbai 400056, India

e-mail: mpkyada@gmail.com

M. Teli

e-mail: mrunal.teli@gmail.com

R. Singh

e-mail: riasingh13597@gmail.com

R. Mangrulkar

e-mail: ramchandra.mangrulkar@gmail.com

you would scan all inbound and outbound traffics; however, doing so might create a bottleneck that would impair the overall speed of the network. When designing a IDS, the mission is to protect the data's confidentiality. IDS basically does not actually maintain integrity of data. As it is not a preventing system, it is detection system. But it indirectly helps as there are other ways to prevent the attack from happening. This is done by prediction algorithms. It can prevent the attacks from happening in future by storing current attack prone data.

2 NIDS Background

Network intrusion detection systems are software tools that monitor the network of an organisation, hospital, institution, banks, etc. NIDS is used for detection of abnormal behaviours on network based on the assumptions that the behaviour of the intruder is going to be different from that of a normal user. It is capable of detecting activities which cannot be detected by conventional firewalls. It is a passive alert system (it can detect and alert the system and not prevent it by itself) [1].

Imagine ants and their hunt for food. They have unique ability to find the shortest path from their respective homes and food. Despite of their inability of lack of vision and deprived speech, they seem to have a stern conduct. They communicate by depositing pheromone along the path they find towards the food. As soon as they reach the food and carry it back to their home, they leave traces of pheromone. The amount of ants following the same path increases the density of pheromone on that specific path. This pheromone evaporates with time, so the probability of no pheromone traces is highest for the longest path. After time ' t ', ants would not be able to find those traces and eventually they will follow the shortest path.

Ant colony optimisation algorithm for constructing decision tree uses a new metaheuristics approach in machine learning procedures. Each ant selects a suitable attribute in this algorithm based on the heuristic function and the pheromone values. The heuristic function enables ants to split the objects into two groups connected to the examined values of the attributes. Therefore, the one which allows most in the splitting process is regarded as the best condition for the building of the decision tree. The division is considered the best when the model defines with highest possible uniformity, the same number of objects in the left and right subtrees. Pheromone values are the best way (link) between root and child nodes—all possible subtree combinations [2] (Fig. 1).

Each ant builds a decision tree at the beginning of its work. At the end, the best decision tree will be chosen and the pheromone will be upgraded as per the splits produced during the decision tree building process. Agents—ants examine previous systems while building the tree, and other adjustments are conducted in the single node. This process is iterated till the best decision tree is obtained [3].

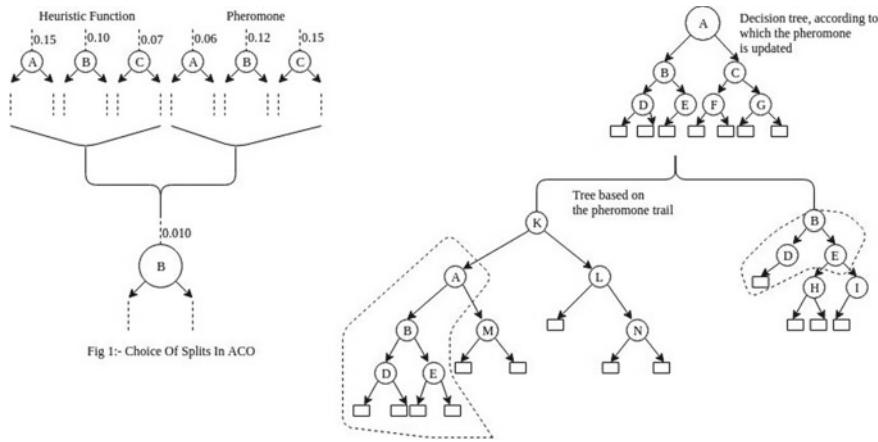


Fig. 1 Building the decision tree with phenomene

3 Proposed Network Intrusion Detection System

These days, achieving high-level security is quite essential to ensure secure and trustworthy, information, communication between different organisations. Intrusion detection system is a scheme of secure technology following conventional technologies like firewall, message encryption and so on. The worldwide Web of today is hazardous to humans and specifically to their bank savings. To conquer attacks on processes that are particularly vulnerable, this network traffic trying to circumvent system helps to identify nefarious attackers and prevents security issues. The proposed model can detect and prevent the negative aspects of the network with the support of operable algorithms adhered with preventive dataset. The proposed system will help identify four types of attacks. The system is initially provided with the input of KDD'99 datasets. The data is churned by the scale of the feature. Generally speaking, the data is in two parts of the training and test datasets. Cleaning and pre-processing of the training set. The data is supplied to the algorithm of SVM. SVM algorithm checks whether the sample is good or bad. The suitable results are demonstrated and further classified on the basis of outputs [4].

Network intrusion detection can be scrutinised as classification problem where each packet is identified either as normal or one of the attack types based on some existing data.

Intrusion detection systems are categorised as network- or host-based approach for safeguarding the network. In either case, these products look for attack-specific patterns that usually indicate malevolent interruption. In network based, IDS monitors entire network and in host based, it looks into system log files (Fig. 2).

A completely secure system is an illusion in today's world. There are two approaches to maintain the privacy and security of the system, viz., detect the vulnerabilities or prevent them. Cryptographic methods come to terms if the passwords and

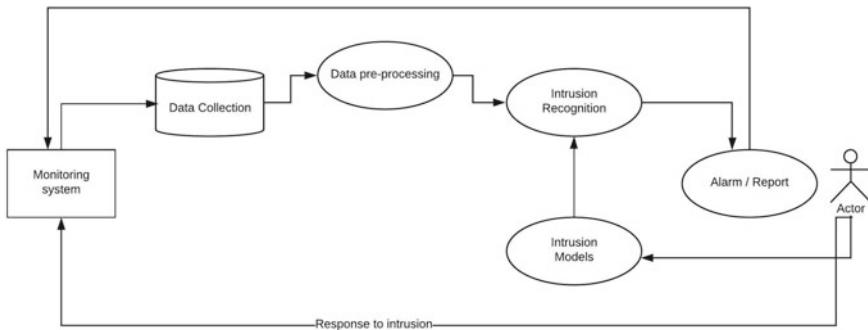


Fig. 2 NIDS architecture

keys are stolen. It is susceptible to insiders, even though the system is protected, who abuse their privileges. The level of access control and efficiency are inversely proportional. More access control hierarchy makes the system less user friendly. Intrusion detection software evaluates what happened throughout any program execution and finds evidence of misuse of the computer. An intrusion detection system encompasses a preventive function, but it works as the last defensive measure to secure the system. Various techniques are sculpted from statistics fields, pattern recognition, natural language processing and databases. Classification is a technique studied as useful for models of intrusion detection. We examine the decision tree as a model for intrusion detection in this section of the book.

There are various algorithms being developed and evolved but there is no proper parameter or measure to find out which one suits the best. The process of examining events and analysing the signatures is known as intrusion detection. Detection of intrusion is divided into two types:

- Misuse intrusion detection system.
- Anomaly intrusion detection system.

3.1 Misuse Intrusion Detection System

Misuse intrusion detection uses straight behaviours of attack which really affect system efficiency and application software to define intrusions. These patterns are pre-ciphered which are then used to reflect the user behaviour to detect intrusions.

3.2 Anomaly Intrusion Detection System

Anomaly intrusion detection uses patterns of user behaviour to detect the intrusion. They are constructed from the statistical measures of the system, for example, the CPU and I/O activities by a particular user or program. The deviation in user behaviour is noted [5].

4 Decision Tree Approach for Intrusion Detection

Detection of intrusion can be scrutinised as a problem of classification where all individual connections or users are discerned either as an attack type or a normal packet based on existing dataset. The algorithm used in the project helps solve the classification problem by learning the model from available training dataset and can discern some test dataset into one of the classes mentioned above in the KDD'99 Dataset [6].

Iterative Dichotomiser 3—algorithm tends to give maximum accuracy with huge dataset and the amount of traffic that flows in a network is tremendous; therefore, decision tree is used to solve this classification problem. It is used in real-time intrusion detection because of its high performance [7]. Iterative Dichotomiser 3 algorithms construct simple decipherable models, which ultimately are of great help to security officers for inspection and editing. Another useful property of NIDS is ‘Generalisation Accuracy’. This property helps to detect some new attacks which are small variations of known attacks after the NIDS model is built [8] (Fig. 3).

NSL-KDD Dataset is used for experimentation in this chapter. MIT Lincoln Labs prepared this dataset (NSL-KDD) in the year 1998 by DARPA Intrusion Detection Evaluation Program. The researchers at MIT acquired raw TCP dump data for almost about nine weeks since the day it all started. It is said that there are in all 24 attack types

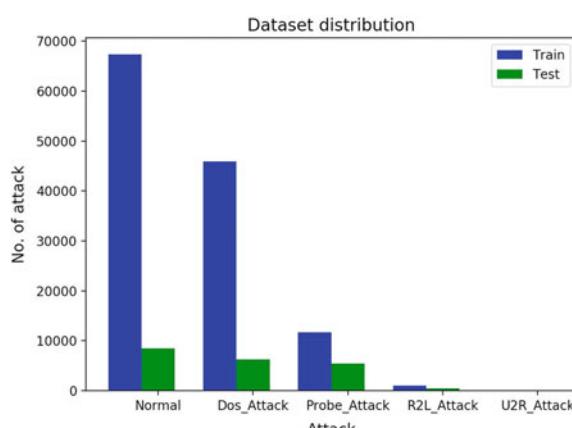


Fig. 3 Dataset distribution (train + test)

in NSL-KDD dataset. When the raw data was processed into connection records, it turned out that there are about five million such records in the dataset [9].

There are four main categories of attacks in the dataset:

1. DOS—Denial Of Service
2. Probe
3. R2L—Remote to User
4. U2R—User to Root

Probe: The attacker collects information about the system or computer network to find (known) vulnerabilities, by scanning a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited in order to compromise the system.

DOS: The attacker does not allow legitimate users access to computing resources or overloads them so that requests cannot be processed in real time. The result of this attack is the unavailability of resources, i.e. resources are too busy or too full to serve legitimate networking requests and hence denying users access to a machine.

U2R: The attacker explores vulnerabilities in order to acquire administrator privileges (root access to the system). Attacker starts off on the system with the normal user account and looks for vulnerabilities in order to gain super user privileges.

R2L: The attacker does not have a user account on the victim machine, hence tries to obtain access to the remote system without having the account.

ID3 algorithm provides a digestible visual representation of a classification model in which the internal nodes correspond to the decision nodes and the child nodes correspond to the class labels which are already predicted. To classify, the tree is traversed from the root to the child node in a top-down approach, moving down the tree by selecting branches based on subordinate node attribute test results until the last child node is reached [10].

An Iterative Dichotomiser 3 algorithm selects attributes based on the information gathering (derived from the entropy measure commonly used in the theory of information) [11].

To select the best attribute to create a node, it avails an entropy-based criterion, called the information gain ratio. In crux, entropy measures a collection of examples 'impurity relative to their class attribute values', where higher entropy values are more consistently distributed examples, whereas lower entropy values are more homogeneous examples. The entropy of an example S collection is given by,

$$\text{Entropy}(S) = \sum_{c=1}^m -p_c \cdot \log_2 p_c \quad (1)$$

where p_c is the percentage of examples associated with the c th class label in S and m is the number of class labels in total. Using the entropy measure, the gain of information from an attribute A equates to the expected decrease in entropy achieved by fragmenting the training examples into T subsets, where T is the series of different values in the attribute A domain and is delineated as,

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v=1}^T \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v)$$

where $|S_v|$ is the number of examples in the S subset under which A attribute has the v th value in the domain A and $|S|$ is the sample number in S . The information gain margin computation incorporates a sanction for attributes which separate the training instances into rather small sets of data, called the split information, as calculated

$$\text{Split information}(S, A) = \sum_{v=1}^T -\frac{|S_v|}{|S|} \cdot \log_2 \frac{|S_v|}{|S|}$$

Lastly, the information gain margin of A attribute is extracted from both the information gain and split measures and is specified by

$$\text{Gain Ratio}(S, A) = \frac{\text{Gain}(S, A)}{\text{Split information}(S, A)}$$

At each and every step in the top-down strategy, the selection of the decision tree recommends an attribute that optimises the conversion rate of information gain, which later correlates with the attribute that gives the greater entropy gain [12].

5 Working of NIDS Model

Network intrusion detection systems are software tools that monitor the network of an organisation, hospital, institution, banks, etc. NIDS is used for detection of abnormal behaviours on network based on the assumptions that the behaviour of the intruder is going to be different from that of a normal user. It is capable of detecting activities which cannot be detected by conventional firewalls. It is a passive alert system (it can detect and alert the system and not prevent it by itself).

The decision tree algorithm for intrusion detection follows four basic steps:

- *Data pre-processing*
- *Feature selection*
- *Build the model*
- *Prediction and evaluation (validation)*.

5.1 Data Pre-processing

Using one-hot-encoding, all features are made numerical. The features are adjusted to avoid high-value characteristics that can weigh much more in the results.

Table 1 Original categorical data

	Protocol_type	Service	Flag
0	tcp	ftp_data	SF
1	udp	other	SF
2	tcp	private	S0
3	tcp	http	SF
4	tcp	http	SF

Table 2 Data after one-hot-encoding

	Protocol_type_icmp	Protocol_type_tcp	Protocol_type_udp
0	0.0	1.0	0.0
1	0.0	0.0	1.0
2	0.0	1.0	0.0
3	0.0	1.0	0.0
4	0.0	1.0	0.0

On-hot-encoding which is one-of-K is used to convert all categorical functionality in binary functionality. One-hot-encoding requirement: ‘The input to this generator should be an integer matrix, signifying the values assumed by categorical characteristics also called as discrete features’. The features must therefore first be transformed with label Encoder in order to transform each category into a number (Tables 1 and 2).

5.2 Feature Selection

Eradicate completely useless and inconsequential data by selecting a tiny proportion of appropriate elements which thoroughly depicts the issue in discussion. Selection of univariate feature with ANOVA F-test. This examines each feature independently to simulate the intensity of the feature-label relationship. Use second percentile (sklearn.feature selection) strategy to choose features centred on the strongest scores percentile. Recursive feature elimination—RFE—is applied once this fraction of set is found (Figs. 4 and 5).

5.3 Build the Model

There are, in theory, enormously several decision trees that can be built from a load of attributes. Although some of the trees are much more accurate than many others,

```

: print('Features selected for DoS:',rfecolname_DoS)
print()
print('Features selected for Probe:',rfecolname_Probe)
print()
print('Features selected for R2L:',rfecolname_R2L)
print()
print('Features selected for U2R:',rfecolname_U2R)

Features selected for DoS: ['src_bytes', 'dst_bytes', 'wrong_fragment', 'num_compromised', 'same_s
rv_rate', 'diff_srv_rate', 'dst_host_count', 'dst_host_same_srv_rate', 'dst_host_error_rate', 'ds
t_host_srv_error_rate', 'service_ecr_i', 'flag_RST', 'flag_S0']

Features selected for Probe: ['src_bytes', 'dst_bytes', 'error_rate', 'dst_host_same_srv_rate',
'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_error_rate', 'service_finger',
'service_ftp_data', 'service_http', 'service_private', 'service_smtp', 'service_telnet']

Features selected for R2L: ['duration', 'src_bytes', 'dst_bytes', 'hot', 'num_failed_logins', 'num
_access_files', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_same_s
rc_port_rate', 'dst_host_srv_diff_host_rate', 'service_ftp_data', 'service_imap4']

Features selected for U2R: ['duration', 'src_bytes', 'dst_bytes', 'hot', 'root_shell', 'num_file_c
reations', 'num_shells', 'srv_count', 'dst_host_count', 'dst_host_same_srv_rate', 'dst_host_srv_di
ff_host_rate', 'service_ftp_data', 'service_other']

```

Fig. 4 Features selected by RFE

```

: print(X_rfeDoS.shape)
print(X_rfeProbe.shape)
print(X_rfeR2L.shape)
print(X_rfeU2R.shape)

(113270, 13)
(78999, 13)
(68338, 13)
(67395, 13)

```

Fig. 5 Count of features selected

it is computationally impossible to find the ideal tree due to the search space's exponential size. Nonetheless, in a reasonable period of time, cost-effective algorithms were developed to stimulate a relatively accurate, though suboptimal, decision tree. Usually, these analysis tools use a greedy approach that develops a decision tree by creating a series of locally optimum decisions as to which attribute to be used for data partitioning. One such algorithm is ID3—Iterative Dichotomiser Version 3—which is the principle of so many established algorithms for decision tree induction (Fig. 6).

```

: # selected features
clf_rfeDoS=DecisionTreeClassifier(random_state=0)
clf_rfeProbe=DecisionTreeClassifier(random_state=0)
clf_rfeR2L=DecisionTreeClassifier(random_state=0)
clf_rfeU2R=DecisionTreeClassifier(random_state=0)
clf_rfeDoS.fit(X_rfeDoS, Y_DoS)
clf_rfeProbe.fit(X_rfeProbe, Y_Probe)
clf_rfeR2L.fit(X_rfeR2L, Y_R2L)
clf_rfeU2R.fit(X_rfeU2R, Y_U2R)

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_split=1e-07, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      presort=False, random_state=0, splitter='best')

```

Fig. 6 Procedure for building the model with selected features

5.4 Prediction and Evaluation (Validation)

Decision tree algorithm is perhaps the primary algorithm used to train the model using initially constructed help functions. First, the function called Train Test Split splits the set of data into train and test datasets. Once the data is divided, the function Data Pure and Classify is called to test data purity and characterise data based on purity.

Use the test statistics to make model predictions. Various scores such as:

- Accuracy score
- Recall
- F -measure
- Confusion matrix [13].

6 Results and Discussions

A. Tree interpretation

The consequent clustering algorithm can be easily deciphered when building trees that use the decision tree algorithm, is also one of decision tree's greatest advantages [15]. Consider a subtree given in Fig. 7 that was acquired after enforcing the algorithm to the NSL-KDD dataset; where, respectively, 'ecr_i' and 'pod' signifies 'echo_response_ICMP' and 'ping of death'. This implies that a service type 'ecr_i' connection specific example with both a count ≤ 20 as well as $\text{src_byte} > 1256$ will be marked as an attempted POD attack.

B. Performance measure

The proposed model would have to compute the values of True Positive—TP, False Positive—FP, True Negative—TN and False Negative—FN to demonstrate Accuracy, Precision and Recall as our key performance metrics. TP represents the instances that are an attack and classified as an attack. FP symbolises cases that are, in fact, normal but categorised as an attack. FN reflects instances that are, in fact, an attack but listed as normal. TN signifies instances that are, in fact, normal and identified as normal.

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

Fig. 7 Decision tree (subtree) pseudocode

```

1 'service' = ecr_i
2   'count' <= 20.500000
3     'src_bytes' <= 292.000000
4       'src_bytes' <= 25.000000 : ipsweep
5         'src_bytes' > 25.000000 : normal
6       'src_bytes' > 292.000000
7         'src_bytes' <= 1256.000000 : smurf
8         'src_bytes' > 1256.000000 : pod

```

Table 3 Comparative results

	DOS (%)	PROBE (%)	U2R (%)	R2L (%)
Accuracy	99.64	99.57	99.66	97.95
Precision	99.50	99.39	86.48	97.22
Recall	99.66	99.27	91.67	96.98
F-measure	99.58	99.33	88.63	97.09

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{Predicted Positive}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{Actual Positive}}$$

Simply put, Accuracy symbolises the number of sets properly classified, Precision puts forth—among all instances classified as an attack how many were actually an attack, and Recall tries to represent the number of attacks properly classified—the percentage of attacks caught [14] (Table 3).

Confusion Metrics: A confusion matrix is a table which is often used to define a classification model's performance (or classifier) on a set of test data known for the true values. It enables the performance of an algorithm to be visualised. It enables confusion between classes to be easily identified, e.g. one class is generally mislabelled as the other. The confusion matrix computes most performance measurements. A CM is a summary of outcomes of predictions of classification. The amount of right and inaccurate predictions is summarised and broken down by each class with count values. This is the answer to the confusion matrix.

The confusion matrix demonstrates how, when making predictions, your classification model is confused. It provides us insight not only into a classifier's errors, but more importantly, the kinds of errors being produced (Fig. 8).

Here,

Class 1: Positive

Class 2: Negative.

Definitions of the terms:

Positive (*P*): Observation is positive.

Negative (*N*): Observation is not positive.

True Positive (TP): Observation is positive, and is predicted to be positive.

True Negative (TN): Observation is negative, and is predicted to be negative.

	<i>Class 1 Predicted</i>	<i>Class 2 Predicted</i>
<i>Class 1 Actual</i>	TP	FN
<i>Class 2 Actual</i>	FP	TN

Fig. 8 Confusion matrix

Out[45]:		
Predicted attacks	0	1
Actual attacks		
0	9499	212
1	2830	4630

Fig. 9 Dos attack

Out[46]:		
Predicted attacks	0	2
Actual attacks		
0	2337	7374
2	212	2209

Fig. 10 Probe attack

Out[47]:		
Predicted attacks	0	3
Actual attacks		
0	9707	4
3	2573	312

Fig. 11 R2L attack

Out[48]:		
Predicted attacks	0	4
Actual attacks		
0	9703	8
4	60	7

Fig. 12 U2R attack

False Positive (FP): Observation is negative, but is predicted positive.

False Negative (FN): Observation is positive, but is predicted negative.

Confusion metrics for all attack types:

See Figs. 9, 10, 11 and 12.

7 Conclusion and Future Work

Keeping in mind the requirement of security in today's hazardous world, the proposed model is implemented. World is globally connected through Internet. With the

involvement of Internet, security is necessary. User relies on security. It is important to win user trust. For various reasons, we have tried to fulfil users need to our best. With the increasing trend in private data, wide usage of machines, it is necessary to maintain confidentiality.

The world today is globally connected through Internet and it is an era of automation. With the advancement in technology, there is also the need to safeguard the information and maintain information integrity. With the increasing trend in personalisation and wide usage of computers, it is necessary to maintain machine's performance and life. In this chapter, we have tried to drive exposure upon security techniques.

NIDS is the system that immortally keeps monitoring the traffic in the network and alerts the user when a malicious intrusion is detected. The system solution provided above mainly falls into classification algorithms of machine learning. The system is implemented using decision tree algorithm and NSL-KDD dataset is used for the experimentation purpose. The output is represented in the form confusion metrics and graph. Performance of NIDS is constituted in terms of Accuracy and Precision.

Lastly, after successful detection of malicious attacks on the system, a prevention system can be implemented in the later part. So that malicious attack intrusion is prohibited by the prevention system before the user even realises that the system is in danger, is the extended scope of the project. There is no need for the user to monitor the system continuously. With extensive efficiency, prevention can be done unnoticed.

This research work can be extended as follows:

- Other machine learning algorithms can be used to get better Accuracy and Precision results.
- Detection is performed by this system also prevention can be done using regression techniques, so the user can carry on the work without any interruptions.
- Other data packets could be used to increase detection rate of variety of attacks in the incoming traffic.

References

1. Chie-Hong L, Yann-Yean S, Yu-Chun L, Shie-Jue L (2017) Machine learning based network intrusion detection. In: IEEE 2nd international conference on computational intelligence and applications, pp 79–83
2. Constantinos K, Kambourakis G (2014) Intrusion detection in wireless networks using nature inspired algorithms. In: University of Aegean (Doctoral thesis), pp 93–96
3. Amarnath P, Jyoti V (2015) Classification rule and exception mining using nature inspired algorithms. Int J Comput Sci Inf Technol 6(3):3023–3030
4. Obinna I, Ihab D, Tarek S (2016) Distributed network intrusion detection systems: an artificial immune system approach. In: IEEE first international conference on connected health: applications, systems and engineering technologies (CHASE), pp 101–106

5. Sanjay K, Ari V, Timo H (2017) Machine learning classification model for network based intrusion detection system. In: 11th international conference for internet technology and secured transactions (ICITST), pp 242–249
6. Ali M (2018) Computer network intrusion detection using various classifiers and ensemble learning. In: 26th signal processing and communications applications conference (SIU), pp 1–4
7. Jabbar MA, Shirina S (2016) Intelligent network intrusion detection using alternating decision trees. In: 2016 international conference on circuits, controls, communications and computing (I4C), pp 1–6
8. Nerijus P, Juozas A (2014) Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset. In: 2017 open conference of electrical, electronic and information sciences (eStream), pp 1–5
9. Noureldien A, Izzedin MY (2016) Accuracy of machine learning algorithms in detecting DoS attacks types. *Sci Technol* 6:89–92
10. Preeti A, Sudhir KS (2015) Analysis of KDD dataset attributes—class wise for intrusion detection. *Proced Comput Sci* 57:842–851
11. Amrish T, Preeti S (2015) An efficient approach for intrusion detection in reduced features of KDD99 using ID3 and classification with KNNGA, pp 445–452
12. Fernando EB, Alex AF, Colin GJ (2012) Inducing decision trees with an ant colony optimization algorithm. *Appl Soft Comput* 12:3615–3626
13. Cetin K, Oktay Y, Sinan A (2016) Performance analysis of machine learning techniques in intrusion detection. In: 24th signal processing and communication application conference (SIU), pp 1473–1476
14. Swapnil U, Sanyam S (2018) Analysis of heuristic based feature reduction method in intrusion detection system. In: 5th international conference on signal processing and integrated networks (SPIN), pp 717–720
15. Negandhi P., Trivedi Y., Mangrulkar R. (2019) Intrusion Detection System Using Random Forest on the NSL-KDD Dataset. In: Shetty N., Patnaik L., Nagaraj H., Hamsavath P., Nalini N. (eds) Emerging Research in Computing, Information, Communication and Applications. Advances in Intelligent Systems and Computing, vol 906. Springer, Singapore

Chapter 26

Ascent of Pre-trained State-of-the-Art Language Models



Keval Nagda, Anirudh Mukherjee, Milind Shah, Pratik Mulchandani and Lakshmi Kurup

1 Introduction

Applications involving natural language processing (NLP) have become significantly easier, faster and more economical to build these days largely due to the immense computational power in our hands which can be used to develop pre-trained models that can be used to come up with a number of solutions to a wide array of tasks via transfer learning and fine-tuning. Language modeling is a fundamental problem of NLP that needs to be addressed appropriately in order to come up with proper solutions for a number of NLP tasks. A language model works toward estimating the probability of various linguistic units such as characters, words, sentences and paragraphs. In order for such words to be processed by these models, they need some form of numeric representation. Using traditional embeddings wherein such words are represented as vectors has always been a popular approach but they have a major limitation. They fail to consider the context behind a word and assume its meaning to be the same across all sentences. For example, the word “rose” in “Debbie rose to give her speech.” and “Debbie gave a rose to her mother.” have different meanings,

K. Nagda · A. Mukherjee · M. Shah (✉) · P. Mulchandani · L. Kurup

Department of Computer Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai, India

e-mail: mlndshh63@gmail.com

K. Nagda

e-mail: knagda008@gmail.com

A. Mukherjee

e-mail: anirudhm130@gmail.com

P. Mulchandani

e-mail: pratikm1910@gmail.com

L. Kurup

e-mail: lakshmidkurup@gmail.com

but are represented via the same vector in such traditional embedding models which are thus trained on shallow representations. This is where we bring in pre-trained state-of-the-art models which have been highly successful in tackling this issue.

Language modeling has shown to capture many aspects of language relevant for downstream tasks such as hierarchical relations, question answering, paraphrasing and machine translation. Such language models have seen countless developments in recent days, the freshest of them being the use of pre-trained models to perform NLP tasks. A few of these models are task-specific, while the others are more generalized. Using such pre-trained models is highly beneficial due to the fact that it can be fine-tuned to solve any specific task doing away the need to train the model from scratch. It thereby reduces the amount of labeled data needed to train it in order to solve that task and also reducing the computational load which leads to faster results and an efficient model. Researchers have used a number of different methodologies in order to pre-train such models, almost all of them being trained on huge datasets. This approach of using pre-trained models leads to the model being able to learn both the lower-level and higher-level features of the language, which leads to a much better performance in almost all standard NLP tasks. In the following sections, we review some of the most popular cutting-edge state-of-the-art models, namely ULMFiT, ELMo, Transformer-XL, BERT and OpenAI's GPT-2.

2 ULMFiT

Computer vision (CV) has achieved a drastic enhancement using transfer learning methods, and this approach can be used to tackle a major set of NLP tasks too. Several state-of-the-art models used to solve CV problems are obtained by fine-tuning of models pre-trained on heavy datasets like ImageNet and MS-COCO.

Early NLP research majorly consisted of the transductive transfer learning. Universal Language Model Fine-Tuning (ULMFiT) [1] introduces an inductive transfer learning approach with different fine-tuning methods which tackles the problem of overfitting of language models on small datasets and can be used for any NLP task.

Pre-trained word embeddings [2] are also used in most of the state-of-the-art models, but this technique adds a constraint where models using pre-trained embeddings consider them as fixed parameters. The ULMFiT proposes various fine-tuning approaches such as discriminative fine-tuning, slanted triangular learning rates and gradual unfreezing techniques to maintain long-term dependencies in the model.

2.1 *Working*

ULMFiT consists of two major tasks:

General-domain LM pre-training: Pre-training the language model (LM) on a large general-domain corpus helps with generalization even with small datasets.

Target task LM fine-tuning: Fine-tuning the pre-trained LM model on the target task using the following proposed novel techniques.

- *Discriminative fine-tuning:* It is observed that different layers of a model are activated according to different types of information [3], and thus, discriminative fine-tuning should be applied in order to tune each layer with different learning rates rather than using the same learning rate for all layers of the model.
- *Slanted triangular learning rates:* To obtain task-specific feature learning, slanted triangular learning rates (STLR) can be used by the model to initially converge to an appropriate region in the parameter space by a linear increase in learning rates and then optimize its learning rate by refining the parameters.
- *Gradual unfreezing:* In this approach, instead of fine-tuning all the layers of a model in a single shot, a layer-wise unfreezing and training occur. Starting from unfreezing the last layer as this layer contains the least general knowledge [3], fine-tuning of all unfrozen layers for one epoch is carried out. Then, the next lower frozen layer is unfrozen and the process is repeated until all the layers of the model are fine-tuned as required.

2.2 Dataset

Six widely studied datasets, with a varying number of documents and varying document lengths, are used. The datasets are TREC-6 (6 classes, 5.5 k examples), IMDb (2 classes, 25 k examples), Yelp-bi (2 classes, 560 k examples), Yelp-full (5 classes, 650 k examples), AG (4 classes, 120 k examples) and DBpedia (14 classes, 560 k examples). The language model is pre-trained on Wikitext-103 [1].

2.3 Architecture

ULMFiT consists of the state-of-the-art language model AWD-LSTM, as a base LSTM model with tuned dropout hyperparameters. It consists of an embedding size of 400 with three layers, 1150 hidden activations per layer and a backpropagation through time (BPTT) batch size of 70. A dropout of 0.4 is applied to layers, 0.3 to RNN layers, 0.4 to input embedding layers, 0.05 to embedding layers and 0.5 of weight dropout to RNN hidden-to-hidden matrix. Adam optimizer is used with $\beta_1 = 0.7$ instead of $\beta_1 = 0.9$ and $\beta_2 = 0.99$. A batch size of 64 with a base learning rate of 0.004 and 0.01 for fine-tuning the LM is chosen. The LM is fine-tuned for 15 epochs on small datasets while for 50 epochs on large datasets.

3 ELMo

Pre-trained word representations [2, 4] are very crucial elements in many state-of-the-art language models. However, these elements lack the representation of complex word characteristics (syntax and semantics) and are often used across varied linguistic contexts. ELMo proposes a deep contextualized word representation approach that tackles various challenges faced by traditional word embeddings. In ELMo, vectors are extracted from a bidirectional LSTM model coupled with language model (LM) that is pre-trained on a huge textual corpus. These context-dependent representations derived from the internal layers of the deep bidirectional language model (biLM) outperform models which use a neural machine translation encoder model to extract contextualized representations.

3.1 Working

ELMo can be used for any target-specific task by just combination of intermediate layers. Initially, the supervised target model forms context-independent token representation using pre-trained word embeddings and later context-sensitive representations are formed. A set of $2L + 1$ representations are computed by a L-layer biLM for each token. Usage of ELMo for any downstream supervised model can be done by concatenating all the layers into a single vector after freezing the weights of the biLM. Activations of each layer have variance in distribution, and thus, layer normalization aids if applied before weighting. A moderate amount of dropout addition is observed to be beneficial in few scenarios.

3.2 Dataset

The bidirectional language model (biLM) is pre-trained on 1B Word Benchmark and can be fine-tuned for most of the target-specific downstream tasks [4].

3.3 Architecture

The pre-trained bidirectional language model (biLM) is similar to the architecture in [5] with an additional connection between LSTM layers of the model. Embedding and hidden dimensions are halved as compared to the CNN-BIG-LSTM model [5].

The model consists of 2 biLSTM layers with 4096 units and 512-dimensional projections where the first and second layers are connected. The model is trained

for 10 epochs. The context-independent representation follows 2048 character n-gram convolutional filters, two highway layers and a linear 512 projections. The traditional word embedding methods provide only one layer of representation for tokens, whereas the biLM provides three layers of representations for each input token.

4 Bert

BERT, or Bidirectional Encoder Representations from Transformers [6], is a language representation model developed by Google’s AI team which was able to achieve state-of-the-art performance in a number of natural processing tasks. Unlike traditional models [7] which take the previous n tokens for its prediction, BERT takes into consideration both the previous and the next token. BERT thus takes into consideration both the left and right contexts of a token in all of its layers, unlike the standard models which only take the left context into consideration. BERT is also trained on the next sentence prediction in order to handle tasks that require a certain degree of knowledge about the relationship between sentences. BERT makes use of a fine-tuning approach which introduces a minimum number of task-specific parameters and is trained on downstream tasks by fine-tuning all of the pre-trained parameters. Thus, with a little bit of fine-tuning, the BERT model can be used to create state-of-the-art models for a number of tasks.

4.1 Working

BERT makes use of a “masked language model” (MLM) pre-training objective in order to do away with the unidirectional constraint of traditional language models. MLM randomly masks or hides some of the tokens in the input and predicts the masked word based only on its context. The BERT framework includes two steps, namely pre-training and fine-tuning.

In the pre-training stage, BERT is trained on unlabeled data over a large number of different tasks. Pre-training of BERT occurs in two steps—masked LM and next sentence prediction (NSP).

Masked LM. In this step, a certain percentage of input tokens are masked at random and then the original vocabulary of these masked tokens is then predicted.

NSP. This step ensures that the model understands the relationships between sentences. To accomplish this, two sentences A and B are chosen for each pre-training example, 50% of the time B is the actual sentence after A and 50% of the time it is a random sentence from the corpus.

For the fine-tuning step, the task-specific inputs and outputs are plugged into the model and all parameters are fine-tuned end to end to obtain the fine-tuned model.

4.2 Dataset

BERT uses the BookCorpus (800 M words) and English Wikipedia (2,500 M words) [6]. Only the text passages are extracted for the Wikipedia corpus, thereby ignoring lists, tables and headers.

4.3 Architecture

BERT makes use of a unified architecture along all tasks which basically means that its architecture remains the same irrespective of the task. BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described in [8]. The Transformer architecture is a model that does not use recurrent connections at all and uses attention over the sequence instead. Unlike RNNs, the Transformer architecture cannot take the order of the inputs into account. Hence, to overcome this problem, BERT makes use of positional embeddings to express the position of a word in a sentence.

BERT was developed on two model sizes, namely BERT base and BERT large. BERT base has 12 layers or Transformer blocks, 768 hidden layers, 12 self-attention heads and 110 M parameters in total, whereas BERT large has 24 layers or Transformer blocks, 1024 hidden layers, 16 self-attention heads and 240 M parameters. BERT base was chosen to have the same model size as OpenAI's GPT for comparison purposes.

BERT makes use of bidirectional self-attention, whereas GPT uses constrained self-attention and thus can only take into consideration the left context of a certain token, whereas BERT because of its bidirectionality can make use of both the left and right contexts. BERT and GPT are similar in terms of model architecture apart from the attention masking.

5 OpenAI's GPT

GPT, by OpenAI, was released on June 2018. GPT is a task-agnostic system with very good results on various language tasks [7]. GPT is an amalgamation of Transformers and unsupervised pre-training. GPT shows that large gains on various tasks like textual entailment, question answering, document classification and semantic similarity assessment can be seen by generative pre-training of a LM on a varied corpus of unlabeled text, continued then by discriminative fine-tuning on each specific task. A Transformer [8] model is first trained on large amounts of data in an unsupervised manner, with language modeling as a training signal. This model is then fine-tuned on smaller supervised datasets, so that it can solve specific tasks.

GPT2, the successor to generative pre-training (GPT) was announced by OpenAI in February of 2019 and as they have put it, is a direct scale-up of GPT, with more than $10 \times$ the parameters and trained on $10 \times$ the amount of data. As noted by Radford et al. [9], ML systems do great at tasks that they are trained for by a large number and variety of datasets, high-capacity models and supervised learning.

5.1 Working

The goal of GPT is to learn a common representation that carries forward with little adaptation to a varied variety of tasks. The setup does not need the target tasks to be in the same domain as the unlabeled text. A language modeling objective is first used on the unlabeled data to learn the initial parameters of the neural network model. Later, these parameters are adjusted to target tasks using respective supervised objectives. Using the Transformer model gives the choice of more structured memory to handle long-term dependencies in text compared to other options like recurrent networks. During the transfer, task-specific input adaptations are used which process structured text as a single continuous sequence of tokens. These adaptations allow for effective fine-tuning with very little to no changes to the architecture of the pre-trained model.

GPT2 works on similar principles as GPT, with modifications in the input representation. GPT2 uses byte pair encoding (BPE), which interpolates between word-level inputs for frequent word sequences and character-level inputs for infrequent word sequences. Since BPE may include many variations of common words, they prevent BPE from merging across character categories for any byte sequence. A special case is added for spaces which significantly improves the compression efficiency while adding only minimal fragmentation of words across multiple vocab tokens. This input representation combines the empirical benefits of word-level LMs with the generality of byte-level approaches.

5.2 Dataset

GPT used various datasets for the different tasks. For natural language inference, SNLI, MultiNLI, Question NLI, SciTail, RTE datasets were used. For question answering, RACE, Story Cloze were used, for sentence similarity, MSR Paraphrase Corpus, Quora Question Pairs, STS Benchmark were used and for classification, Stanford Sentiment Treebank-2, CoLA were used [7].

The approach used for GPT2 involved building as big and varied of a dataset as possible so that texts from varied domains and contexts could be gathered, by scraping all outgoing links from Reddit with Karma score of more than three was used. The final dataset known as WebText after de-duplication and heuristic cleaning contains over 8 million documents and over 40 gigabytes in text. All Wikipedia links

were removed since it is a common source for other datasets which would have led to complications including overlapping training data.

5.3 *Architecture*

A Transformer [8]-based architecture is used for the LMs. The model adopts the details of the OpenAI GPT model [7] with a few changes. Layer normalization is moved to the input of each sub-block, similar to a pre-activation residual network [10], and additional layer normalization was added after the final self-attention block. The initialization is changed, where a residual path with depth model is used which accounts for the accumulation. The weights of residual layers are scaled at initialization by a factor of $1/\sqrt{N}$ where N is the number of residual layers. The vocabulary is expanded to 50,257. The context size is also increased from 512 to 1024 tokens, and a larger batch size of 512 is used.

6 XLNet

Two of the most commonly used language models are BERT [6] and Transformer-XL [11], which use techniques such as autoencoding (AE) and autoregressive (AR) language modeling, respectively, during the pre-training process.

BERT, the earlier SoTA model, allows for the use of bidirectional context to construct sequences from a given input, which helps reduce the risk of context fragmentation. However, it requires certain tokens to be converted into an artificial symbol [MASK] in order to predict these tokens during training. Due to this mechanism, all of the tokens in the real data cannot be used, as some of them are masked in the input. This may cause discrepancies, and BERT assumes that the predicted tokens are completely independent of each other and ignores their joint probability.

To overcome this shortcoming of BERT, XLNet was autoencoding (AE) proposed, which uses a generalized AR method, similar to that used in Transformer-XL.

6.1 *Working*

XLNet maximizes the log-likelihood of a sequence for all the available permutations of the factorization order. It captures bidirectional context and learns from it. To ensure that the input sequence order is kept intact, it uses positional encodings.

Since it is an AR-based model, it does not use the data corruption mechanism that BERT uses, thus neglecting the independence assumption. This helps increase its accuracy, as 100% of the input data is being used as context.

The mechanisms borrowed from Transformer-XL are:

- *Relative positional encoding*, which helps improve the performance during pre-training. Since this encoding is sequential, it allows for XLNet’s permutation operation.
- *Segment recurrence mechanism*, which helps cache the previous segment and uses it as context for the next segment.

6.2 Dataset

Similar to BERT, XLNet was pre-trained using English Wikipedia dataset (13 GB of plain text), as well as CommonCrawl, Giga5 and ClueWeb 2012-B datasets [11]. The large variant of the model has a sequence and memory length of 512 and 384, respectively. The model was trained on 512 TPU v2 chips for 500,000 steps for about two and a half days.

6.3 Architecture

A two-stream self-attention mechanism is implemented in order to overcome the requirements presented by the permutation system; i.e., only the position, not the content, of a token must be used during predictions, and previous tokens should also be encoded to preserve contextual information.

The last tokens are predicted in a factorization order, in order to make convergence faster. While trying to maximize the log-likelihood of the target sequence, only a select few (K) tokens are selected. The unselected tokens’ query representations are not calculated, which helps speed up computation (Table 1).

7 Results

7.1 GLUE

The General Language Understanding Evaluation (GLUE) [12] benchmark is a collection of resources which includes a certain set of tasks based on which a certain natural language model can be evaluated. GLUE consists of a set of nine different sentences or sentence-pair based tasks which get evaluated over a diagnostic dataset. Models are thus evaluated as per their average accuracies across all tasks. Currently, the XLNet model stands at the top of the GLUE leaderboard with an average GLUE score of 88.4 beating previous state-of-the-art models in a number of tasks.

Table 1 Observed experimental scores

System	MNLI (m/mn)	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	SNLI	SQuAD (EM/F1)	GLUE
BiLSTM + ELMo + Attn	74.1/74.5	79.8	63.1	58.9	90.4	84.4	33.6	74.2	88.7	78.5/85.8/33	70.0
BERT	86.7/85.9	92.7	72.1	70.1	94.9	89.3	60.5	86.5	–	87.4/93.2	80.5
GPT	82.1/81.4	88.1	70.3	56.3	91.3	82.3	45.5	82.0	89.9	–	72.8
XLNet (Multi-task ensembles on test)	90.2/889.7	98.6	74.2	86.3	96.8	93.0	67.8	91.8	–	86.35/89.13	88.4

7.2 SNLI

The Stanford Natural Language Inference (SNLI) [13] corpus is a large collection of natural language inference problems. Each problem exists as a pair of two sentences, the premise and the hypothesis, along with a label. The model thus has to predict the label correctly based on the premise and the hypothesis. Based on the models reviewed in this paper, OpenAI’s GPT has attained the highest accuracy on the SNLI corpus with a score of 89.9.

7.3 SQuAD

The Stanford Question Answering Dataset (SQuAD) [14] is a set of crowdsourced question–answer pairs obtained from Wikipedia articles. The model is provided with a question and a passage from Wikipedia, and the goal of the model is to predict the answer text given in the passage. The exact match (EM) score and the F1 score are used for evaluation. Among the models reviewed in this paper, BERT stands at the top with the highest EM and F1 score at 87.4 and 93.2, respectively.

8 Conclusion

Language models can be used to solve a number of trivial and non-trivial NLP tasks with state-of-the-art accuracies which have started to exceed the accuracy with which humans can solve the same set of tasks. These tasks involve text generation, machine translation and question answering to name a few. We have reviewed a number of ensemble multi-task learning models which have given state-of-the-art performances in a number of benchmark tests and have set the bar higher with each new discovery. All the performance measures of the models reviewed in this paper belong to a common subset of tests performed on the same by the original authors of the respective models and duly cited in their respective papers. Each model is evaluated based on the scores obtained in this subset of common tests. The most recent model, the XLNet, has surpassed the previous state-of-the-art scores on 18 tasks. From bidirectionality to self-attention, we have seen language models evolve in a number of innovative ways. With every new model besting the previous state-of-the-art scores, this trend of advancements shall not stop anytime soon.

References

1. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification
2. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems
3. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in neural information processing systems, pp 3320–3328
4. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: EMNLP
5. Jozefowicz R, Vinyals O, Schuster M, Shazeer N, Wu Y (2016) Exploring the limits of language modeling. CoRR abs/1602.02410
6. Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
7. Radford Alec, Narasimhan Karthik, Salimans Tim, Sutskever Ilya (2018) Improving language understanding with unsupervised learning. Technical report, OpenAI
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I: Attention is all you need. In: Advances in neural information processing systems, pp 6000–6010
9. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners
10. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: European conference on computer vision. Springer, pp 630–645
11. Dai Z, Yang Z, Yang Y, Cohen WW, Carbonell J, Le QV, Salakhutdinov R (2019) Transformer-xl: attentive language models beyond a fixed-length context. arXiv preprint [arXiv:1901.02860](https://arxiv.org/abs/1901.02860)
12. Wang A, Singh A, Michael J, Hill F, Levy O, Bowman S (2018) Glue: a multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the EMNLP workshop BlackboxNLP: analyzing and interpreting neural networks for NLP, pp 353–355
13. Bowman SR, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. In: Proceedings of the Conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics
14. Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016) Squad: 100,000 + questions for machine comprehension of text. In: Proceedings of the conference on empirical methods in natural language processing, pp 2383–2392

Chapter 27

Syt-AJ: Treating Lazy Eye Using Virtual Reality



Tejas Ved, Jay Chauhan and Neha Katre

1 Problem Definition

Amblyopia or commonly known as “Lazy Eye” is a vision development disorder wherein the vision of the normal eye overpowers the vision of the affected eye and the affected eye is unable to achieve a visual acuity of a normal eye [1]. This condition is normally observed in a single eye and in rare cases affects both the eyes. The disorder is caused due to poor coordination between the brain and the eye. Some of the physical factors involved in cause of amblyopia are poor eye alignment, irregular eye shape or irregularity in the visual acuity [2]. Because the brain does not develop the ability to see clearly in one or both eyes, the eyesight cannot be improved with glasses alone.

This disorder commonly occurs in children and younger adults. Early detection helps in boosting the success rate of treatment. The system we designed for treating amblyopia includes therapy as a series of exercises and activities that help a person improve their visual skills. Vision therapy helps restore a person’s binocular vision, which is the root cause of the condition.

Our system overcomes the shortcomings for the existing the solutions. Our approach defines deployment of a VR environment application which can be accessed by anyone and everyone throughout the world. This can eliminate the availability factor for the amblyopia treatment.

T. Ved (✉) · J. Chauhan · N. Katre

Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

e-mail: tejas141096@gmail.com

J. Chauhan

e-mail: chauhan.jay99@yahoo.in

N. Katre

e-mail: neha.mendjoge@djsce.ac.in

Considering the cost factor for the technique, instead of available commercial VR sets and controllers switching them with available Google Cardboard and a Bluetooth controller for interacting with the VR environment, this significantly reduces the cost factor of the technique. This helps the individuals who cannot afford surgeries or costly VR software to receive proper treatment for amblyopia. But, the usage as discussed should be carried out under proper guidance or an ophthalmologist.

2 Literature Survey

Detection of amblyopia or rather symptoms of this disorder for those who suffer from mild form is not aware until they get tested, since the normal eye views the environment ordinarily.

There are a range of active and passive therapies in terms of the methods used to treat amblyopia. The diagnosis of amblyopia is done by identifying the eye with lower visual acuity and correcting it to the good eye. Treatment is continued as long as the player is able to perceive depth and see objects and the environment properly. It is not suggested to continue eye patch for more than 6 months if no improvement is recorded [3]. Deprivation amblyopia is treated by removing the opacity as soon as possible by eye surgery, followed by patching the good eye to boost the use of the amblyopic eye [4]. The earlier the amblyopic eye is detected and treatment is initiated, the easier and faster the treatment is and the less subjectively stressful. Also, the chance of achieving 20/20 vision is greater if treatment is initiated early [5].

The following is the list of available treatments for amblyopia [6]:

- Eye patches
- Eye surgery
- Vision therapy software
- Vivid vision gear
- Lazy eye shooter.

The above approaches are discussed below while considering their advantages and disadvantages.

2.1 Eye Patches

Based on the principle on which amblyopia occurs, occlusion therapy is a passive and effective method to treat amblyopia. With eye patches, the good eye is covered, so that maximum processing is done by the lazy eye as shown in Fig. 1 [7].

The X ganglion cells in the cupped area of retina center and the dual-eye drive cells in vision cortex of the brain are stimulated to growth [8]. The main objective of using patches is to compel the lazy eye to concentrate on the objective. Covering

Fig. 1 Eye patches of amblyopia



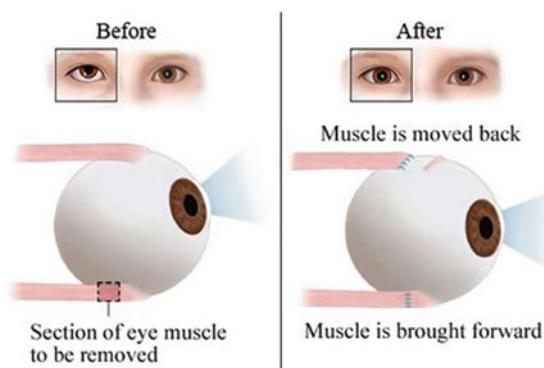
the eyes is a simple solution, but it is more effective than traditional methods. This method needs to be carried out in supervision of adults for children and can often cause uneasiness due to its anesthetic way.

2.2 Eye Surgery

Eye surgery is performed to strengthen the eye muscles which help in free movement in eye as shown in Fig. 2 [9]. This is usually performed in case of strabismic amblyopia [10]. Amblyopia is often commonly confused with strabismus. The idea of lazy eye generally makes people think of misaligned eyes or wandering eyes which is strabismus.

Strabismus is the most common cause of amblyopia. To avoid double vision caused by poorly aligned eyes, the brain ignores the visual input from the misaligned eye, leading to amblyopia in that eye (the “lazy eye”). This type of amblyopia is called strabismic amblyopia [1]. It can only be treated when the patient is a child.

Fig. 2 Lazy eye surgery



The younger it is detected and treated the better. In fact, after age 6, the success rate of treatment goes way down [11].

2.3 *Vision Therapy Software*

Vision therapy software is a computer program that helps detect the symptoms of amblyopia. Amblyopia is a visual defect that glasses alone cannot fix. It involves a problem in the vision as well as the eye muscles which work out of sync with each other. The muscles of the eyes must work together toward proper focus and rest as and when required. The uncoordinated cause decreased visual acuity. Vision therapy software consists of a series of eye exercises done toward the computer, like a computer game as shown in Fig. 3 [12]. The initial tasks are simple. You need to find hidden 3D boxes across the screen using follow arrows on the screen which helps in increasing the visual acuity and in identification of movements of images. The computer dynamically increases the level of the game based on the player's improvement [13]. Each Vision therapy program is tailored to one's own personal binocular problem.



Fig. 3 Vision therapy software (VTS)

2.4 Lazy Eye Shooter

Experiments show that action video games have displayed improved vision as well as coordination in normal as well as amblyopic individuals. Playing action video games results in a range of improved spatial and temporal visual functions including visual acuity [14].

The lazy eye shooter has a dichoptic display-based architecture. The game involves a 3D stereographic display with faded image shown to the normal eye and the complete game environment shown to the amblyopic eye. Points and penalty in the game are partially based on focus with the amblyopic eye of the patient [14].

2.5 Vivid Vision Gear

Vivid vision treats lazy eye using a special mode called “diplopia mode” within which the game shows different image to the lazy eye compared to the normal eye, forcing the eyes to work in sync in order to get through to the next level as shown in Fig. 4. It uses a simple 3D brick breaker game in which the paddle is controlled by gestures on which the ball bounces and the player earns points based on the level of bricks [15].

In the game, paddle is visible only to the strong eye, the ball is visible only to the weak eye, and the bricks are focused on in the weak eye and dimmed in the strong eye. Using this technique, the game is able to break through the player’s suppression. The game limits the amount of information visible to each eye and thus requires the

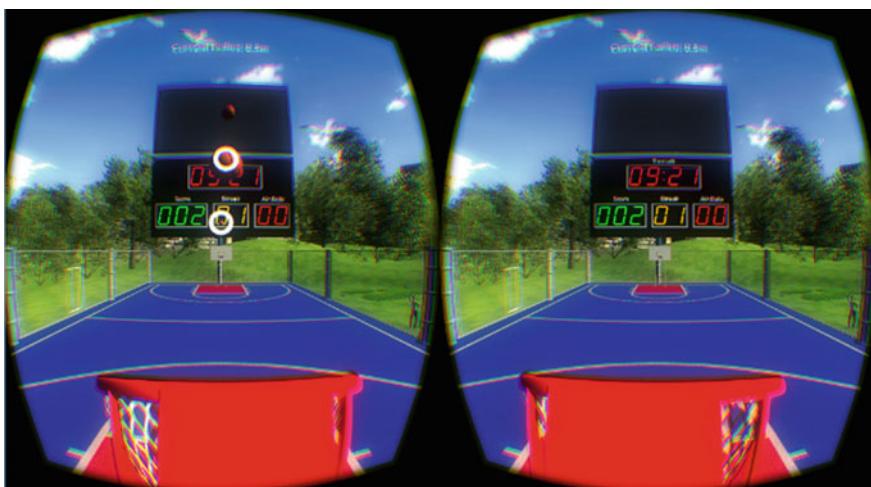


Fig. 4 Vivid vision hoopie

player to use both of them together and to make the brain understand the position of the objects in the game and merge the two sets of information available visually to develop a single frame or picture of the game environment. The fast-moving ball shown only to the weaker eye allows the patient to exercise the muscles of the weaker eye more than the normal eye which has the easier task of tracking the paddle and is not required to be more active. This allows the patient to perceive depth and 3D objects and as the treatment progresses allow the player to gain 3D vision in normal conditions [15].

3 Comparison of Existing Technologies

A comparison of the existing technologies is given in Table 1.

As the above table indicates, all the above technologies have their own merits based on different parameters. Eye surgery can be only performed on children below 16 years of age. The cost factor is also high, and time to recover on the other hand is definite. Complications can happen during surgeries, but the probability is very less. Eye patches again only work for children below 12 years of age. In most amblyopic cases, the age is a determining factor. Cost factor is very less, but the recovery time is again dependent on the severity of amblyopia. Use of patches can cause strain on the eyes of the children. Vision therapy systems do not have any age limit for usage, so as the other VR-based approaches for treatment. But, these commercial software lie on the higher end of cost factor. The recovery time is utility based. Usage of these

Table 1 Comparison of technologies

Treatment criteria	Eye surgery	Eye patch	VTS	Vivid vision gear	Lazy eye shooter
Usage	Children below age of 16	Children below age of 12	Everyone	Everyone	Everyone
Cost	High	\$10 for 30 patches	\$1000–\$7000 with \$152/visit to the doctor	\$2000–\$2500	Proprietary software of HTS (VTS).
Time taken in treatment	2–4 weeks for muscle movement	Usability based	Usability based (causing negative effects as well)	Usability based	Usability based
Side effects/complications	Complications in surgery	Strain in the eye	Headache, sore eyes	VR sickness, radiation due to continuous usage of system	Radiation due to continuous usage of system

devices is generally advised under expert guidance such as ophthalmologists. All these VR-based software are proprietary and are charged session based. Each doctor session costs around hundreds of dollars. This price range is not feasible for every individual. Also, the necessary gear required to use these software is costly. The overall affordability of these treatments for an individual is very high. This calls in a new technique for treatment which overcomes all the down factors for the exiting techniques.

4 System Architecture

Various components are included in our architecture (Fig. 5):

- User Input from Controller
Takes the input from the user through Bluetooth controller.
- Input Control Mapping
Maps the input to the player as well as the environment.
- Decision Making
This module makes use of algorithms to take required actions in the gaming environment.
- Rendering the Changes in Environment
The models are retrieved through the algorithm and are loaded in the environment.
- Display Output on the HMD
The rendered changes are displayed in the HMD.

4.1 Tools and Technologies

- Blender 2.79 and Unity 2018.2 are used as a base to create our project.

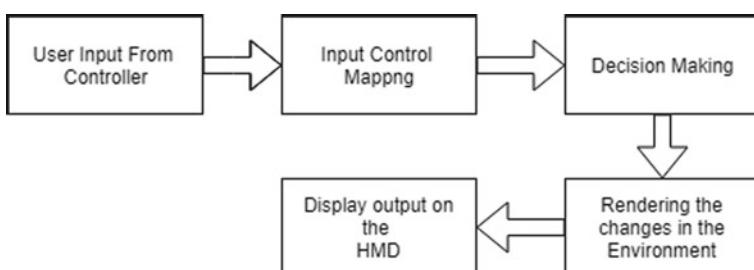


Fig. 5 System architecture

- Unity is a cross-platform game engine developed by Unity Technologies, which is primarily used to develop both three-dimensional and two-dimensional video games and simulations for computers, consoles and mobile devices.
- These open-source software are used for creating the game and uploading it on Google Play Store, so the application is available to all in need. Also, no internet connectivity is required while executing the project.
- Unity and Blender have a very strong and ever-increasing community of freelance developers, thus increasing the exposure and reducing the costs required to get assistance from developers.

5 Implementation

5.1 Referenced Scaling

Scaling the objects on the screen with respect to refractive index error in the eyes is also known as visual acuity. Visual acuity is dependent on optical and neural factors, i.e., (i) the sharpness of the retinal focus within the eye, (ii) the health and functioning of the retina and (iii) the sensitivity of the interpretative faculty of the brain, as shown in Fig. 6.

This specifies an individual's ability to percept depth in the virtual environment and increases the individual's stereovision.

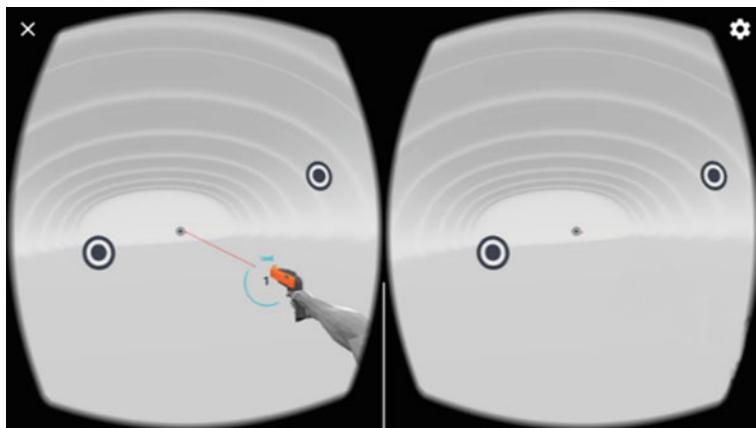


Fig. 6 Target shooter

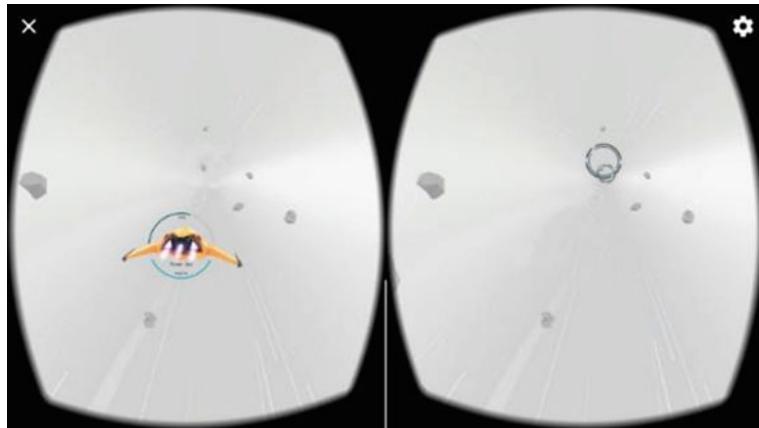


Fig. 7 Space shooter

5.2 Differential Viewing for Each Eye on Screen

It is the ability to adapt the contrast automatically which degrades the screen image which is visible to the player's good eye in a manner that is tailored for each player and the ability to display the game environment to the amblyopic eye for each player to interact with the game environment as shown in Figs. 7 and 8.

6 Testing

- The Syt-AJ system was tested on patients for 20 min (10 min each game), twice a week for 4 weeks.
- It was tested on patients of different age groups, having refractive amblyopia in left, right or both the eyes.
- The system was tested on two types of VR headsets—Google Cardboard and Aura VR (provided by system developers).
- The results obtained after testing are represented in Table 2: Testing result set.

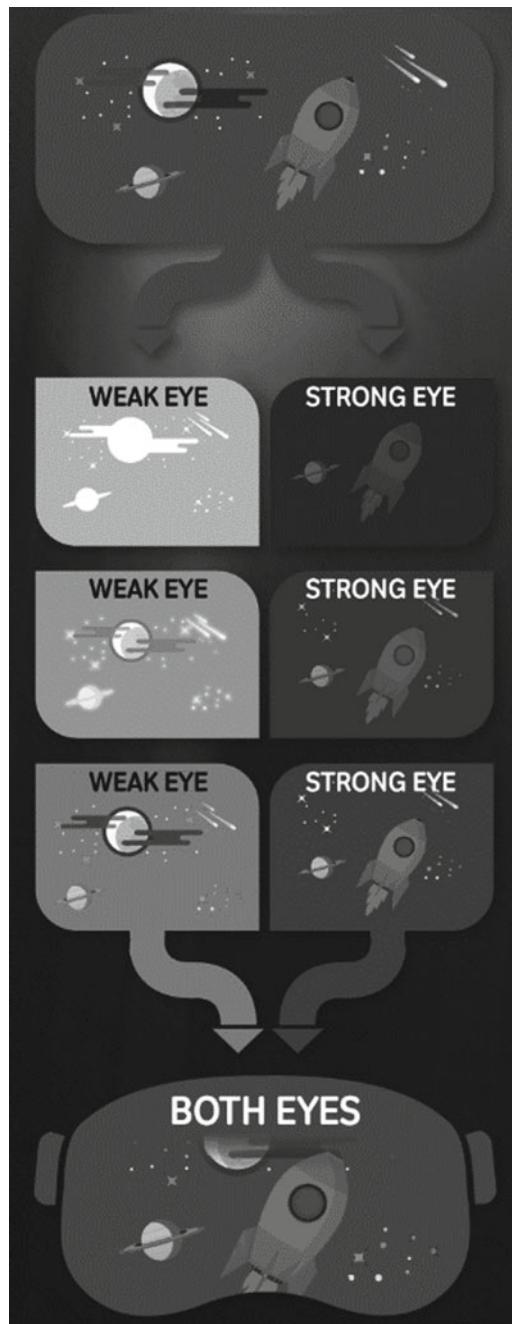


Fig. 8 Graphical representation of the system

Table 2 Training result set

Patient	Age	Amblyopic eye	Healthy eye RE	D sph. D cyl	Amblyopic eye RE	D sph. D cyl	BCVA—before training	BCVA—after training	Stereoacluity—before training	Stereoacluity—after training
1	7	Left	0.125	0.5	-0.5	0.5	0.5	0.3	Nil	140
2	12	Left	1.25	1	4	0.5	0.6	0.6 (personal lens)	140	220 (personal lens)
3	26	Left	-0.75	0	3.25	0	0.2	0.2	300	250
4	45	Right	1.75	0.5	2.75	0.75	0.2	0.2	260	260
5	9	Left	2.5	0	3.5	0.5	0.3	0.2	Nil	400
6	14	Right	-0.625	1	2.5	2	0.4	0.4	Nil	160
7	10	Right	0.25	0.5	1.75	3	1	0.7	40	200
8	18	Right	-2	0.5	-3.5	0	0.5	0.4	Nil	50
9	27	Right	0.5	1.5	2.25	1	0.1	0.1	Nil	20
10	23	Left	0.75	0	1.875	0	0.4	0.3	130	140
11	9	Left	0.5	1	1.5	0	0.6	0.5	Nil	400

RE—Refractive error

BCVA—Best corrected visual acuity

7 Conclusion

There is a direct need for a cost-effective system to deal with lazy eye for people belonging to all age groups. From the literature review, it is clear that the traditional systems are not cost-effective and efficient enough. Our system makes use of virtual reality to treat the lazy eye disorder replacing the existing systems in cost-effective and efficient manner. Also, the results after testing the system show important signs of progress in a patient's vision, indicating the success of the system.

References

1. Amblyopia (Lazy Eye) Explained. All about vision [Online]. Available <https://www.allaboutvision.com/conditions/amblyopia.htm>. Accessed 07 Feb 2019
2. Facts About Amblyopia. National Eye Institute. September 2013. Archived from the original on 27 July 2016. Retrieved 27 July 2016
3. Cunningham ET, Riordan-Eva P Vaughan and Asbury's general ophthalmology (18th edn). McGraw-Hill Medical. ISBN 978-0071634205
4. Holmes JM, Repka MX, Kraker RT, Clarke MP (2006) The treatment of amblyopia. *Strabismus* 14(1):37–42
5. Williams C, Northstone K, Harrad RA, Sparrow JM, Harvey I (2002) Amblyopia treatment outcomes after screening before or at age 3 years: follow up from randomised trial. *BMJ* 324(7353):1549
6. Jefferis JM, Connor AJ, Clarke MP (November 2015). "Amblyopia". *BMJ*. 351: h5811. <https://doi.org/10.1136/bmj.h5811>. PMID 26563241
7. To Patch or Not to Patch: Summary of PEDIG Amblyopia Treatment Studies, OptometryStudents.com, 02 Nov 2018 [Online]. Available <https://www.optometrystudents.com/patch-not-patch/>. Accessed 18 Feb 2019
8. Qiu F, Wang LP, Liu Y, Yu L (2007) Interactive binocular amblyopia treatment system with full-field vision based on virtual reality. 2007 1st international conference on bioinformatics and biomedical engineering
9. Lazy Eye Surgery · Top Eye Specialist · Best Rated NYC Ophthalmologist. Manhattan eye doctors and best rated specialists in NYC [Online]. Available <https://www.eyedoctorophthalmologynyc.com/procedures/lazy-eye-surgery/>. Accessed 20 Feb 2019
10. "Vision and health|CooperVision." [Online]. Available <https://coopervision.com/eye-health-and-vision>. Accessed 14 Feb 2019
11. Lazy Eye Surgery Facts. American Academy of Ophthalmology, 12-Apr-2017 [Online]. Available <https://www.aao.org/eye-health/tips-prevention/lazy-eye-surgery-facts>. Accessed 09 Feb 2019
12. Vivid Vision Launches its VR Vision Therapy System. VRFocus [Online]. Available <https://www.vrfocus.com/2017/10/vivid-vision-launches-its-vr-vision-therapy-system/>. Accessed 22 Feb 2019
13. Home Vision Therapy. SABAL EYE CARE [Online] Available <https://www.sabaleye.com/home-vision-therapy.html>. Accessed 18 Feb 2019
14. Bayliss JD, Vedamurthy I, Bavelier D, Nahum M, Levi D (2012) Lazy eye shooter: a novel game therapy for visual recovery in adult amblyopia. 2012 IEEE international games innovation conference
15. Blaha J, Gupta M (2014) Diplopia: a virtual reality game designed to help amblyopics. 2014 IEEE virtual reality (VR)

Chapter 28

Deep Learning Challenges in Medical Imaging



Vaibhav Saraf, Pallavi Chavan and Ashish Jadhav

1 Introduction

In the era of artificial intelligence and machine learning, medical imaging has benefited with these latest technologies to improve the performance and diagnosis, thereby helping the individuals to take precautionary measures for the diseases. Medical imaging processes create and use images of internal parts of the human body for clinical purposes. The medical science is evolving with the prediction of the diseases. With technological development, medical science is improving and bringing out the latest tools that can help the doctors for better detection and prediction of the diseases. Deep learning significantly contributes to the medical imaging in developing such predictive tools. Over the past few years, systems have been built by researchers for automated medical image analysis. Initially, this task of processing and analyzing medical images was done by low-level pixel processing and mathematical operations to construct complex rule-based intelligent systems. After the inception of tools in supervised learning techniques, new systems are being developed and trained using historical data to extract features. The extraction of discriminant features with minimal or zero noise is a pivotal step in the model of such systems [1, 2]. By analyzing the generic variations in the images, the doctors and researchers are able to enhance

V. Saraf · P. Chavan (✉) · A. Jadhav

Department of Information Technology, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai, MH, India

e-mail: pallavi.chavan@rait.ac.in

V. Saraf

e-mail: sarafvaibhav1070@gmail.com

A. Jadhav

e-mail: ashish.jadhav@rait.ac.in

their ability to predict and accurately diagnose the disease. Artificial intelligence (AI) and machine learning (ML) techniques have developed rapidly in the past few years. ML techniques were composed of some traditional algorithms like support vector machine (SVM), general adversarial network, supervised and unsupervised neural network and convolutional neural network (CNN). Supervised and unsupervised are broad categories of artificial neural networks. Deep learning algorithms are composed of multiple levels of abstractions which transform the raw input data to outputs by learning higher-level features. These deep learning algorithms show promising performance in image processing, image interpretation, image segmentation and many other areas of medical domain [3].

2 Deep Learning Algorithms Used in Medical Imaging

2.1 *Supervised Learning Methods*

Deep learning systems are considered as simplified mathematical models having functions similar to human nervous system. However, neural network can learn new associations, new functional dependencies and new patterns. Beside that, the important advantage of neural network is their adaptability. Neural network along with learning ability are adaptive in nature. Learning allows the network to gain knowledge of new patterns, while adaptation is different with respect to learning in the sense that it acquires every change suggested by learning methodology. Neural networks automatically adjust their weights during learning. Supervised learning has a supervisor. Network output in this category is associated with two types of responses, viz. desire response produced by the supervisor and actual response produced by the network. Learning process repeats till the actual response reaches to desire response. Weights are adapted in each iteration. A learning process terminates for the constant weights. Iteration is referred as a learning step. For n number of patterns in the input set, n learning steps are performed. For constant weights, the error signal reduces to zero. Widely used algorithms in this category are: Hebbian learning, perceptron learning, delta rule learning. Supervised learning scenario is shown in Fig. 1.

2.1.1 Convolutional Neural Network

The convolutional neural networks (CNNs) are suitable to perform complex image processing and recognition tasks for two-dimensional and three-dimensional structures of the organs. CNN is a type of network architecture formed with multiple layers of convolution. The convolutional neural networks are convenient for the image recognition tasks due to their local spatial relationships, i.e., neighboring pixels. The distinctive feature of preservation of relations of local images, while performing

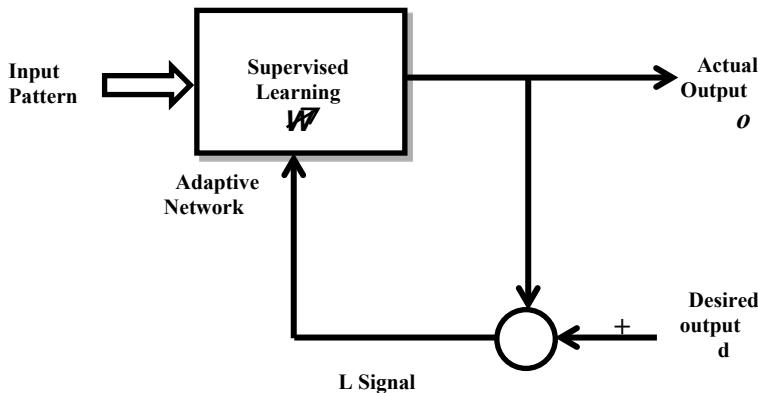


Fig. 1 Supervised learning scenario

dimensionality reductions minimizes the number of parameter computation increasing the overall efficiency. It is a really useful advantage in designing systems for medical use, where some functionalities like MRI and CT scans are 3D while some are 2D like X-rays. CNN has multiple convolution layers. It takes raw pixels as input and performs transformation. The transformation is done through convolutional layers, rectified linear unit layers and pooling layers. The result of the transformation is given as input to fully connected layers. Fully connected layers compute the class scores and assign probability. This is how the classification of input into higher probability class by fully connected layers takes place.

- Convolutional Layer

A convolution is a binary operation (defined on two functions or values). In image transformation, input function is input values of image in the form of pixels and the other function is a filter through which input pixel values are passed. The convolution layer computes the dot product between input function and the filter. The dot product operation helps the CNN to derive low-level features of the image. Some of the low-level features are points, lines and edges. These features progressively build up the high-level features in subsequent layers. The high-level features are: eye, ear, lips, etc. in the image of human face. CNN neurons work in contrast to some other neural networks by establishing sparse connections (few inputs are connected to subsequent layer) and gradually learning features and reducing calculation of a number of weights and thereby increasing efficiency.

- Rectified Linear Unit Layer

The rectified linear unit layer improves the calculation speed by making use of activation function which converts the negative input values to zero which helps in avoiding the complex problem of gradient vanishing. The activation function to suppress negative inputs to zero is defined as: $f(z) = \max(0, z)$, Where z is negative input.

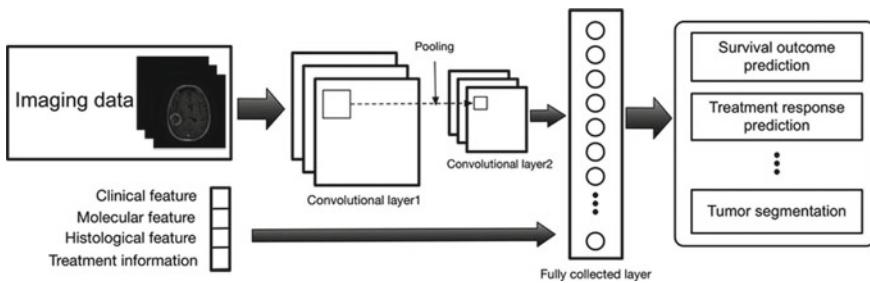


Fig. 2 Convolutional neural network using imaging and biomedical data [5]

- **Pooling Layer**

The pooling layer exists between the rectified linear unit layer and convolution layers. It reduces the number of weight box calculations and parameters to be calculated and to reduce the size of the input image. The location of dominant features is identified with the help of pooling layer.

- **Fully Connected Layer**

In the fully connected layer, all the neurons in the preceding layer are connected to all the neurons in fully connected layer. This layer computes the probability score which is used for further classification. For example, if two-class classifier is designed for MRI images (class 1: Diseased, class 2: Normal), the MRI image can be labeled either as diseased or normal. The calculation of the weights and the gradient is often done by the method of back-propagation. In this method, an error is computed at the output and distributed backward throughout the network layers. The error back-propagation algorithm (EBP) and gradient descent network allow CNN to learn association pattern from training images. EBP continuously reduces the error term till it reaches the maximum permissible error in the network [4]. Figure 2 is an example of convolutional neural network using biomedical data for brain tumor analysis. It consists of different convolutional layers, pooling layers and the fully connected layer. This network learns the abstraction of the input data. The clinical features and the imaging features are used for outcome evaluation (survival of disease, disease prediction, etc.) [5].

2.1.2 Recurrent Neural Network

Recurrent neural networks are mainly used in text analysis tasks such as speech recognition, text prediction and caption generation for images. The basic property of RNN is the existence of feedback loop from output side to input. In the field of medical imaging, RNNs are generally employed for the task of image segmentation. Various methods have been developed to combine RNN with CNN to segment fungal and neuronal structures from 3D electron microscope images. In the RNN, the output

is added to the next input and again fed back into the layer which provides capacity for contextual memory. The RNN are evolving into long short-term memory and gated recurrent units in order to avoid gradient vanishing problem.

2.2 *Supervised Learning Methods*

2.2.1 Auto-encoders

Auto-encoder is an unsupervised learning and representation technique which learns the features of input data without any labels. Auto-encoders reconstruct output data based on obtained coded input data. The auto-encoders work on a principle that output data and the input data carry similar features. Auto-encoders are manipulated and handled by weight matrix W and the bias b to reconstruct the input on output through hidden layers. For every dissimilarity, the auto-encoder function penalizes the model. Auto-encoders work in three general steps. In the first step, auto-encoders work as feature detectors which work in unsupervised manner. In the next step, it tries to reduce the complexity and model dimensionality. In the last step, auto-encoders reconstruct the output, i.e., new data with similar features to input training data. Auto-encoders are especially used in segmentation application of images. Such applications are: brain MRI segmentation, depiction of cancer cells, brain tumor detection, etc. [6].

2.2.2 Generative Adversarial Networks

Generative adversarial networks (GANs) are a promising unsupervised learning method in the field of medical imaging. Generative adversarial networks are a generative model comprising of two competing simultaneously trained models. One trained model generates the artificial training images. The other model classifies whether the images are artificially generated (by first model) or the real training images. The successful or desired execution of this method is one where the second model is unable to tell the difference between the artificial images and the real training images. Generative adversarial networks are relatively new concept and have various applications in synthetic medical data generation and MRI segmentation, etc.

3 Deep Learning Challenges in Medical Imaging

Application of deep learning to medical imaging is becoming the most ingenious technology since the inception of digital imaging. The major advantage of using deep learning in medical imaging is the hierarchical relationships within the image pixel data. This data can be discovered mathematically and algorithmically without

any laborious manual crafting of features. Various key research areas such as classification, localization, segmentation and object identification are moving forward with the application of deep learning. Dramatically increased use of electronic records in the medical industry has helped toward the contribution of large data required for the accurate deep learning applications. Major applications of deep learning in medical imaging are: recognizing symptom severity from psychiatric evaluation, brain tumor detection and segmentation, digital pathology, medical image analysis and diabetes self-management. The application of deep learning-based algorithms to medical field is a growing and captivating research area, and however, various challenges are slowing down its progress. These are discussed in the next subsection.

3.1 Dataset

Deep learning algorithms work on a large amount of training dataset to achieve greater accuracy. The deep learning model in any application such as regression, prediction, classification and segmentation is largely dependent on the size and quality of the dataset. The limited availability of training dataset is one of the biggest obstacles in the accomplishment of deep learning in medical imaging sector. The generation of such large medical image data is quite exigent for medical experts and requires a huge amount of work by the experts. Moreover, annotations of each and every predicted disease may not be possible due to the fact of scarcity of qualified experts or availability of sufficient cases of rare diseases [7].

3.2 Privacy and Legal Issues

The data privacy is much more complicated and difficult issue to tackle when the real-world images are used for deep learning in medical imaging. Data privacy must be addressed as both sociological and technical issue. Health insurance portability and accountability act of 1996 provides legal rights to patients regarding their personal information and medical records and establishes regulations for healthcare providers to protect, restrict its disclosure or use. Discarding personal information makes it difficult to link the data to unique person. However, privacy attackers can easily identify the personal information using association techniques. The privacy challenges are important to deal with in the initial stage as it might impact negatively both in terms of legal and ethical perspective. The limited and restricted data access reduces the information content which can reduce the accuracy of the deep learning model.

3.3 Dataset Standards and Interoperability

The heterogeneous nature of training data has led to dataset standards and interoperability becoming one of the major barriers. The type or nature of data differs in different hardware environments and thus large variation arises in medical imaging due to various factors such as country standards and sensor type. The deep learning in medical imaging sector requires a large amount of training data, and hence, combining several different datasets to achieve better accuracy is essentially required. Interoperability is a critical property in the health sector, and its implementation is still challenging. Health sector data must be standardized to achieve greater accuracy in deep learning. Various standardization bodies like HIPAA, HL7 are working toward this issue to define some guidelines and protocols toward greater interoperability.

3.4 Black Box Problem

Deep learning in medical imaging initiated many medical imaging applications and gave birth to new possibilities. In spite of its high performance in many applications, from classification to segmentation, it often proves difficult to explain the decisions it makes in a way that average person can understand. It is known as black box problem. The deep learning techniques get large input data, identify features and build predictive model, but its internal operations are not very well understood, and hence, they are often uninterpretable.

3.5 Noise Labeling

Even when the dataset images are annotated by the medical experts, accurate algorithm development is facing the noise as a significant barrier, e.g., detection of nodules in lung CT through LIDC-IDRI dataset. In this dataset, different radiologist annotates pulmonary nodules independently. No consensus was forced on this task, and it turned out that the number of nodules they did not unanimously agree to be a nodule was 3 times larger than the number they did agree on. A careful consideration must be taken while training a deep learning system on such data to deal with uncertainty and noise in the reference standard [8].

3.6 Class Imbalance Problem

In the field of medical imaging, finding images of abnormal classes might be challenging. For example, the breast cancer screening program has resulted in generating vast amount of dataset of mammograms worldwide. But, the majority of these mammograms are normal. So in designing such deep learning systems, efficiency and accuracy can become an important area of research. Leveraging clinical data is another challenge to deep learning. Physicians not only use medical images but also leverage a wealth of data on patient history, demographics, age to derive better conclusions. Various deep learning methods often incorporate this information in deep learning networks with medical images, and however, the improvements which were obtained were not as promising as expected. Hence, it is one of the major problems in deep learning, i.e., to balance the imaging features with various clinical features and prevent the clinical features from being drowned out.

4 Conclusion

In this paper, the major challenges of deep learning to medical imaging are discussed. Successful diagnosis systems in medical imaging also have been discussed. The authors focused on challenges in supervised and unsupervised approaches. It has been observed that deep learning plays a significant role in improving the accuracy in recognition of medical images for various diseases (breast cancer, brain tumor, etc.). Thus, deep learning has attracted the researchers in developing the medical imaging tools. These tools are based on past data of patients. Key challenges deep learning facing today are: massive datasets, overfitting of data, variations in input, quality of data, understanding of the context and high volume data. Prediction of the disease for a human being is observed as the future of deep learning.

References

1. Madabhushi A, Lee G (2016) Image analysis and machine learning in digital pathology: challenges and opportunities. *Med Image Anal* 33:170–175
2. Ker J, Wang L, Rao J, Lim T (2017) Deep learning applications in medical image analysis
3. Razzak MI, Naz S, Zaib A (2016) Deep learning for medical image processing: overview, challenges and future
4. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JA, van Ginneken B, Sanchez CI (2017) A survey on deep learning in medical image analysis
5. Zhou M, Scott J, Chaudhury B, Hall L, Goldgof D, Yeom KW, Iv M, Ou Y, Kalpathy-Cramer J, Napel S, Gillies R (2017) Radiomics in brain tumor: image assessment, quantitative feature descriptors, and machine-learning approaches. *Am J Neuroradiol* 39 <https://doi.org/10.3174/ajnr.a5391>
6. Lara J, Cooper R, Nissan J, Ginty AT, Khaw K-T, Deary IJ, Lord JM, Kuh D, Mathers JC (2015) A proposed panel of biomarkers of healthy ageing. *BMC Med* 13(1):222

7. Lixie E, Edgeworth J, Shamir L (2015) Comprehensive analysis of large sets of age-related physiological indicators reveals rapid aging around the age of 55 years. *Gerontology* 61(6):526–533
8. Yang J, Huang T, Petralia F, Long Q, Zhang B, Argmann C, Zhao Y, Mobbs CV, Schadt EE, Zhu J et al (2015) Synchronized age-related gene expression changes across multiple tissues in human and the link to complex diseases. *Sci. Rep.* 5:15145

Chapter 29

Home Security System Using Face Recognition



**Janhavi Baikerikar, Vaishali Kavathekar, Yash Agarwal, Sanika Bhat,
Christine Polly and Saloni Juwatkar**

1 Introduction

With the rising number of thefts in urban areas, there is a need to design a system for ensuring the safety of the homes. We propose a low-cost home security system which detects intruders. Whenever an intruder enters the house, the homeowner may not always be aware of the intrusion at that moment. This system intimates the user about the intrusion by capturing images on the spot and comparing it with the faces of the family members and triggers alarm in case of an intruder. The Internet of things model proposed uses Raspberry Pi 3B+ with Raspbian Stretch OS works at the base of the system using dependency libraries like OpenCV, NumPy, SciPy, scikit-image, Dlib, Python for executing face recognition and system hardware like PiCamera, passive infrared sensor, Wi-Fi module. Many existing systems cannot differentiate between the intruders and family members. The motion sensors may trigger an alarm even when the family member is present at the door resulting in a false alarm. Our

J. Baikerikar · V. Kavathekar (✉) · Y. Agarwal (✉) · S. Bhat (✉) · C. Polly (✉) ·
S. Juwatkar (✉)

Information Technology Department, Don Bosco Institute of Technology, Mumbai, India
e-mail: vaishali.dbit@dbclmumbai.org

Y. Agarwal
e-mail: yashdbit@gmail.com

S. Bhat
e-mail: bhat.sanika@gmail.com

C. Polly
e-mail: christinedbit@gmail.com

S. Juwatkar
e-mail: salonijuwatkar@gmail.com

J. Baikerikar
e-mail: janhavi.dbit@dbclmumbai.org

proposed system tries to minimize these false alarms by making use of a camera in addition to the sensors to detect intruders.

The paper is organized as follows—Sect. 2 gives the literature survey which gives details about the existing systems. This is followed by our proposed system in Sect. 3 and the algorithm used. Section 4 gives the details of the face recognition module followed by results in Sect. 5. This is followed by conclusion and future work.

2 Literature Survey

Automated security systems play an important role in security to track illegal intrusions within the home (indoors and outdoors). There has been research done in the design of various types of automated security systems.

Sahani et al. [1] mention about designing a wireless access control system which is ZigBee-based and uses an image processing technique called as principal component analysis algorithm for face recognition. The system developed is basically a Raspberry Pi-based door access control and home security controlled through a Web-page. The basic mechanism of the system is to confirm visitor identity with the user/house owner and acts based on the users' reply to electronic mail.

The reply is recognized by the algorithm and then fed into the Raspberry Pi which handles the door mechanism. There are various modules in the system, namely face recognition, ZigBee, wireless sensor network, lock module, Internet of things gateway, message module. SQLite is used for local data storage purpose. Alerts to the user are sent via an electronic mail or a message to which the user can respond in a predefined manner. All in all, this system is a low-cost and easy-to-install system in such an environment. The basic working of the system will be such that the images are captured when the human is detected. The system confirms the visitor by emailing the photograph and taking into account its reply. The reply will ultimately decide whether the door should be unlocked or not.

Venkatesan [2] deals with a home security system designed for houses, banks and offices. It is mainly provisioning for theft detection and fire detection. Alerts to the user are sent via text messages. The leverage of the system is that the user can be alerted irrespective of his location and whether he is connected to the Internet. The system also consumes less power as the camera is only activated when the passive infrared sensor detects a human presence instead of a daylong live feed. The basic working here is that whenever the passive infrared sensor senses a motion, the Raspberry Pi triggers the Raspberry Pi Camera to capture an image and send it to the user as an alert via text. The system architecture consists of Raspberry Pi as the processing unit, passive infrared sensor and fire sensor, messaging/alert unit. The disadvantage of the system is that it will alert the user even if a strong wind blows or there is a slight movement because of pets or any other environmental circumstances.

Brundha et al. [3] mention a home automation system where client–server interaction is involved. Client is connected to a sensor which sends data to server, server adjusts the lightning, and the user is notified. Added feature includes security which

captures images of the person entering the house and compares the facial features with that in the database using MATLAB. If an intruder is detected, the image is sent to the user. This system makes use of Arduino Uno along with passive infrared sensor and brightness sensor. MATLAB is used for facial recognition to alert the system. The client–server interaction works on the data received from different servers and adjusts the environment brightness accordingly. The security system involves passive infrared sensor triggering Arduino to interact with MATLAB which compares facial features of those trained with the database. The user can access image of the intruder through node Internet protocol or domain name server. A prototype is designed where a low-power, cost-effective system is provided which does presence detection, identification and authentication of the stranger.

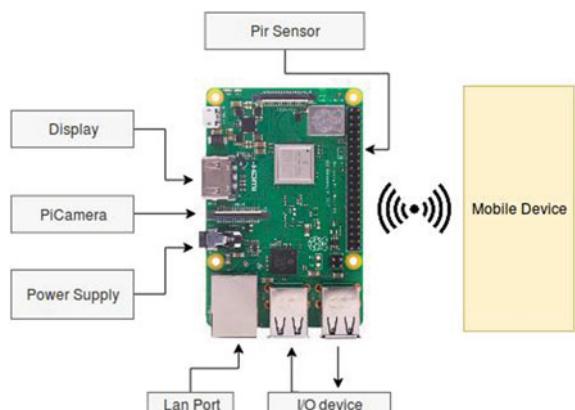
The system [4] makes use of Web camera to capture images and the electric door strike which is an actuator, and telegram is the user interface which is used by the user to monitor the activity of the stranger.

The Raspberry Pi is connected to the doorbell; when it is pressed, a signal is sent to the Raspberry Pi. The Web camera is turned on to capture the photographs from it, and the photograph is sent to the telegram user. Telegram uses an application program interface which is compatible with the Raspberry Pi. The telegram server responds to request and sends message to the user. The user then authenticates the visitor allowing him through the application.

3 Proposed System

Our system in Fig. 1 comprises a Raspberry Pi model which is connected to different components such as sensor, keyboard and camera. This model communicates with the user interface using a server. The system would sense the motion, causing the camera to capture the video stream. Only when a face is detected, it would be compared with the existing face data, so that the intruder can be differentiated from the homeowners

Fig. 1 Architecture of home security system



or visitors approved by the homeowners. We are making use of the Viola–Jones approach to detect a feature. The image is first converted into gray scale, and all pixel values are calculated. This value is subtracted from the total of white boxes and compared with the threshold of predefined values. If criteria are met, the feature is selected.

The steps followed are as follows:

1. The Raspberry Pi detects a person with the help of passive infrared sensor [5].
2. Passive infrared sensor triggers the camera to start a video stream [6].
3. Faces are detected and then compared with the ones in the database with the help of algorithm [7].
4. If not recognized, name it as a intruder and then capture and store the images of the intruder.
5. An alert is sent to the user along with these images [8].
6. The images are sent via a server which is common between the system and android application [9].

4 Face Detection System

The algorithm has four stages:

1. Haar feature selection.
2. Creating an integral image.
3. AdaBoost training.
4. Cascading classifiers.

The features that are universally involved in the sum of image pixels with rectangular areas are generally sought by detection framework. In practice, these frameworks bear some resemblance to Haar basis functions, which has been used for image-based object detection in real world. Moreover, features used by Viola and Jones algorithms reply to more than one rectangular area which is generally more complex to use. The value of the given features is sum of the pixels within the area of particular clear rectangles minus sum of shaded area. Features sorted from the rectangle are primitive in nature when compared to steerable filters that are alternatives. They are also sensitive to vertical and horizontal features; meanwhile, their feedback is considerably coarser.

5 Results

Our work aims to assist the homeowners in an efficient manner by giving an easy setup that monitors any intrusion. This personal system enables a homeowner to monitor his/her home on a routine basis and facilitate a secure environment. The

Table 1 Test cases

Test Id	Test cases	Actual result	Expected result	Result
1	Hair left open/tied hair	Face detected and recognized	Face detected and recognized	Pass
2	Shaved face or beard	Face detected and recognized	Face detected and recognized	Pass
3	Glasses	Face detected and recognized	Face detected and recognized	Pass
4	Bad lighting	Face detected and recognized	Face detected and recognized	Pass
5	Facial expression	Face detected and recognized	Face detected and recognized	Pass
6	Side profile	Wrong or no face detected	Face detected and recognized	Fail
7	Multiple faces	Face wrongly detected	Face detected and recognized	Fail

application running on the android phone/device easily interfaces with the Wi-Fi-enabled Raspberry Pi 3 and is well suited for homes in residential areas. Also, an application is built that can interface with the Raspberry Pi with ease and retrieve data over the network displaying it in a CardView template for the homeowner. The processed data by the Python face recognition program coded on the Raspberry Pi helps us recognize intruders and send images of them to the android application providing a proof and backup for the homeowner. The faceviews recognized are indicated in Table 1.

Advantages over existing methods are given below

- The main thrust of this proposed system is to provide instant and accurate results to its users. Our system is highly beneficial to residential areas.
- Notifying homeowners on their android devices and through email which is a leverage over other existing models [10].
- The system detects faces and recognizes it, distinguishing between intruder/s and homeowner/s, and hence reduces repetitive monitoring by the homeowner as in live streaming applications [11].
- The entire sensor analysis can take place in a wireless environment.
- This system assists in proper alerts and helps monitor residential areas.
- The interface displayed in Fig. 2 shows the intruder's image displayed over the email. It is a useful feature for homeowners who may be at some other location. The intruder detected by the system is notified to the homeowner by sending an email as a proof or backup.

The interface displayed in Fig. 3 shows the android application interfaced with Raspberry Pi that retrieves data over the network displaying it in a CardView template for the homeowner. The intruder detected by the system is notified to the homeowner over the android application as a proof and backup to any intrusion.

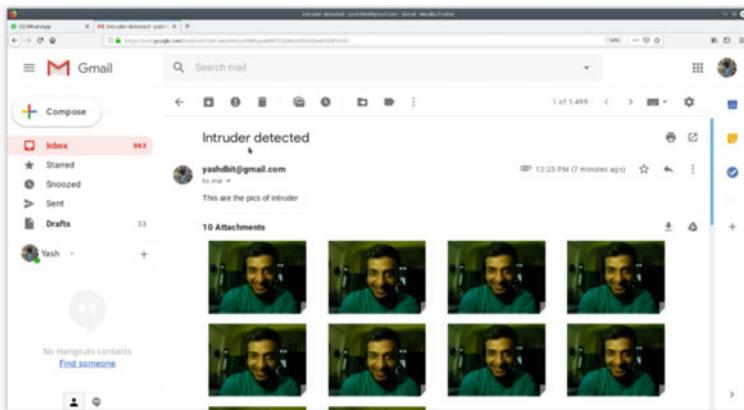


Fig. 2 Notification by email

6 Conclusion and Future Work

No one is completely safe from becoming a victim of intrusion. As a result, everyone needs some type of home surveillance system to protect themselves, their property and their loved ones. The Raspberry Pi is a good platform for building low-cost, but highly capable, embedded systems. The interfaces are built using simple low-cost electronics and a bit of configuration to create very functional and flexible systems. The inclusion of a dedicated camera interface and networking interfaces gives the homeowner everything they could possibly need for an Internet-connected home security system. A series of experiments have been carried out on the proposed system to detect any intruder; additionally, our work illustrates the way to monitor and track the home through an image processing camera and the way to send notifications to the homeowner about the actions outside the home. The Raspberry Pi is a really compact processor which is fast and efficient. Initially, a localhost server is used and alerts are sent to android application connected to same network. In the future, the system and android application can run on different networks. The system can be trained to predict age and gender of the intruder as well. It will also help in getting proper identification at the door, and it may be possible to have cameras with built-in facial recognition software. In this way, the camera can take footage of an intruder and could cross-reference that face with the social media profiles and police can instantly identify the culprit.

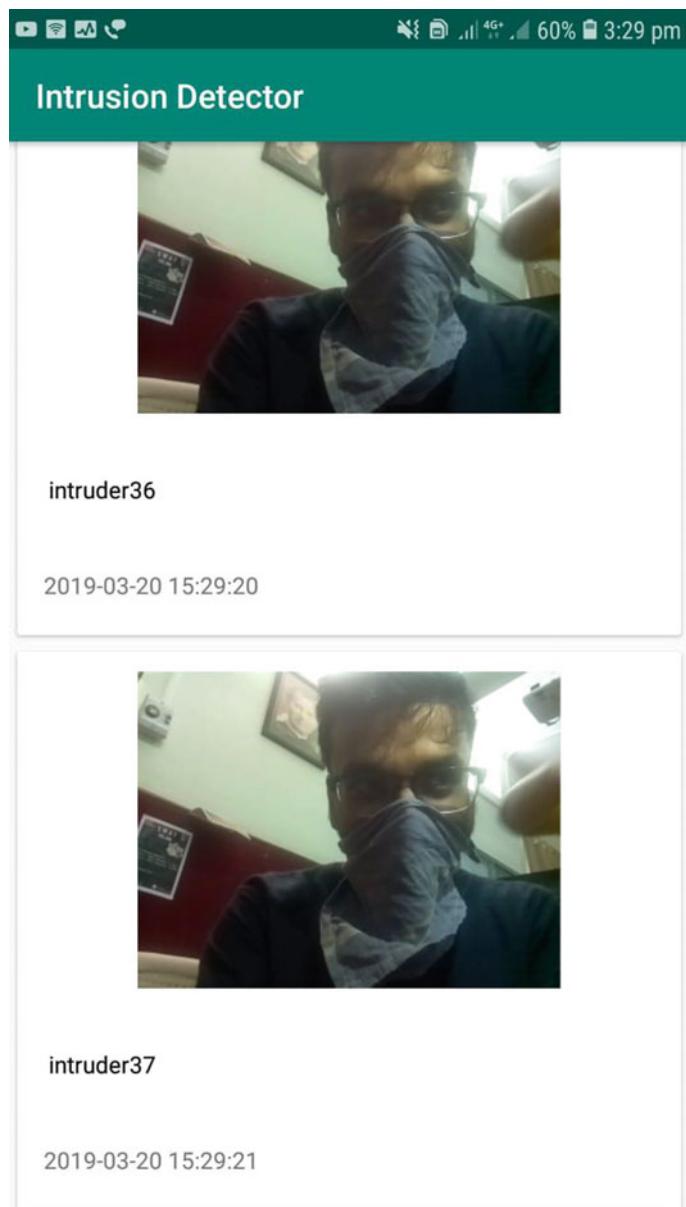


Fig. 3 Android application interface

References

1. Sahani M, Nanda C, Sahu AK, Pattnaik B (2015) Web based online embedded door access control and home security system based on face recognition. In: 2015 international conference on circuit, power and computing technologies (ICCPCT)
2. Venkatesan S, Jawahar A, Varsha S, Roshne N (2017) Design and implementation of an automated security system using Twilio messaging service. In: 2017 international conference on smart cities, automation and intelligent computing systems (ICON-SONICS), Yogyakarta, pp 59–63. <https://doi.org/10.1109/icon-sonics.2017.8267822>
3. Brundha SM, Lakshmi P, Santhanalakshmi S (2017) Home automation in client-server approach with user notification along with efficient security alerting system. In: 2017 international conference on smart technologies for smart nation (SmartTechCon), Bangalore, pp 596–601. <https://doi.org/10.1109/smartechnetcon.2017.8358441>
4. Anvekar RG, Banakar RM (2017) IoT application development: home security system. In: 2017 IEEE technological innovations in ICT for agriculture and rural development (TIAR), Chennai, pp 68–72. <https://doi.org/10.1109/tiar.2017.8273688>
5. Dmello A, Deshmukh G, Murudkar M, Tripathi G (2016) Home automation using raspberry Pi 2. In: 2016 international journal of current engineering and technology
6. Sruthy S, George SN (2017) WiFi enabled home security surveillance system using raspberry Pi and IoT module. In: 2017 IEEE international conference on signal processing, informatics, communication and energy systems (SPICES), Kollam, pp 1–6. <https://doi.org/10.1109/spices.2017.8091320>
7. Hussein NA, Al Mansoori I (2017) Smart door system for home security using raspberry pi3. In: 2017 international conference on computer and applications (ICCA), Doha, pp 395–399. <https://doi.org/10.1109/comapp.2017.8079785>
8. Jain S, Vaibhav A, Goyal L (2014) Raspberry Pi based interactive home automation system through E-mail. In: 2014 international conference on reliability optimization and information technology (ICROIT), Faridabad, pp 277–280. <https://doi.org/10.1109/icroit.2014.679833>
9. Quadri SAI, Sathish P (2017) IoT based home automation and surveillance system. In: 2017 international conference on intelligent computing and control systems (ICICCS), Madurai, pp 861–866. <https://doi.org/10.1109/iccons.2017.8250586>
10. Tanwar S, Patel P, Patel K, Tyagi S, Kumar N, Obaidat MS (2017) An advanced internet of thing based security alert system for smart home. In: 2017 International conference on computer, information and telecommunication systems (CITS), Dalian, pp 25–29. <https://doi.org/10.1109/cits.2017.8035326>
11. Othman NA, Aydin I (2017) A new IoT combined body detection of people by using computer vision for security application. In: 2017 9th international conference on computational intelligence and communication networks (CICN), Girne, pp 108–112. <https://doi.org/10.1109/cicn.2017.8319366>

Chapter 30

Semantic Web-Based Knowledge Extraction: Upper Ontology Guided Crime Knowledge Discovery



Kaneeka Vidanage, Noor Maizura Mohamad Noor, Rosmayathi Mohemad and Zuriana Abu Bakar

1 Introduction

Crime is a controversial social hazard, spreading across the globe. As pointed by Ghani in [1], the root cause of crime rate escalation across the world is identified as urbanization. The author further elaborates in the paper, how this actually happens. As claimed by Ghani [1], many rural residents progressively approach to suburban and city areas assuming they would get more business, occupational opportunities and a comfortable life zone. Even though, the majority will not succeed to attain these goals. In fact, they will start to experience, higher cost of living in suburban areas compared to the rural life habitats they are familiar with. Therefore, soon, this becomes unbearable for many, and it will act as the catalysts to incite criminal activities [1, 2]. As elaborated by Badiora and Afon in [3], these crime rates are escalating at an alarming rate, adversely affecting the socio-economic and quality of life. Both Ajaegbu [4] and Katsina [5] argue unemployment and the economic hardships mixed with urbanization are the main triggering points for deadly crimes.

As there is emerging enthusiasm growing in semantic Web-based technologies, it is decided to seek the potentials of using semantic technologies to resolve the problem of crime escalation.

With the introduction of the semantic Web, the whole Internet has become a Web of knowledge. This has become even more fascinating as the knowledge represented in the semantic Web is machine-readable [6]. Even though construction of a domain constrained semantic Web or in other words, an ontology from the scratch is not an easy task [7], as there are no fully automated methods identified up to date. Human intervention is essential for knowledge verification [8].

K. Vidanage (✉) · N. M. M. Noor · R. Mohemad · Z. A. Bakar
School of Informatics and Applied Mathematics, University Malaysia Terengganu (UMT), Kuala
Nerus, Malaysia
e-mail: Kaneeka.online@gmail.com

Therefore, as mentioned above, a few facts are apparent. Those would be ontology construction is not an easy task [7, 8]. But if created, the advantages of it would be ample, as of their machine and human readability [6]. Therefore, as much as possible maximum benefits should be derived from created ontologies. Even though, obtaining maximum outcomes from created ontologies have become impossible due to two main causes.

One is the difficulty of understanding the schematic structure of a defined ontology or the knowledge model [9, 10, 11]. Because without knowing the schematic structure, knowledge retrieval from an existing ontology would not be feasible [12, 11]. The second aspect is the requirement of high technical knowledge to understand a schema and necessity of writing queries for knowledge retrieval [11, 12, 13]. Both these facts imposing a great barrier on ontology reusability as well as knowledge comprehension presented in ontology formats [7, 14]. Therefore, the implications of this will affect both technical and non-technical audiences in multiple ways [15]. Additionally, most of the hardly created ontologies will become stagnated soon on the Internet, wasting intellectual and cognitive efforts of the creators [7].

Crime domain as a whole is a very vast discipline with lots of sub-regions as weapons, standard operational procedures (SOPs), evidence gathering, etc. [16]. Ontologies being domain rich conceptualizations [17], what if, a specific sub-region related crime domain knowledge is presented in the form of an ontology? Then, with the introduction of the proposed knowledge discovery framework, despite all the technical barriers conversed above, non-technical criminologists, criminology students also can infer the knowledge represented via the ontology. Because suggested knowledge extraction framework is capable of encapsulating complex schematics and querying barriers related to the ontology and presenting the stored knowledge in natural English. Non-tech crime specialists and students are also language literate, and hence, suggested framework will efficiently disseminate relevant information amidst specialists and students, facilitating investigations and learning requirements. This is one potential use case of the suggested framework and crime domain. But, this framework is not only limited to the crime domain. It is planned to design as a domain and schemata independent framework, to support the rapid growth of use in semantic technologies, both inside and outside the computing domain, facilitating non-technical consultants' usage requirements as well [15, 18].

Therefore, this paper is focusing on proposing an architectural design, leading towards a domain and schema independent, semantic knowledge-based, knowledge extraction framework, to maximize the use of existing knowledge models and at the same time facilitates natural language-based knowledge dissemination among both newly created or existing knowledge models, making them to be the main research contributions.

The remaining section of the paper will discuss, about related works, methodology, results and discussion and conclusion, respectively.

2 Related Work

Ontologies are domain rich conceptualizations. This is how Spasic et al. have defined ontologies in [17]. Ontologies can be defined to present knowledge associated with any domain. Currently, also there are thousands of already defined ontologies available on the Internet. Few of the locations where you could access these predefined ontologies are on Vocab.org [19], Swoogle [20], LOV [21] and Protégé Wiki [22]. AberOWL [23] and BioPortal [24] are two other important ontology repositories, which contains thousands of biomedical ontologies. Additionally, Covert To RDF [25] provides numerous converters capable of translating comma-separated values (CSV) and spreadsheet-based data into Resource Description Framework (RDF) format. (RDF) or OWL (Ontology Web Language) formats, are the most popular formats for semantic Web knowledge representations [26]. Protégé IDE for the ontology development also promotes a variety of plugin series [27] which are capable of converting a variety of data formats into above-listed RDF or OWL formats. Therefore, as conversed above, it is very apparent that there are plenty of existing ontologies and ontology browsers as well as alternative mechanisms of converting information dispersed across the Web into semantic Web friendlier formats. This is good evidence to show the current enthusiasm towards semantic Web-based technologies. Consequently, it can be determined as there is enormous potential for ontology reusability and ontology creations.

Even though, as already conversed above, complexity in understanding the schemas, ontology querying barriers, inability associated with comprehension of OWL or RDF-based technical representations act as main bottlenecks for both tech and non-tech audiences in hindering the ontology reusability and creation [9, 10, 11, 12, 13] despite the above-discussed potentials.

As we are trying to align the application outcomes of this research towards the criminal domain due to the timely relevance of that region, next it will be investigated on what are the already existing intelligent computational prototypes linked with the criminal domain analysis. As the first result, it is located Masitha et al. had attempted in [28] to come up with a crime ontology with the intention of facilitating relevant officials to react quickly on crime matters. In this research, they have constrained their crime domain to motorcycle thefts only because crime, in general, is a very vast subject discipline [16]. Researchers have initially investigated multiples of criminal case reports and determined a few important elements as vital in extracting crime information.

After this, they have come up with a taxonomized structure representing the inter-relations among identified components to assure methodical storing of crimerelated information collected. Eventually, the designed ontology is implemented via the Top-Braid Standard composer tool. Additionally, researchers have developed a case base repository to store information associated with similar types of cases occurred in past. Once the user enters the new information about the reported case, initially it will be reasoned by the implemented crime ontology, and afterwards, crime ontology will talk with the relevant case base and will extract out the most related set of past

crimes occurred, evidence collected, decisions arrived and variety of other necessary information.

As the next finding, another ontology and decision support system [29] working in combination to reason about evidence collected from a crime scene. Investigation analysis of complex and dynamic crimes is not an easy task.

There need to be established and carefully governed procedures [29]. In fulfilling this requirement, having an ontology in place to guide the respective decision support systems as needed for the crime investigation mining will enhance the overall throughput of the entire process.

Likewise, this discussion can be continued as there are a sufficient amount of researches where intelligent systems have been developed for the criminal domain.

The accuracy, reliability and practicality of these knowledge models would be high in value as those are developed and tested by teams of professors, researchers and experts in the respective fields [30]. Even though, the plight is, criminologists and students following criminology pathways are not computer specialists. Hence, they cannot comprehend the valuable pieces of knowledge expressed, in semantic Web forms though they are available free of charge on the Internet [26]. Authors of this paper consider this as an utter cognitive and intellectual waste of the experts and researchers who are involved in developing those knowledge structures as they become soon stagnant on the Web, after serving one specific purpose. From the other hand, computational reusability of those will also become very low due to the difficulties of comprehending complex schematics, as being ignorant about the schema, querying for knowledge retrieval will be infeasible [9, 10, 11, 12, 13]. Therefore, what if there is a framework introduced, which can overcome those technical barriers and hurdles associated with effective knowledge reusability and dissemination?

For instance, as one practical use case, assume the process of training criminology officers, detectives and investigators. They need to gain a comprehensive knowledge of crime types, evidence gathering procedures, crime analysis, etc. [31]. In fulfilling these purposes, there are ample of carefully designed knowledge models available in popular semantic Web formats on the Internet, mostly free of charge [16]. If required formats of knowledge models are not available, these could be created through collaborative efforts of computer scientists and criminology domain specialist. In fact, this could be a one-time effort. Because then the created knowledge models can be used again and again over batches of criminology students and specialists for effective knowledge dissemination assuring knowledge reusability aspects as well.

Next, what about the existing frameworks based knowledge extraction from OWL or RDF knowledge models? Subsequently, it is decided to widen up the literature investigation on the assessment of similar frameworks.

Ghorbel et al. proposed a tool by the name “Memo Graph” in [32]. This is an ontology taxonomy visualization tool. It will depict the taxonomic structure of the ontology in visual forms. But, we cannot expect non-technical audiences to look at taxonomy and get an idea to query the ontology. Because still, SPARQL-based querying will be a barrier for them in knowledge retrieval. Another similar type of system located is Semantic Web Portal developed by Ding et al. [33]. Going beyond the graphical visualization of the taxonomical structure of the ontology, this system

is capable of visualizing the triple structure as well. Furthermore, as claimed by Ding et al. in [33], this system can work with any domain. Even though, there is a major bottleneck. That is before applying Semantic Web Portal to a selected domain, portal ontology needs to be created. Portal ontology provides domain associated axioms, facilitating the operation of Semantic Web. Therefore, again, it is not acceptable from a non-technical user to create a portal ontology, to feed the domain knowledge to the Semantic Web Portal. Further, this tool does not have the feature of knowledge extraction and presentation in natural language.

Other than those visualization tools, in [34], team of researchers have implemented a system which is capable of natural language querying of an ontology. However, it is a static, domain and schema-dependent software. In their research, they had created an accommodation ontology and English to SPARQL conversion, and SPARQL queries are statically mapped to the accommodation ontology. Hence, the structure proposed in [34] is not a framework which is capable of the domain and schema-independence analysis as it is statically bound to one specific ontology only.

Therefore, as reviewed so far, the following reflections can be derived. There is great enthusiasm over the use of semantic technologies to overcome, recurring social issues. Even for the considered criminal domain also, it is possible to locate multiples of intelligent systems already deployed. Even though, as previously pointed out in the literature review above, semantic technologies are not doing good in the dimensions of knowledge reusability and dissemination. Further, there are no proper existing frameworks located, which are capable of natural language-based knowledge extraction from popular semantic Web formats with domain-independent knowledge models. Therefore, as the gap of this research, it can be concluded the issues associated with poor performance in reusability and dissemination aspects of the semantic technologies and none existence of domain and schema-independent, natural language-based knowledge extraction framework. Consequently, the remainder of this paper is emphasizing on a proposed architectural design leading towards the framework to address the above-listed gaps, which could be interpreted as the contribution of this research.

3 Methodology

In the process of finalizing an appropriate architecture for this framework, it is identified in [35], a group of researchers have suggested the concept of divide and conquer. As they have pointed out, attempting to fulfil all the tasks from one module will increase the complexity and coupling associated with the module. In [36], they further suggest the definition of a complex problem space via multiple resolution layers will provide the opportunity of attention to detail in knowledge modelling and analysis aspects, and it will further prevent the single module being flooded with lots of information and resulting with complex schematic structures or conditions. Therefore, considering those suggestions, as the first decision point it is concluded that the proposed framework should also comprise with multiple resolution layers,

and hence, the process of knowledge extraction in natural language from RDF and OWL knowledge models is not an easy, straightforward task.

As the next step, arguments presented in ontology designing aspects are considered. Multiple of researchers have justified ontologies would be the strongest conceptual representations associated with elaborating a complex domain, and they would be ideal in process enforcement as well [6, 8]. Here, in this research also, knowledge extraction from RDF/OWL knowledge models is also a complex process, which needs to be enforced carefully. Therefore, recommendations provided in numerous research papers evaluated confirmed the intention of using ontologies for this research as well. Next, question is how to determine the architecture or the structure of the ontology? For this concern, Sowa [37] has provided a good explanation in his research paper. As a claim by Sower [37], there are multiple types of ontologies. Top-level ontologies describe more generalistic concepts associated with a “Thing”. When gradually reaching from top to bottom, knowledge represented in ontologies will also vary from meta-concepts to application and from application to the domain and at the bottommost layer, having task ontologies focusing on specified aspects associated with the considered domain. In this research, the process of knowledge extraction from a given RDF/OWL model is domain-independent. Therefore, it confirms that an upper-level ontology needs to be designed as the considered procedure for knowledge extraction is not domain-specific, and it is generalistic.

Smith proposes [38] mechanics theory, which is enforcing on procedures than declarative aspects associated with the domain. Researchers have recognized mechanics theory to be much appropriate in constructing task ontologies. Helix spindle theory [39] comprises three main stages which are continuing in an incremental manner until a satisfied criterion is met. The first stage is the conceptualizing stage. In this stage, in-depth brainstorming and conceptualizing will take place, pertaining to the considered use case. The output of this phase would be a natural language-based reflective description of the finalized mental image. Then, the second phase would be the elaboration phase, where this natural language description derived from phase one will be graphically represented via a taxonomical structure. Eventually, the final phase would be the definition phase, where the ontology construction will be completed with the required knowledge injections. All these phases are iterative and incremental and should be logically interconnected with each phase as necessary.

Therefore, it is decided to use helix spindle method [39] for the taxonomy derivation of the ontology and mechanics theory [38] for the knowledge injection to the created taxonomic structure. The step-wise progression of the methodology followed is graphically presented in Fig. 1, depicted below.

4 Results and Discussions

Figure 2 illustrates the communicational architecture in between the instructional upper ontology and the relevant decision support systems integrated to the respective endpoints of the upper ontology.

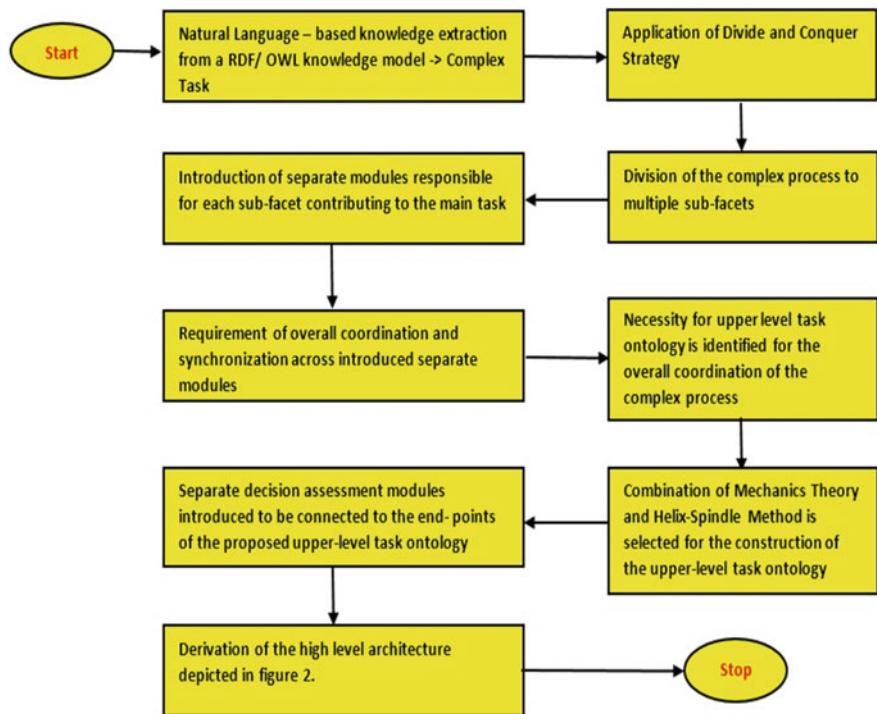


Fig. 1 Overall flow of the methodology

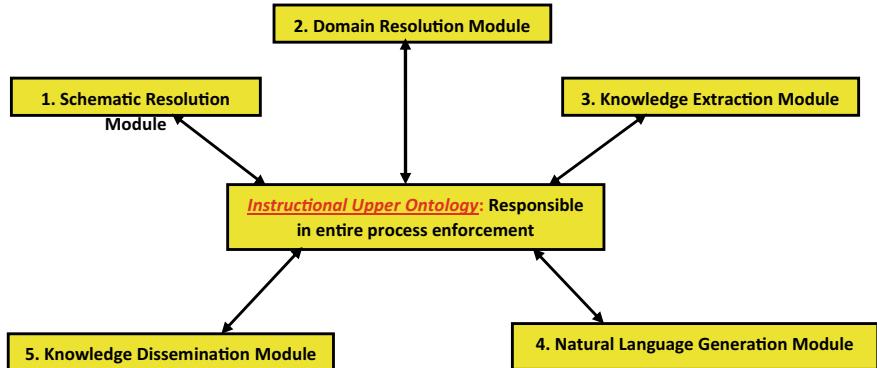


Fig. 2 Communicational architecture in between instructional upper ontology and decision support systems the relevant

Natural language-based knowledge extraction from RDF or OWL semantic formats is not an easy task. Hence, to methodically perform the task according to the enforced process by the instructional upper ontology, multiples of DSS systems are introduced. Knowledge extraction from RDF or OWL-based semantic formats is executed through multiple resolution layers, synchronized under the control instructional upper ontology. Figure 3 mentioned will elaborate on the step-wise execution of the knowledge extraction process associated with the uploaded RDF/OWL-based knowledge model file. Each cell in Fig. 3 clearly elaborates the steps to be executed in one after another sequentially, from start to end. This entire process can be governed by the communicational architectural structure proposed in Fig. 2.

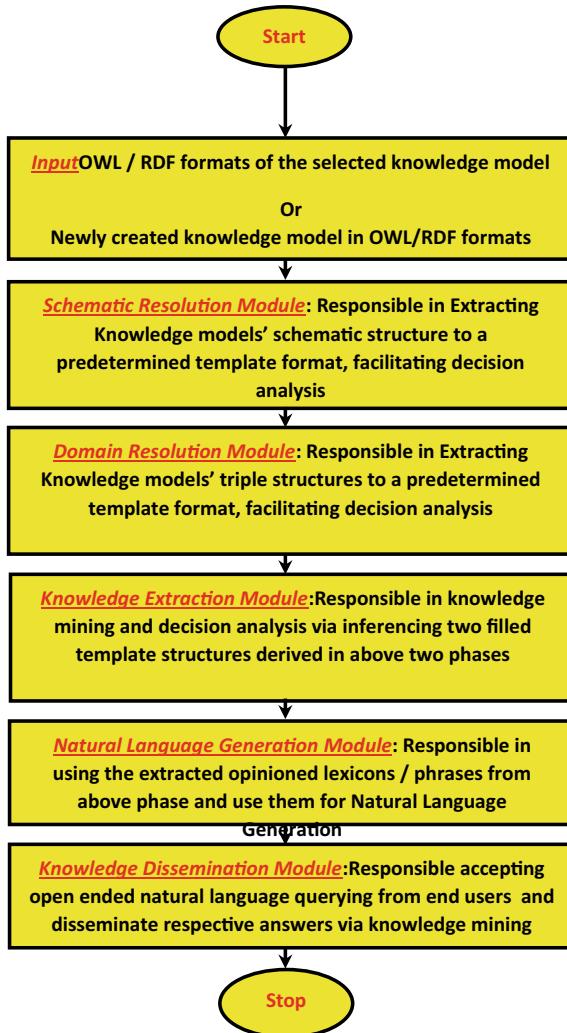


Fig. 3 Overall execution flow of the proposed framework

5 Evaluation

An evaluation strategy can be planned as mentioned in below Fig. 4, to assess the efficacy of the proposed framework for knowledge extraction from semantic Web-based knowledge models.

Figure 4 suggests creating a suitable ontology, covering a sub-discipline related to the crime domain, unless if no appropriate existing ontology is located. The ideal mechanism will be to create a suitable ontology covering a specialized sub-discipline associated with the crime domain. This can be done via brainstorming with a few crime specialists on a selected crime domain.

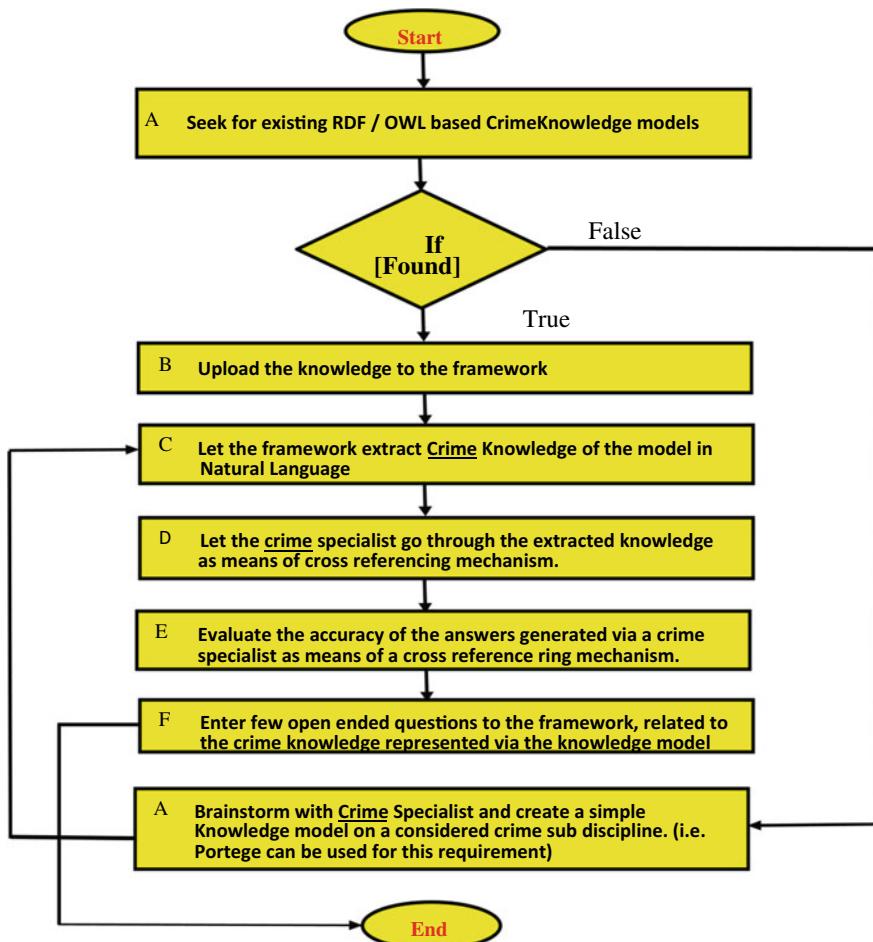


Fig. 4 Suggested evaluation strategy

Table 1 Test statistics of the proposed architecture

Measurement	Accomplishment
Sensitivity	0.78
Precision	0.90
Accuracy	0.86
F-measure	0.80

Crafted ontology can be presented to the suggested framework either as an OWL document or an RDF document. Then, let the framework extract the axioms stored in the ontology and verbalize it in natural English. Henceforth, the verbalized contents can be cross-referenced with the crime specialists who have involved in the ontology creation phase. Then, through their instincts on verbalized output, it can be verified if there any information or important aspects have been missed/lost. Using it as the main evaluation platform, a confusion matrix can be derived to determine the true positives, false positives, false natives and true negatives associated with the verbalized output and ontology contents.

Via the usage of those parameters, evaluation matrices such as recall, precision and F-measure can be derived, to numerically assess the throughput of the verbalization process.

Table 1, depicts the test statistics derived, after exercising the above framework on a crafted crime ontology. Verbalized results are cross-referenced with crime specialists, and Fig. 5 demonstrates the mechanisms associated with deriving of true positives, true negatives, false positives and false negatives leading towards the test statistics calculations presented in Table 1.

Confusion Matrix																			
N:-Total amount of expressions considered in a verbalization →315																			
TP:- Correctly verbalized expressions																			
FP:- Wrongly verbalized expressions																			
TN:- Irrelevant expressions are ignored from verbalization output																			
FN:- Irrelevant expressions are included in the verbalization output																			
N=TP + TN + FP + FN																			
<table border="1"> <thead> <tr> <th></th> <th>Predicted :No</th> <th>Predicted :Yes</th> <th></th> </tr> <tr> <th>Actual :No</th> <td>TN=179</td> <td>FP=12</td> <td>(TN+FP)=191</td> </tr> <tr> <th>Actual :Yes</th> <td>FN=27</td> <td>TP=97</td> <td>(FN+TP)=124</td> </tr> <tr> <th></th> <td>(TN+FN)=206</td> <td>(FP+TP)=109</td> <td></td> </tr> </thead> </table>					Predicted :No	Predicted :Yes		Actual :No	TN=179	FP=12	(TN+FP)=191	Actual :Yes	FN=27	TP=97	(FN+TP)=124		(TN+FN)=206	(FP+TP)=109	
	Predicted :No	Predicted :Yes																	
Actual :No	TN=179	FP=12	(TN+FP)=191																
Actual :Yes	FN=27	TP=97	(FN+TP)=124																
	(TN+FN)=206	(FP+TP)=109																	

Fig. 5 Verbalized results assessment

For the experimented scenario of crime knowledge ontology, proposed architecture-based verbalizer has presented a reasonable performance with an approximate overall accuracy above 80%. Even though, it is suggested to verify the performance of this architecture, via exercising it on multiple more knowledge models obtained from a variety of domains.

6 Conclusion

As a futuristic resolution, authors of this paper have proposed an architectural structure for a potential framework which could resolve the technical barriers associated with the effective use of new and existing semantic Web-based knowledge models. Further, as a functional outcome of the proposed framework, it will facilitate natural language-based information dissemination, allowing criminologists, detectives, police officers and students to experience the benefits of the semantic Web, though they are not ontologists or computer specialists. Finally, as the main contribution of this research, the practical applications of this suggested framework design will widen the horizons of semantic Web-based knowledge comprehension and dissemination as its applications are not limited to the crime domain only, making a progressive step for the betterment of mankind.

References

1. Ghani ZA (2017) A comparative study of urban crime between Malaysia and Nigeria. *J Urban Manage* 6(1):19–29. <https://doi.org/10.1016/j.jum.2017.03.001>
2. Soh MB (2012) Crime and urbanization: revisited malaysian case. *Procedia Soc Behav Sci* 42:291–299. <https://doi.org/10.1016/j.sbspro.2012.04.193>
3. Badiora AI, Afon AO (2013) The spatial pattern of crime in Nigerian traditional city: the Ile-Ife experience. *Int J Criminol Sociol Theory* 6(3):15–28
4. Ajaegbu OO (2012) Rising youth unemployment and violent crime in Nigeria. *Am J Soc Issues Humanit* 2(5):315–321
5. Katsina AM (2013) Trend analysis of poverty and urban crime in Nigeria since 1999. *Int J Arts Commer* 1(2)
6. Kashyap V (2008) Ontologies and schemas. In: *The Semantic Web*, pp 79–135. https://doi.org/10.1007/978-3-540764526_5
7. Trokanas N, Cecelja F (2016) Ontology evaluation for reuse in the domain of process systems engineering. *Comput Chem Eng* 85:177–187. <https://doi.org/10.1016/j.compchemeng.2015.12.003>
8. Noy N, McGuiness D (2001) *Ontology development 101: a guide to creating your first ontology*. Stanford University, Stanford
9. Chergui W, Zidat S, Marir F (2018) An approach to the acquisition of tacit knowledge based on an ontological model. *J King Saud Univ Comput Inf Sci*. <https://doi.org/10.1016/j.jksuci.2018.09.0129>
10. Alavi M, Leidner DE (2001) Knowledge management and knowledge management systems: conceptual foundations and research issues. *Manag Inf Syst Q* 25, 107–136. <https://doi.org/10.2307/23045371>

- 2307/3250961; Anderson JR (1983) The architecture of cognition. Harvard University Press, Cambridge, MA
11. Kotis K, Lanzenberger M (2008) Ontology matching: current status, dilemmas and future challenges. In: 2008 international conference on complex, intelligent and software intensive systems. <https://doi.org/10.1109/cisis.2008.28>
 12. Gutierrez-Basulto V, Ibanez-Garcia Y, Kontchakov R, Kostylev EV (2015) Queries with negation and inequalities over lightweight ontologies. SSRN Electron J. <https://doi.org/10.2139/ssrn.3199213>
 13. Shvaiko P, Euzenat J (2005) A survey of schema-based matching approaches. J Data Semant IV
 14. Tartir S, Arpinar IB, Sheth AP (2010) Ontological evaluation and validation. Theory Appl Ontol Comput Appl 115–130. https://doi.org/10.1007/978-90-481-8847-5_5
 15. Berners-Lee T (2001) The semantic web (PDF). Scientific American
 16. Usman U, Yakubu M, Bello AZ (2012) An investigation on the rate of crime in Sokoto state using principal component analysis
 17. Spasic I, Ananiadou S, McNaught J, Kumar A (2005) Text mining and ontologies in biomedicine: Making sense of raw text. Brief Bioinform 6(3):239–251. <https://doi.org/10.1093/bib/6.3.239>
 18. Guha RV (2013) Light at the end of the tunnel. In: International semantic web conference 2013 keynote
 19. Davis I (2014) vocab.org—a URI space for vocabularies. Retrieved 16 Feb 2019 from <http://vocab.org/>
 20. Yu L (2007) Swoogle. In: Introduction to the semantic web and semantic web services, pp 145–157. <https://doi.org/10.1201/9781584889342.pt3>
 21. Linked Open Vocabularies (2013) Linked open vocabularies (LOV). Retrieved 6 Dec 6 2018 from <https://lov.linkeddata.es/dataset/lov/>
 22. Musen MA, The Protégé Team (2013) Protégé ontology editor. Encycl Syst Biol:1763–1765. https://doi.org/10.1007/978-1-4419-9863-7_1104
 23. Slater L, Gkoutos GV, Schofield PN, Hoehndorf R (2016) Using AberOWL for fast and scalable reasoning over BioPortal ontologies. J Biomed Semant 7(1). <https://doi.org/10.1186/s13326-016-0090-0>
 24. Faria D, Jiménez-Ruiz E, Pesquita C, Santos E, Couto FM (2014) Towards annotating potential incoherencies in bioportal mappings. Semant Web ISWC: 17–32. https://doi.org/10.1007/978-3-31911915-1_2
 25. W3C (2018) ConverterToRdf W3C Wiki. Retrieved 6 Dec 2018 from https://www.w3.org/wiki/ConverterToRdf#CSV,_Comma-Separated_Values.29
 26. Zenuni X, Raufi B, Ismaili F, Ajdari J (2015) State of the art of semantic web for healthcare. Procedia Soc Behav Sci 195:1990–1998. <https://doi.org/10.1016/j.sbspro.2015.06.213>
 27. Protege—DataMaster (2014) DataMaster—Protege Wiki. Retrieved 6 Dec 2018 from <https://protegewiki.stanford.edu/wiki/DataMaster>
 28. Abdul Jalil M, Ling CP, Noor NMM, Mohd F (2017) Knowledge representation model for crime analysis. Procedia Comput Sci 116:484–491. <https://doi.org/10.1016/j.procs.2017.10.067>
 29. Dzemydiene D, Kazemkaitiene E (2005) Ontology-based decision support system for crime investigation processes. In: Vasilecas O, Wojtkowski W, Zupancic J, Caplinskas A, Wojtkowski WG, Wrycza S (eds) Springer, United States
 30. Caldarola EG, Rinaldi AM (2016) An approach to ontology integration for ontology reuse. In: 2016 IEEE 17th international conference on information reuse and integration (IRI). <https://doi.org/10.1109/iri.2016.58>
 31. Tremblay G (2015) Cultural industries, creative economy and the information society. In: Power, media, culture. <https://doi.org/10.1057/9781137540089.0011>
 32. Ghorbel F, Ellouze N, Métais E, Hamdi F, Gargouri F, Herradi N (2016) MEMO GRAPH: an ontology visualization tool for everyone. Procedia Comput Sci 96:265–274. <https://doi.org/10.1016/j.procs.2016.08.139>

33. Ding Y, Sun Y, Chen B, Borner K, Ding L, Wild D, Wu M, DiFranzo D, Fuenzalida AG, Li D, Milojevic S Toma I (2010) Semantic web portal: a platform for better browsing and visualizing semantic data. In: Active media technology, pp 448–460. https://doi.org/10.1007/978-3-642154706_46
34. Habernal I, Konopík M (2013) SWSNL: semantic web search using natural language. Expert Syst Appl 40(9):3649–3664. <https://doi.org/10.1016/j.eswa.2012.12.070>
35. Yang P, Tang K, Yao X (2018) Turning high-dimensional optimization into computationally expensive optimization 22
36. Abeywardana PC, Kodituwakklu SR (2012) Ontology-based information extraction for disease intelligence. Int J Res Comput Sci 2(6):7–19. <https://doi.org/10.7815/ijores.26.2012.051>
37. Sowa JF (1999) Knowledge representation: logical, philosophical, and computational foundations. Brooks/Cole, Pacific Grove, CA
38. Smith B (2003) Ontology and information systems. SUNY at Buffalo, Buffalo, NY
39. Kishore R, Zang H, Ramesh R (2004) Computational ontologies and information systems foundations. In: Communications of the association for information systems 14. <https://doi.org/10.17705/1cais.01408>

Chapter 31

Attribute Reduction for Medical Data Analysis Using Rough Set Theory



Prerna Bhavsar, Parth Jhunjhunwala and Lynette D'Mello

1 Introduction

In recent years of research in the medical field, many different concepts of machine learning and data mining have been employed [1]. The decision making involved in medical diagnosis has a certain amount of inexactness and uncertainty. Therefore, techniques, which cannot handle incorrect or inaccurate data, cannot be used to perform clinical analysis because it may lead to a model, which may not consider these parameters. Rough set theory takes into account the inexactness of medical data and hence provides a better and more innovative approach towards clinical information extracting. In this paper, we focus on the concept of reduction of the attributes in our medical dataset using rough set theory in order to provide results with increased accuracy.

Pawlak first introduced rough set theory in the 1980s. He proposed the rough set theory as a mathematical tool to deal with vague concepts. Rough set theory is a state of development, which deals with uncertain or incomplete data, and knowledge, which helps in providing one of the first non-statistical approaches in data analysis. Rough set theory and fuzzy set theory are considered complementary to each other, which form the basis of soft computing. Both concepts deal with a different kind of vagueness and uncertainty in knowledge mining. Rough set is considered different from fuzzy logic due to one major factor; it deals with the classification of data using lower and upper approximations and boundary regions, whereas fuzzy set theory

P. Bhavsar · P. Jhunjhunwala (✉) · L. D'Mello
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: parthjhunjhunwala25@gmail.com

P. Bhavsar
e-mail: prernabhavsar8@gmail.com

L. D'Mello
e-mail: lynette.dmello@djsce.ac.in

deals with the partial membership of the elements in the sets and their rareness in them. Having said this, rough set theory provides an innovative approach to deal with feature selection and reduction of the data for the knowledge discovery in medical databases.

2 Literature Review

In this paper, our aim is to implement rough set theory on medical data analysis. In order to do this, we studied each and every concept of rough set theory and referred to different papers overtime on the use of rough set theory in medical informatics.

Durairaj and Sathyavathi [2], in their paper on rough set application on medical informatics, have explained in a very elaborate manner the use of rough set in medical data analytics. The rough set concepts have been used on the prediction of IVF success rate. Classification is done using standard voting and batch classifier. Further, it is explained why standard voting is better for their particular dataset. From their paper, we have extracted a good amount of information regarding rough set theory's application in the medical field.

Rissino and Lambert-Torres [3] have worked on the survey of rough set theory and their applications. In their paper, the concentration is on explanation of the concept of rough set theory with careful details and definition. It elaborates on the concepts of approximation, reduction of attributes, dependency of attributes and other rough set theory concepts to provide a rather detailed survey and analysis. This paper has contributed to our study as it has made us understand rough set theory in detail as well as with precise applications.

Chen et al. [4] in their work have focused on proposing an improved genetic algorithm for reduction of attributes in rough set theory. In the study, they have used the relative importance of chromosome to define the fitness function for the algorithm. The work explains the origin and use of genetic algorithm in depth along with its efficiency. From the paper, our study of genetic algorithm and its features was enhanced, and the algorithm is understood. It helped to build the basis for the use of genetic algorithm in our own study.

The study of attribute selection methods in rough set theory by Li [5] aimed at discussing the rough set theory and focusing on the selection measures to improve accuracy and completeness. The study formed the basis of our understanding of Johnson's algorithm and its accuracy. It deals with implementation of two reduction algorithms, the particle swarm optimization (PSO) and Johnson's algorithm. It explains Johnson's algorithm and also compares it with PSO and genetic algorithm giving careful observations and conclusions.

3 Rough Set Theory

3.1 Definition

In rough set theory, the information is represented in the form of tables called information systems. It consists of rows represented by object tuples and columns which are the attributes associated with the particular data. This information system is a decision table if it contains a class label, i.e. a decision attribute.

Rough set deals with imprecise information using boundary value conditions. Rough set can easily be defined by the concept of approximation as given by Pawlak.

3.2 Indiscernibility Relation

Indiscernibility relation is considered the central concept in rough set theory. It is considered as a relation between two objects or more, where all the attribute values are the same in relation to a subset of attributes that are taken into consideration. Indiscernibility relation is an equivalence relation, where all the similar objects are considered elementary [3].

3.3 Approximation

Indiscernibility is the starting point of rough set theory. Approximation is another important concept related to the rough set theory. It defines the technique by which the boundary regions are evaluated and consists of two types of approximations of the crisp set.

Let (T, A) define the information system such that U is the set of objects and A is the set of non-empty attributes. The indiscernibility of a subset S can then be defined by as follows:

$$\text{IND}(S) = \{(x, y) \in U | \forall a \in P \text{ and } a(x) \neq a(y)\} \quad (1)$$

3.3.1 Upper Approximation

It is a set of elements that possibly belong to the target set, i.e. the elements that may or may not belong to the set.

$$\overline{A}(X) = U\{Y \in U | A : Y \cap X \neq \emptyset\} \quad (2)$$

3.3.2 Lower Approximation

It is a set of elements that certainly belong to the target set. These elements completely belong to the set.

$$\underline{A}(X) = U \{Y \in U | A : Y \subseteq X\} \quad (3)$$

3.3.3 Boundary Region

Boundary region B is considered as the set of objects that neither belongs completely to the target set nor are they thoroughly outside it. These objects can be described as the objects that are vaguely present in the target set. If the boundary region is an empty set, then it is called a crisp set or if it is a non-empty set, then it is called as a rough set.

$$\text{Positive Boundary } B = \underline{A} \quad (4)$$

$$\text{Negative Boundary } B = U - \overline{A} \quad (5)$$

$$\text{Positive Boundary BR} = \overline{A} - \underline{A} \quad (6)$$

3.4 Reduction of Attributes

There may be several indiscernible objects multiple times in the information system that may be superfluous and redundant. These redundant objects can be reduced to find subsets of objects that maintain the indiscernibility relation and also preserve the set approximation. There are several such subsets but the ones that provide the minimal set are called reducts. Reducts are the feature set that are unique and wholly represent the knowledge of the databases. Rough set theory makes use of reducts in order to isolate the key attributes which affect the final outcomes of the classification system.

3.5 Application

Rosetta is a rough set toolkit for analysis of data [6]. Rosetta has been developed by two groups: Knowledge Systems Group Norwegian University of Science and Technology, Trondheim, Norway, and the Group of Logic, Inst. of Mathematics, University of Warsaw, Poland under the guidance of Komorowski and Skowron [3, 7].

It is a software application that is used for the implementation of rough set theory concepts. It has various algorithms for finding reducts and association rules. Rosetta can then be used to classify the databases on the rules generated by the reducts. It implements other features like discretization algorithms, rule pruning, reduct computation, classifiers and analysis. Rosetta provides a very user-friendly GUI and is mainly used for modelling based on discernibility. In this paper, we have employed the use of Rosetta for finding reducts and then the classification of the medical data.

4 Implementation

4.1 Proposed Model

The proposed model to the implementation of rough set theory on the Indian Liver Patient Dataset is illustrated in Fig. 1.

4.2 Dataset Description

The dataset used is an Indian Liver Patient Dataset taken from University of California at Irvine (UCI) Machine Learning Repository [8]. It has comma-separated values containing 10 attributes and 583 instances of patient records. This dataset contains a total of 441 male patients and 142 female patients out of which 416 are liver patients and 167 are non-liver patients. The dataset is split into the training and testing sets

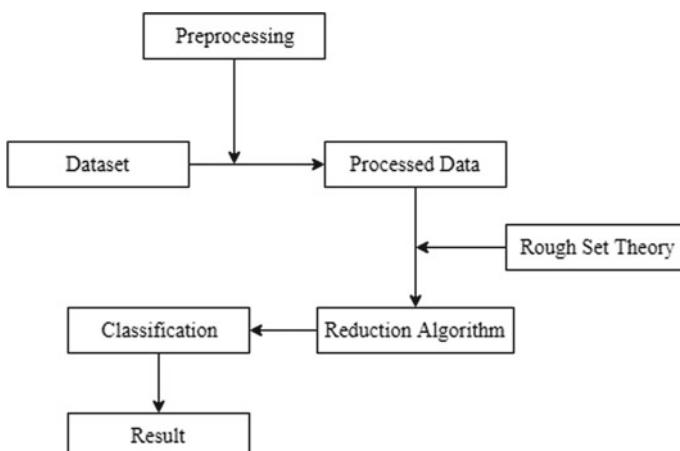


Fig. 1 Proposed model

with a split factor of 0.8. Thus, the training set consists of 466 tuples and the testing set consists of 116 tuples.

4.3 Data Preprocessing

The dataset needed preprocessing in dealing with missing values for the albumin and globulin ratio attribute. Data was preprocessed using Weka. Weka is a machine learning software containing tools for data preprocessing, classification, clustering, generating association rules and visualizing the data. Using Weka, missing values were replaced by the modes and means of the dataset.

4.4 Reduction Process

We have used the following two algorithms to find reducts for the training data. RSES is Rough Set Exploration System. It is a toolkit used for the analysis of data based on different methodologies and algorithms from the rough set theory concepts.

4.4.1 Johnson's Algorithm

Johnson's algorithm is a greedy algorithm which is used to find the shortest path between all pairs of vertices in a graph. RSES Johnson Reducer is an advanced version of a simple Johnson's algorithm. In the case of rough set theory, it is used as a reduction algorithm to evaluate and produce reducts that contain the attributes most frequently used for decision making. It, thus, considers the attribute that appears as the most significant one. It usually leads to finding an optimal solution (Fig. 2).

4.4.2 Genetic Algorithm

Genetic algorithm globally optimizes throughout the dataset without depending on specific areas. It is widely used for finding reducts and mimic the behaviour of natural evolution. RSES Genetic Reducer implements a variant of the genetic algorithm and goes on till the maximum number of reductions is noticed. Attribute reduction algorithms based on rough set theory and genetic algorithm tend to give an optimal solution for the attribute reduction results (Fig. 3).

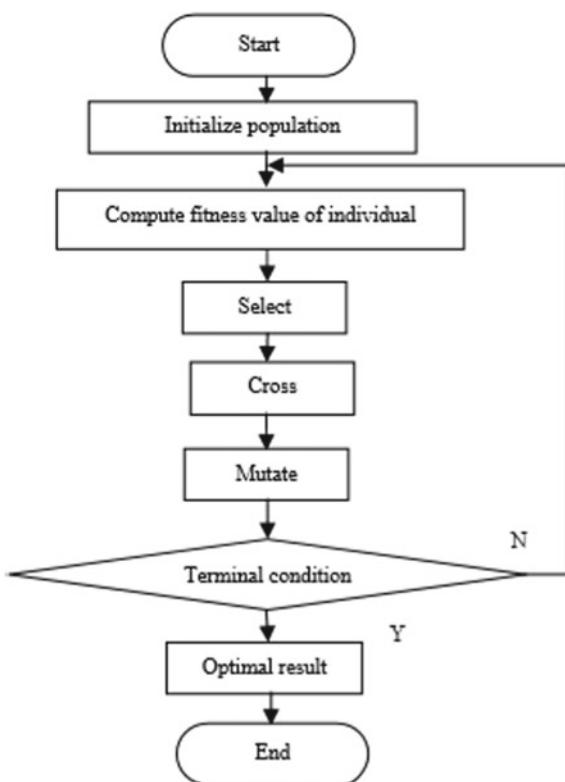
Fig. 2 Johnson's algorithm
[2]

Johnson algorithm
Johnson(C,fD)

C, the set of conditional attributes
fD, the discernibility function.

- (1) R \emptyset ; bestc=0;
- (2) while(fDnot empty)
 - (3) for each a \in C that appears in fD
 - (4) c= heuristic(a)
 - (5) if(c >bestc)
 - (6) bestc=c; bestAttr_a
 - (7) R \sqcup R \cup a
 - (8) fD_removeClauses(fD, a)
- (9) returnR

Fig. 3 Genetic algorithm [4]



4.5 Classification

In our study, we have implemented the classification of our testing data using both the above algorithms. The classification method used is standard voting. Standard voting takes into consideration the rules generated by the implementation of the rough set reduction algorithms. For both algorithms, we have classified the data for each fallback factor. Fallback is the classification assigned if a classifier indicates more than one classification. In our case, there were two values for the decision variable, i.e. 1 or 2. Thus, these two are used as the fallback factor for the classification.

5 Results and Discussions

Reduction using both Johnson's and genetic algorithm is done on the training set producing respective reducts and reduction rules. These rules are then used to classify the testing data using standard voting (Fig. 4).

Reduction using Johnson's algorithm generated 20 reduct sets and genetic algorithm produced 148 reduct sets. Some of them are as illustrated in Fig. 5.

These reduct sets are then used by ROSETTA to generate reduction rules for each of the algorithms. The reduction rules elaborate on the support, accuracy, coverage

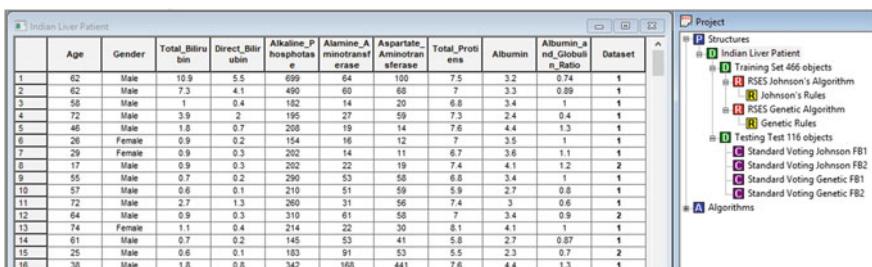


Fig. 4 Dataset and project tree

The figure shows two side-by-side tables from the ROSETTA interface. The left table, titled 'RSES Johnson's Algorithm', lists 16 reduct sets. The right table, titled 'RSES Genetic Algorithm', lists 16 reduct sets. Both tables include columns for 'Reduct', 'Support', and 'Length'.

	Reduct	Support	Length
1	(Total_Bilirubin)	1	1
2	(Direct_Bilirubin)	1	1
3	(Age)	1	1
4	(Gender, Age)	1	2
5	(Aspartate_Aminotransferase)	1	1
6	(Direct_Bilirubin, Age)	1	2
7	(Alkaline_Phosphotase)	1	1
8	(Alanine_Aminotransferase)	1	1
9	(Albumin_and_Globulin_Ratio)	1	1
10	(Alanine_Aminotransferase, Age)	1	2
11	(Total_Bilirubin, Age)	1	2
12	(Alkaline_Phosphotase, Age)	1	2
13	(Total_Proteins)	1	1
14	(Aspartate_Aminotransferase, Age)	1	2
15	(Albumin_and_Globulin_Ratio, Total_Bilirubin)	1	2
16	(Albumin)	1	1

	Reduct	Support	Length
1	(Direct_Bilirubin)	1	1
2	(Aspartate_Aminotransferase)	1	1
3	(Alkaline_Phosphotase)	1	1
4	(Total_Bilirubin)	1	1
5	(Albumin_and_Globulin_Bilirubin)	1	1
6	(Alanine_Aminotransferase, Total_Proteins)	1	2
7	(Alanine_Aminotransferase, Albumin)	1	2
8	(Total_Proteins, Albumin)	1	2
9	(Age, Total_Proteins)	1	2
10	(Age, Alanine_Aminotransferase)	1	2
11	(Total_Bilirubin, Total_Proteins)	1	2
12	(Total_Bilirubin, Albumin)	1	2
13	(Total_Bilirubin, Alanine_Aminotransferase)	1	2
14	(Age, Albumin)	1	2
15	(Age, Total_Bilirubin)	1	2
16	(Alkaline_Phosphotase, Aspartate_Aminotransferase)	1	2

Fig. 5 Reduct sets for Johnson's and genetic algorithm

	Rule	LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stability	LHS Length	RHS Length
1	Total_Bilirubin(10.9) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
2	Total_Bilirubin(3.9) => Dataset(1)	2	2	1.0	0.004292	0.006079	1.0	1	1
3	Total_Bilirubin(18.4) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
4	Total_Bilirubin(3.1) => Dataset(1)	2	2	1.0	0.004292	0.006079	1.0	1	1
5	Total_Bilirubin(8.9) => Dataset(1)	3	3	1.0	0.006438	0.009119	1.0	1	1
6	Total_Bilirubin(2.8) => Dataset(1)	2	2	1.0	0.004292	0.006079	1.0	1	1
7	Total_Bilirubin(2.4) => Dataset(1)	4	4	1.0	0.008584	0.012158	1.0	1	1
8	Total_Bilirubin(5.7) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
9	Total_Bilirubin(8.6) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
10	Total_Bilirubin(5.2) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
11	Total_Bilirubin(3.8) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
12	Total_Bilirubin(6.6) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
13	Total_Bilirubin(5.3) => Dataset(2)	2	2	1.0	0.004292	0.014599	1.0	1	1
14	Total_Bilirubin(12.7) => Dataset(1)	2	2	1.0	0.004292	0.006079	1.0	1	1
15	Total_Bilirubin(15.9) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1
16	Total_Bilirubin(18) => Dataset(1)	1	1	1.0	0.002146	0.00304	1.0	1	1

Fig. 6 Reduction rules for Johnson's algorithm

	Rule	LHS Suppo	RHS Suppo	RHS Accura	LHS Covera	RHS Covera	RHS Stabilit	LHS Length	RHS Length
1	Direct_Bilirub	1	1	1.0	0.002146	0.00304	1.0	1	1
2	Direct_Bilirub	2	2	1.0	0.004292	0.006079	1.0	1	1
3	Direct_Bilirub	3	3	1.0	0.006438	0.009119	1.0	1	1
4	Direct_Bilirub	10	10	1.0	0.021459	0.030395	1.0	1	1
5	Direct_Bilirub	1	1	1.0	0.002146	0.00304	1.0	1	1
6	Direct_Bilirub	2	2	1.0	0.004292	0.006079	1.0	1	1
7	Direct_Bilirub	9	9	1.0	0.019313	0.027356	1.0	1	1
8	Direct_Bilirub	2	2	1.0	0.004292	0.006079	1.0	1	1
9	Direct_Bilirub	5	5	1.0	0.01073	0.015198	1.0	1	1
10	Direct_Bilirub	1	1	1.0	0.002146	0.00304	1.0	1	1
11	Direct_Bilirub	3	3	1.0	0.006438	0.009119	1.0	1	1
12	Direct_Bilirub	3	3	1.0	0.006438	0.009119	1.0	1	1
13	Direct_Bilirub	1	1	1.0	0.002146	0.00304	1.0	1	1
14	Direct_Bilirub	5	5	1.0	0.01073	0.015198	1.0	1	1
15	Direct_Bilirub	1	1	1.0	0.002146	0.00304	1.0	1	1
16	Direct_Bilirub	2	2	1.0	0.004292	0.006079	1.0	1	1

Fig. 7 Reduction rules for genetic algorithm

and length. Johnson's algorithm produced 356 reduction rules, while genetic algorithm produced 6635 reduction rules, some of which are illustrated in Figs. 6 and 7, respectively.

These reduction rules are used for classification. A confusion matrix for the following classification is obtained as the output. A confusion matrix is a table which describes how well the test data is classified in comparison to the training data whose values are known. It calculates the final accuracy for each classification and thus allows for analysis of the data. Each column of a confusion matrix represents a predicted class, while each row represents the actual class. It reports the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives(FN). The accuracy of the classification is then calculated (Fig. 8).

The confusion matrix for each algorithm with different fallback values is as shown in Figs. 9 and 10.

Genetic algorithm produced reduct sets of length greater and larger in number than Johnson's. We further derive conclusion from the confusion matrix that even though Johnson's algorithm derived fewer reduct sets; it produced similar results as

Fig. 8 Confusion matrix

		Predicted	
		False	True
Actual	False	TN	FP
	True	FN	TP

The figure consists of two side-by-side windows titled "Standard Voting Johnson FB1" and "Standard Voting Johnson FB2". Both windows display a 2x2 confusion matrix table.

Standard Voting Johnson FB1 Data:

		Predicted	
		1	2
Actual	1	79	7
	2	23	7
	0.77451	0.5	0.741379

Standard Voting Johnson FB2 Data:

		Predicted	
		1	2
Actual	1	51	35
	2	11	19
	0.822581	0.351852	0.603448

Fig. 9 Confusion matrix for Johnson's algorithm with fallback factor 1 and 2, respectively

The figure consists of two side-by-side windows titled "Standard Voting Genetic FB1" and "Standard Voting Genetic FB2". Both windows display a 2x2 confusion matrix table.

Standard Voting Genetic FB1 Data:

		Predicted	
		1	2
Actual	1	76	10
	2	20	10
	0.791667	0.5	0.741379

Standard Voting Genetic FB2 Data:

		Predicted	
		1	2
Actual	1	76	10
	2	20	10
	0.791667	0.5	0.741379

Fig. 10 Confusion matrix for genetic algorithm with fallback factor 1 and 2, respectively

genetic algorithm. Thus, as Johnson's algorithm produced similar results with less number of reduction rules and length, it outperforms genetic algorithm.

Even though Johnson's algorithm gives accuracy lower than genetic algorithm for fallback 2, as Johnson's algorithm produces a minimal set of reducts more effectively than Genetic algorithm. Thus, it is a better algorithm for reduction. After our careful examination, we have come to the conclusion that as the size of the dataset increases, accuracy of Johnson's algorithm increases and produces better results than genetic algorithm. As Johnson's algorithm uses lesser reducts, it is more efficient and faster. The rules generated by Johnson's keep all the information in the training set and thus features discernibility.

6 Conclusion

Rough set is a useful mathematical tool which produced optimal results for large amount of data. It is helpful in extraction of information and analysis from seemingly ambiguous and large data. In this research work, different reduction algorithms were to produce reduct sets. These reduct rules were then used to classify the medical data and accuracy is calculated. Weka toolkit was implemented to pre-process the data and deal with missing values. ROSETTA toolkit, which deals with applications of roughs set, was used to perform Johnson's and genetic algorithm for the reduction process.

Standard voting is then implemented using ROSETTA for classification and accuracy is calculated. The results show that Johnson's algorithm performs better than genetic algorithm and produces fewer reduction rules for the classification. Further, we can also observe that the results of classification will have better accuracy if the size of the data is increased. Johnson's algorithm is the fastest reduct algorithm and also guarantees discernibility.

References

1. Nahato KB, Harichandran KN, Arputharaj K (2015) Knowledge mining from clinical datasets using rough sets and backpropagation neural network. *Comput Math Methods Med* 460189:13
2. Durairaj M, Sathyavathi T (2013) Applying rough set theory for medical informatics data analysis. *Int J Sci Res Comput Sci Eng* 1(5):1–8
3. Rissino S, Lambert-Torres G (2009) Rough set theory fundamental concepts, principals, data extraction, and applications. <https://doi.org/10.4304/jsw.9.9.2276-2282>
4. Chen L, Liu H, Wan Z (2014) An attribute reduction algorithm based on rough set theory and an improved genetic algorithm. *J Softw* 9. <https://doi.org/10.4304/jsw.9.9.2276-2282>
5. Li X (2014) Attribute selection methods in rough set theory. In: Doctoral dissertation. San José State University, San Jose. <http://archive.ics.uci.edu/ml>
6. Abbas Z, Burney A (2016) A survey of software packages used for rough set analysis. *J Comput Commun* 4:10–18. <https://doi.org/10.4236/jcc.2016.49002>
7. Øhrn A, Komorowski J (1997) ROSETTA: a rough set toolkit for analysis of data. In: Proceedings of third international joint conference on information sciences, fifth international workshop on rough sets and soft computing (RSSC'97), vol 3, pp 403–407. Durham, NC, USA, 1–5 Mar 1997
8. Dua D, Graff C (2019) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. <http://archive.ics.uci.edu/ml>

Chapter 32

Emotion Identification Using CNN-Based Transfer Learning



Aarti M. Karnade, Prachi Dalvi and D. R. Kalbande

1 Introduction

Image classification is the necessity of today's security system. Humans can classify images very easily, but in the case of computers, it is a very complicated task. Each image is made up of set of pixels represented with different values. This paper focuses on convolutional neural network (CNN), which uses a machine learning algorithm for automatic classification of the images [1].

1.1 Image Classification

Image classification helps to differentiate all pixel points from the image matrix as per classes or any themes. Classification includes processes such as preprocessing, detection of object, training, and classification of the objects. Image classification is used for finding trends in images from large databases. Following are the techniques used for image classification.

Pixel based: Image can be classified based on independent features (e.g., color, texture, and shape) or based on domain specific features (e.g., dependent features like human faces, fingerprints, and conceptual features). Image pixels are analyzed

A. M. Karnade (✉) · P. Dalvi · D. R. Kalbande
Sardar Patel Institute of Technology, Mumbai, India
e-mail: aartimkarande@spit.ac.in

P. Dalvi
e-mail: prachi_dalvi@spit.ac.in

D. R. Kalbande
e-mail: drkalbande@spit.ac.in

by the spectral information. Color features can be robustness, effectiveness, implementation simplicity, computational simplicity, and low storage requirements. Different pixel-based classifications are maximum likelihood, minimum-distance-to-mean, and minimum Mahalanobis distance.

- Unsupervised classification: It extracts color information from the image by labeling the pixels of the image to different classes. Large amounts of unlabeled data can be used to pre-train the network's parameters. This can be further fine-tuned.
- Supervised classification: It analyzes and trains the classifier on the labeled images and extracting features from them. Maximum likelihood is a supervised image classification technique in which the probability value of pixels is used for classifying the pixels [2]. Large amount of reliable labeled data is needed by supervised methods, which may be complicated to obtain in many domains. Therefore, it is desirable to use an unsupervised learning strategy.

1.2 *Image Feature Extraction*

Image content exploration is effectively done by hierarchical decomposing the images into a series of semantic components or as per the feature extraction. This semantic image content (structure and texture) can be matched with other images features. Following are the techniques used for image feature extraction [3].

- Spatial pooling: It divides image into different subregions and computes feature vector of each subregion. Final image representation is a concatenation of all subregion feature vectors to classify the image.
- Histogram encoding: It uses local descriptor constructed using K-means from image. Then, each descriptor is assigned to one of the visual words which indices are accumulated in a histogram.
- Fisher vector encoding: This description vector is the gradient of the image's probability with respect to the feature distribution. FV describes a set of descriptors deviates from an average distribution of descriptors, modeled by a parametric generative model.

1.3 *Facial Emotion*

Facial emotion is a significant clue for evaluation of human affective behavior. While various methods have been projected for vision-based facial emotion recognition, the majority of them focus on emotion recognition based on static [4]. Limited variation in appearance and pose is described by the digit image and face task classification. Therefore, these two domains are relevant to our task, and the applied techniques can be proficiently used to traffic sign classification task. Due to the current growth

as well as a broad use of smart phones, services, and applications, emotion recognition is becoming a necessary part of providing emotional aid to people. It is very challenging to develop a system which can collect, analyze, and process emotional communications in real time and highly accurate manner with a minimal computation time. However, research on the human visual system has demonstrated that temporal information can be taken into the account for better judgment of the facial motion [5]. A hidden Markov model is utilizing the dynamics of facial emotion [6]. This classification divides image into blocks. Then, feature vector is applied on each block by grouping statistics.

- Support vector machine classifier: By using a linear boundary, binary classifier separates the classes. It optimizes the use of training data; it also speedup the classifier. It also decreases the classification errors which occur due to prior assumptions on the unsupervised data
- Dynamic Bayesian networks [7]: It represents a set of image pixel variables in the form of nodes on a directed acyclic graph. It is used to map the conditional independencies of these variables.
- Geometrical displacement [8]: This is used to change the size of the images, i.e., zooming or cropping the image.
- Dynamic texture descriptors [9]: It can be used in discovering the similarities between images in multimedia databases. Shape features can be circularity, aspect ratio, discontinuity angle irregularity, length irregularity, complexity, sharpness, etc. [10].

2 Transfer Learning

2.1 Convolutional Neural Network (CNN)

CNN is artificial neural networks designed to automatically and adaptively learn spatial hierarchies of features. It handles it though backpropagation algorithm by using layers, such as convolution layers, pooling layers, and fully connected layers. CNN is based on supervised deep learning approach which requires large labeled data for training on the network. CNNs use preprocessing for selecting the features from original input using filters. Based on the feature map, output from pooling weights is improved for the accuracy. Suppose that two-dimensional image “ I ” is taken as the input, and the two-dimensional convolution kernel is represented by “ K ;” the convolution of the input image is as follows:

$$S(I, j) = (I * K)(I, j) = \sum_m \sum_n I(m, n) \cdot K(i - m, j - n) \quad (1)$$

Equation 1 is the convolution of the input image.

Sparse connectivity and parameter sharing are the two important concepts of CNN's convolution [10, 11]. By reducing the necessary memory resources and computational operations and to increase efficiency, the sparse connectivity is used. Parameter sharing allows to use the same parameter in several parts of a model. The features in a specific location from the image are not allowed to learn by the convolution, but it can extract all the features from all areas of image data.

2.2 *Working of CNN for Image Classification*

To process an image, CNN uses neural network made up of convolution layers, subsampling layers, pooling layers, and softmax layers. These layers use activation function ReLU for the classification. Convolutional layers are considered as a feature detector. CNN's subsampling layers are used to perform input dimension reduction by preserving the information of the image. CNN uses maximum pooling for accurate prediction. Feature detection is based on scanning the input image with the filter of a given size. It applies matrix computations in order to derive a feature map. Pooling layer works on every input neural network and resizes it spatially based on the MAX operation for generating the reduced output.

2.3 *Transfer Learning*

It is one of the machine learning techniques in which a model is trained and developed for some task and is then reapplied on another related task. It transfers knowledge known as source domain to the target domain. This is normally used when there is a new dataset is smaller than the original dataset for training the pre-trained model. Based on the preliminary training, transfer learning reuses the learned features on the source dataset and modifies these features to suit the new dataset instead of starting the learning process on the data done using random weight initialization in the network. This concept saves time and resources for feature identification. It targets to convey knowledge between related source and target domains. It tries to conquer the discrepancy of training samples for some categories by adapting classifiers trained for other categories. The performance for the target domain or task may improve by it [12].

2.4 *CNN with Transfer Learning*

Conventional transfer learning algorithms consider changing any representation of the data or adapting classifiers or both of them [13] to improve the divergences across two domains, i.e., source and target. Natural image datasets are used to pre-train the

CNN models. Decoding pathway is useful for the supervised learning to achieve a better optimum [14].

3 Methodology

3.1 Existing Methodology

For predicting the six basic emotions of human facial expression, usually supervised learning model will use the one-versus-all (OVA) approach to train and predict. When performing these operations on faces, existing methodology proceeds with the one-versus-all (OVA) approach for every emotion, where all non-target emotion training samples will be gathered together.

3.2 Propose Methodology

Here, we propose the feature extraction and facial expression recognition method using a transfer learning. The embedded camera of a desktop machine captured a facial video. After applying a proposed technique on that video, anyone can easily extract the emotions from the human faces. It is basically a face detection, feature extraction, and recognition technique. These are some important phases of it:

- Face detection
- Feature extraction
- Reusing the existing features to the next feature
- Features recognition.

Steps of execution:

1. **Capture the image:** from the video using OpenCV-Python
2. **Preprocessing the image:** Capture image is used as an input by the system. Then, input image is resized to 3*3 window sizes in order to apply further processing.
3. **Face detection using skin segmentation:** The processed image is used as input and skin region is mined with the help of YCbCr color space. Process mainly uses the skin color in order to distinct the face region. Extra statistics are removed to obtain the facial features region.
4. **Feature extraction:** This phase is again classified into two parts.

4.1. Eyes detection:

To extract the eyes from the human face, the information of dark pixels on the human face is needed.

As the colors of the eyes are different from the skin, so by using a color space and separating the colors, we locate the eyes. The spotted eye region is then marked, and the eye is extracted from the image and stored to the database. The support vector

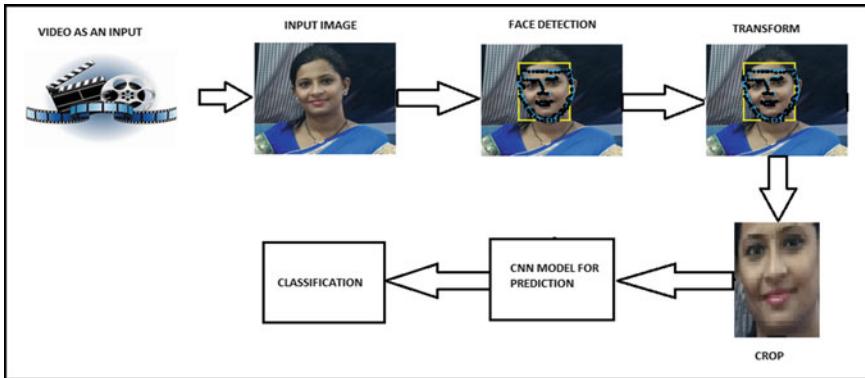


Fig. 1 Proposed methodology

machine (SVM) algorithm is used for it. Above database of eye, later transfer and mapped with the new eye pattern.

4.2. Lips detection: Three lip points are extracted from the face for detecting the lips. These are some points which help

- A. Lip's center point
- B. Lip's left corner point
- C. Lip's right corner point

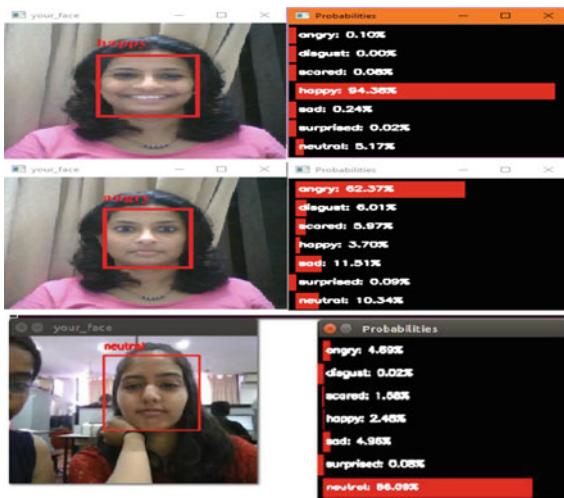
Hidden Markov models (HMMs) have proved to be reasonably flexible modeling tool for this purpose. This extracted lip pattern is also stored in one database for further model learning purpose

5. **Reusing the existing features to the next feature:** Later, the database containing the crop transformed images is used as an input to train the CNN model for getting the better prediction.
6. **Feature recognition:** Feature recognition stage is also called as classifications which can also deal with extracted data and group them according to certain parameters. Classification is applied to identify, which of a set of categories a new observation, i.e., extracted feature belongs, classification is applied. On the basis of training, a set of database contains the observations whose category association is famous (Fig. 1).

4 Areas of Application

In the era of fast development of technologies, it is a need to develop an intelligent system that can recognize human emotion. Facial emotion recognition is an active area of research with the following several fields of applications. Some of the significant applications are [15].

Fig. 2 Facial expressions and corresponding parameters



- Alert system while driving.
- Social robot emotion recognition system.
- Medical practices.
- Mental state recognition.
- Automatic counseling system.
- Face expression synthesis.
- Light or music as per mood.
- In understanding human behavior.
- In interview.

5 Result

Proposed methodology was applied on Japanese Female Facial Expression (JAFFE) database, with 213 images while seven emotions (angry, happy, sad, scared, disgust, natural, surprise) classes [16].

In Fig. 2, the results steadily approve that the proposed methodology can guide to be further robust for facial expression study.

6 Conclusion

Deep efforts have been made over the past two decades in academia as well as research to find more robust methods of evaluating honesty and credibility during human interactions. With transfer learning, efforts have been made to catch human

emotions of anyone using significantly less data, then we would essential if we had to train from scratch. Emotions are due to any motion in brain, and it is recognized using the face. The main objective of this research paper is to give a brief introduction of new methodology of automatic emotion recognition system.

References

1. Ramprasath M, Anand MV, Hariharan S (2018) Image classification using convolutional neural networks. *Int J Pure Appl Mathe* 119(17):1307–1319. ISSN: 1314-3395
2. Chora RS (2007) Image feature extraction techniques and their applications for CBIR and biometrics systems. *Int J Biol Biomed Eng* 1(1)
3. Jmour N, Zayen S, Abdelkrim A (2018) Convolutional neural networks for image classification. IEEE. 978-1-5386-4449-2/18/C2018
4. Hossain MS, Muhammad G An emotion recognition system for mobile applications, digital object identifier. <https://doi.org/10.1109/access.2017.2672829>
5. Shoaieilangari S, Yau WY, Nandakumar K, Li J, Teoh EK (2015, July) Robust representation and recognition of facial emotions using extreme sparse learning *IEEE Trans Image Process* 24(7)
6. Wehrle T, Kaiser S, Schmidt S, Scherer KR (2000) Studying the dynamics of emotional expression using synthesized facial muscle movements. *J Pers Soc Psychol* 78(1):105–119
7. Aleksic PS, Katsaggelos AK (2006) Automatic facial expression recognition using facial animation parameters and multi stream HMMs. *IEEE Trans Inf Forensics Secur* 1(1):3–11
8. Zhang Y, Ji Q (2005) Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Trans Pattern Anal Mach Intell* 27(5):699–714
9. Kotsia I, Pitas I (2007) Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Trans Image Process* 16–1:172–187
10. Zhao G, Pietikainen M (2007) Dynamic texture recognition using localbinary patterns with an application to facial expressions. *IEEE Trans Pattern Anal Mach Intell* 29–6:915–928
11. Lee SJ, Chen T, Yu L, Lai CH (2018) Image classification based on the boost convolutional neural network. IEEE. *Trans Content Min* 6. <https://doi.org/10.1109/access.2018.27967222169-3536>
12. Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. *CVPR*
13. Ding Z, Shao M, Fu Y (2016) Transfer learning for image classification with incomplete multiple sources. IEEE. 978-1-5090-0620-5/16/\$31.00c 2016
14. Face recognition https://www.researchgate.net/publication/236953573_Face_Recognition_Based_on_Facial_Features
15. Dubey M, Singh L Automatic emotion recognition using facial expression: a review. *Int Res J Eng Technol (IRJET)*
16. The Japanese Female Facial Expression (JAFFE) Database <http://www.kasrl.org/jaffe.html>

Chapter 33

Secure and Decentralized Academic Transcript System Based on Blockchain Technology



Jalla Manikanta Swamy and Keyur Parmar

1 Introduction

When a student applies for a higher study in an academic institute or a job in an organization, the concerned institute/organization needs to verify the information (e.g., grades, affiliated institute, etc.) provided by the student. The hard copies of degree certificates, grade-sheets, etc., submitted by students are not tamperproof. Therefore, to verify the information provided by the student, the concerned institute/organization generally asks the student to produce a transcript in a sealed cover (of the affiliated/parent institute of the student). A “transcript” is a document that validates the student’s academic records, that is, courses taken and grades received by the student, degrees conferred to the student, etc. The student contacts his/her parent institute (where he/she is/was affiliated) and pays required fees to obtain a transcript in a sealed cover to be communicated to the concerned institute/organization where a student has applied for a higher study or job. In order to process each transcript, an institute needs to collect and verify the records and communicate the transcript to other institutes/organizations. The process not only requires significant man-hours to collect and verify the records, but it also requires funds to send the transcript to other institutes/organizations. The process needs to be repeated for each institute/organization where the student applies for a higher study or job to ensure the integrity of records. The process is tedious, and it affects students and institutes financially and in terms of man-hours. In addition, the hard copy of transcript can be easily manipulated by insider and outsider adversaries.

J. M. Swamy · K. Parmar (✉)
Indian Institute of Information Technology, Vadodara, India
e-mail: keyur@iitvadodara.ac.in

J. M. Swamy
e-mail: 201552068@iitvadodara.ac.in

In this paper, we propose an academic transcript system based on blockchain technology. In the proposed system, when a student requests the institute (where he/she is affiliated) to provide a transcript, the institute processes (e.g., collect the records, verify the records, etc.) and uploads the transcript in the blockchain. If the transcript is available in the blockchain, it can be seamlessly accessed by institutes and organizations in which the student applies for a higher study or job. As mentioned above, in the proposed system, the transcript of each student needs to be processed only once, and hence, it significantly reduces the overall cost of processing transcripts and manpower required to process transcripts. In addition, in the conventional system, the transcript needs to be physically communicated (e.g., by post) to each institute/organization where a student applies for a higher study or job. Hence, the conventional system not only adds the cost of communicating transcripts, but it also introduces communication delay. In the proposed system, transcripts are stored in the blockchain, and hence, they can be seamlessly accessed without any communication delay and extra cost. Cryptographic algorithms, namely KECCAK-256—a cryptographic hash function [1] and elliptic curve digital signature algorithm (ECDSA)—a digital signature algorithm [2], ensure the security of transcripts stored in the distributed and immutable transaction ledger of the proposed academic transcript system. We design a smart contract, i.e., self-executable code without a trusted intermediary, to upload the transcripts in the Ethereum blockchain. In addition, we created a decentralized application (DApp), i.e., an application that runs on the distributed computing system, to interact with the Ethereum blockchain [3]. The proposed academic transcript system uses Truffle [4] and Ganache [4] to deploy and test the smart contract running on Ethereum blockchain. The prototype of the proposed system validates the feasibility of the blockchain-based academic transcript system.

The rest of the paper is organized as follows. In Sect. 2, we discuss the state-of-the-art literature related to the proposed system. In Sect. 3, we propose the academic transcript system based on blockchain technology. Section 4 provides implementation details and the analysis of results. In Sect. 5, we conclude the paper by emphasizing our contributions.

2 Related Work

In 2008, an author with pseudonym, Satoshi Nakamoto [5], proposed a seminal paper on a peer-to-peer electronic cash system, namely Bitcoin. The central objective of the article is to facilitate the exchange of cash without any trusted intermediary such as the traditional banking system. In order to achieve the objective, records of cash exchanges are stored in a distributed database and secured by the use of cryptographic algorithms such as the hash function and the digital signature. In order to ensure the security (or precisely the integrity) of records, the records are stored as a chain of blocks popularly referred to as the blockchain. The use of blockchain is not only

limited to crypto-currencies such as Bitcoin, but it is also used in various other applications such as healthcare [6], Internet of Things (IoT) [7], and supply chain [8].

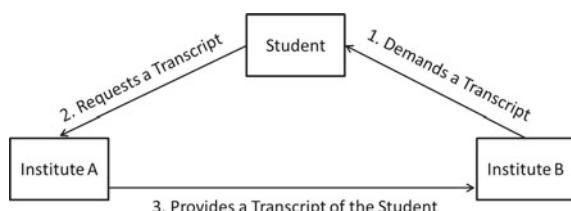
The major applications of blockchain technology use it to enable a smart contract. A smart contract, as proposed by Szabo [9], is a software program that enforces the terms of the contract without any trusted intermediary. The smart contracts use the consensus protocols (e.g., the proof of work protocol of Bitcoin [5]) to ensure the correct execution of the terms of the contract. The major advantage of using a smart contract is that it significantly reduces the transaction costs.

Ethereum [3, 10] is an open-source blockchain platform that executes smart contracts. A smart contract of the proposed academic transcript system runs on the Ethereum virtual machine (EVM). Ethereum has its own crypto-currency, namely Ether. Ethers are required to execute smart contracts in the EVM. Ethereum measures the EVM resource usage in terms of the amount of “gas” required to execute the operations of a smart contract. In order to write a smart contract, there exist different programming languages such as Solidity [11]. In addition, there exist tools, such as Truffle Suite [4] that help developers to build, test, and deploy distributed applications (DApps). A distributed application (DApp) facilitates the direct interaction among the entities interested to execute the smart contract. The DApp directly connects a smart contract to the Ethereum blockchain. Ganache [4] is a private Ethereum blockchain that enables developers to deploy and test the smart contract free of charge.

3 The Proposed Academic Transcript System Based on Blockchain Technology

As shown in Fig. 1, when a student applies for a higher study or job in some institute (institute B) or organization, that institute/organization needs to verify the documents (e.g., grade-sheets, degree certificates, etc.) submitted by the student. Hard copy of documents is not tamperproof, and they can be manipulated by adversaries. Hence, in order to verify the documents, an institute B approaches a student (or sometimes his/her institute directly) and demands a transcript in a sealed cover of an institute A (where the student is/was affiliated). A student requests a transcript in a sealed cover from the institute (institute A) where he/she is/was affiliated. In order to process the transcript, an institute A has to collect and verify the documents of the student each

Fig. 1 Conventional academic transcript system

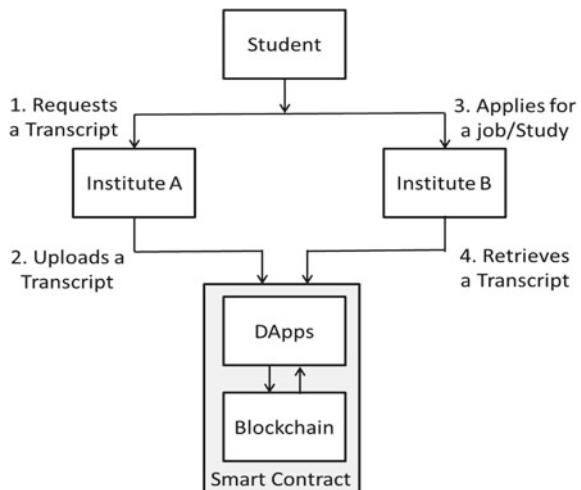


time he/she requests a transcript. In the end, an institute A sends the transcript of a student in a sealed cover to an institute B. The process is time-consuming, and it requires sufficient manpower to process transcripts. In addition, each transcript adds a significant financial overhead, and a student has to bear the same each time he/she applies for a higher study or job. Therefore, the conventional academic transcript system is not only insecure, but also it is costly, time-consuming, and error-prone.

In this section, we propose an academic transcript system based on blockchain technology. In the proposed system, we assume that an institute A has set up a private network comprising a set of nodes working as “miners.” The primary function of miner nodes is to validate the records of the transcript and/or the digital signature used to sign the transcript (using the corresponding public key). As the miner nodes are a part of a private network, they can easily identify the authorized person of an institute who signs the transcript, and hence, they can easily verify the public key of the authorized person. The above mechanism ensures the security of the system when the individual responsible to sign the documents left the system and kept the private-key for malicious use in the future. In order to provide the incentives to the miners, 80% of the total fee collected by an institute A (from students) is distributed among the miner nodes. As shown in Fig. 2, the proposed transcript system involves the following steps.

1. A student requests a transcript from an institute A where he/she is/was affiliated.
2. An institute A generates and verifies the transcript of the student, and digitally signs it using the private-key (e.g., 1024-bit or 2048-bit key). The algorithm used to generate the public/private-key pair and to sign the transcripts is ECDSA [2]. An institute adds the digitally signed transcript to the block.
3. An institute A broadcasts the block (which contains the transcript) to all the miner nodes available in the (private) network.

Fig. 2 Proposed academic transcript system



4. The miner nodes validate the signature and/or records (e.g., the individual that possesses the records and working as a miner node).
5. If at least 51% of total miner nodes validate the block, then the block will be added to the blockchain.
6. An institute A provides a unique 256-bit hash-code to a student (who requested the transcript). The hash-code is generated using the SHA-256 hash algorithm. The 256-bit hash-code is required to access/retrieve the transcript stored in the blockchain. Hence, the transcript available in the public blockchain can only be accessed by those institutes/organizations with whom the student has shared the unique 256-bit hash-code. The above mechanism ensures the privacy of a student. Although the transcripts are available in the public blockchain (unencrypted), they remain inaccessible (confidential) without the 256-bit hash-code.

4 Results and Discussions

We use the Truffle framework [4]—a development environment to deploy and test the smart contract of the proposed academic transcript system that executes on Ethereum blockchain. We use the Ganache [4]—a private Ethereum blockchain to deploy and test the feasibility of the smart contract of an academic transcript system free of charge. In order to write the smart contract, we use Solidity programming language [11]. We use the MetaMask as our wallet software to enable payments in the Ethereum blockchain network. The front end of the proposed academic transcript system is developed using the ReactJS and Web3.

In the conventional academic transcript system, in India, a student needs to pay approximately 3 US dollars to the institute as a processing fee of the transcript. In addition, in order to send the transcript (e.g., by post) to the institute/organization where the student applies for a higher study or job, the student requires to spend approximately 7 US dollars to 150 US dollars (depending on the location of the institute/organization).

In the proposed academic transcript system based on the Ethereum blockchain, the total cost to store a transcript on blockchain is 0.01 ETH (approx. 2.45 US dollars). Hence, in order to recover the processing charges of the transcript and incentives of the miner nodes, an institute may charge students approximately 0.02 ETH (approx. 6.19 US dollars). The reduced transaction cost (i.e., processing and communication charges of the transcript) as compared to the conventional academic transcript system exhibits the viability of the proposed academic transcript system for real-world application scenarios.

The proposed academic transcript system has the following advantages:

1. Constant overhead: A transcript of a student is stored in the immutable distributed transaction ledger (or blockchain), and hence, the process of generating and verifying a transcript only adds constant overhead (in terms of man-hours) for an institute A.
2. Delay: Transcripts are stored in the distributed transaction ledger, and they can be accessed/retrieved using distributed application (DApp). Hence, transcripts are retrieved/accessed by an institute B without any communication delay.
3. Transaction cost (or financial cost): Transcripts are stored in the immutable blockchain permanently. Hence, a student has to pay the transaction/transcript charges only once. In addition, a student (or an institute A) does not have to bear any charges to communicate/send the transcript to an institute B.
4. Security: Institute A generates and verifies the transcript. After verifying the transcript, it will be signed by an elliptic curve-based public key cryptosystem, namely ECDSA—a digital signature algorithm [2], and then, it will be stored in the blockchain. Hence, if an adversary wants to manipulate the contents of the transcript, he/she must possess the private-key that has been used by an institute A to digitally sign the transcript. In addition, due to the use of cryptographic hash function KECCAK-256 [1], the transcripts stored in the blockchain are tamper-resistant. If an adversary manipulates the contents of any transcript stored in the blockchain, it will be easily detected by the miners responsible for appending the block to the blockchain. A unique 256-bit hash-code required to access/retrieve the transcript ensures the secrecy of records stored in the blockchain and protects the privacy of a student.

5 Conclusions

The proposed academic transcript system uses blockchain technology to facilitate institutes and organizations to store and retrieve the transcripts of students securely and efficiently. In the proposed system, once the institute (where the student is/was affiliated) stores a transcript in the blockchain, it can be easily retrieved by other institutes without any delay. In addition, the institute has to process the transcript only once (before storing it in the blockchain). Hence, the overall cost of processing transcripts of a student, every time he/she applies to other institutes and organizations, reduces significantly. Security of the proposed academic transcript system depends on the security of underlying cryptographic algorithms, namely KECCAK-256—a cryptographic hash function and ECDSA—a digital signature algorithm. We used the Truffle framework and Ganache to execute the smart contract on Ethereum blockchain. The result (i.e., the prototype of the proposed system) validates the feasibility of the proposed blockchain-based academic transcript system for real-world application scenarios.

References

1. Bertoni G, Joan Daemen MP, Assche, GV (2011, January) The KECCAK SHA-3 submission <https://keccak.team/index.html>. Submission to NIST (Round 3)
2. Johnson D, Menezes A, Vanstone S (2001) The elliptic curve digital signature algorithm (ECDSA). *Int J Inf Secur* 1(1):36–63. <https://doi.org/10.1007/s102070100002>
3. Wood G et al (2014) Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper 151:1–32
4. Truffle Suite (2018) <https://www.trufflesuite.com>
5. Nakamoto S, Bitcoin A (2008) A peer-to-peer electronic cash system <https://bitcoin.org/bitcoin.pdf>
6. Yue X, Wang H, Jin D, Li M, Jiang W (2016) Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *J Med Syst* 40(10):1–8. <https://doi.org/10.1007/s10916-016-0574-6>
7. Huh S, Cho S, Kim S (2017) Managing IoT devices using blockchain platform. In: The 19th international conference on advanced communication technology (ICACT), IEEE, pp 464–467. <https://doi.org/10.23919/ICACT.2017.7890132>
8. Kshetri N (2018) Blockchain's roles in meeting key supply chain management objectives. *Int J Inf Manage* 39:80–89
9. Szabo N (1997) Formalizing and securing relationships on public networks. *First Monday* 2(9). <https://doi.org/10.5210/fm.v2i9.548>
10. Buterin V (2014) A next-generation smart contract and decentralized application platform. White paper
11. Dannen C (2017) Solidity programming., Apress, Berkeley, USA, pp 69–88

Chapter 34

A Brief Survey of Sentiment Analysis



Ashwini Save and Narendra Shekokar

1 Introduction

A huge amount of data is being created on a daily basis. Making sense of this data is an important task, which is where data mining comes into the picture. Data mining is an interdisciplinary subfield of computer science. Data mining helps in identifying and extracting information and patterns from large data sets.

Opinion mining or sentiment analysis or opinion extraction or review mining is an important specialisation of data mining. It makes use of techniques analysis of texts, natural language processing, and computational linguistics to make sense of the subjective information from a given material, which may be anything from products to services, etc.

Sentiment analysis is extremely useful as it allows us to gain an overview of the wider public opinion behind certain topics. A huge amount of data is available, from which critical information and patterns can be retrieved. This retrieval of information can be a challenging task, which makes it that much more fascinating. Also, the commercial use of this information makes it an important task to be undertaken.

In the Indian perspective, India has a very large Internet user base at over 400 million internet users. With the exponential rise in the Internet user base, there has been a rise in the number of users who are using specific websites and social platforms to express their opinions and also to get first-hand reviews on any product or topics. The popularity of gaining knowledge from the expressed opinions can be gauged by the fact that a particular website (www.mouthshut.com) [1] claims that more than

A. Save (✉) · N. Shekokar

Department of Computer Science & Engineering, D. J. Sanghvi College of Engineering, Mumbai,
India

e-mail: ashwini.save@gmail.com

N. Shekokar

e-mail: narendra.shekokar@djsce.ac.in

lakhs of users have been influenced by the opinions expressed on this particular website.

By 2013, in the electronics section alone only on amazon.com, there were more than 1.2 million reviews [2]. Also, according to Internet and Mobile Association of India, it was believed that more than 66% of Internet users claimed that they were influenced by the online movie reviews [3].

This proves that more and more people are turning to the Internet in order to share their opinions and read reviews from customers to make knowledgeable decisions. This is why sentiment analysis is important. And, this paper gives a brief introduction to the different approaches to sentiment analysis and current research being carried in these different approaches.

2 Sentiment Analysis

Many techniques have been proposed in the field of sentiment analysis, while some of the related works have been discussed in this section.

Farra et al. [4] conduct sentiment analysis for Arabic texts. This is done at both sentence level and document level. For the sentiment analysis at the sentence level, two approaches have been investigated. For document-level classification, they propose to use sentences to classify whole documents, which are of known classes. The major advantages of this paper are that it provides a novel idea of document-level classification.

Tungthamthiti et al. [5] propose a new method to identify sarcasm in tweets (twitter message). The paper focuses on several approaches including concept-level sentiment analysis and use of common-sense knowledge, coherence and classification through machine learning.

The biggest advantage of this paper is that it tries to tackle sarcasm which is one of the most difficult tasks in natural language processing with a respectable accuracy of 80%.

Sharma et al. [6] propose a simple system for sentiment analysis of movie review. With the help of a list containing sentiment scores for opinion words, the opinion sentiment score is obtained. And finally, the scores are aggregated to give a sentiment score at the document level.

Khan and Baharudin [7] propose a rule-based, domain-independent sentiment analysis method. The paper proposes to use the popular sentiment dictionary Senti-WordNet to calculate their polarity of all the words under. The biggest contribution of this paper is the creation and use of knowledge base for domain-independent sentiment classification.

In the Indian context, it is very common to use Hindi words in an English script, known as Hinglish words. General classifiers or feature extractors do not take these words into consideration while performing sentiment analysis. Seshadri et al. [8] provide a way through which the classifiers, in this case Naïve Bayes classifier, can be effectively modified with the addition of a corpus containing Hinglish words.

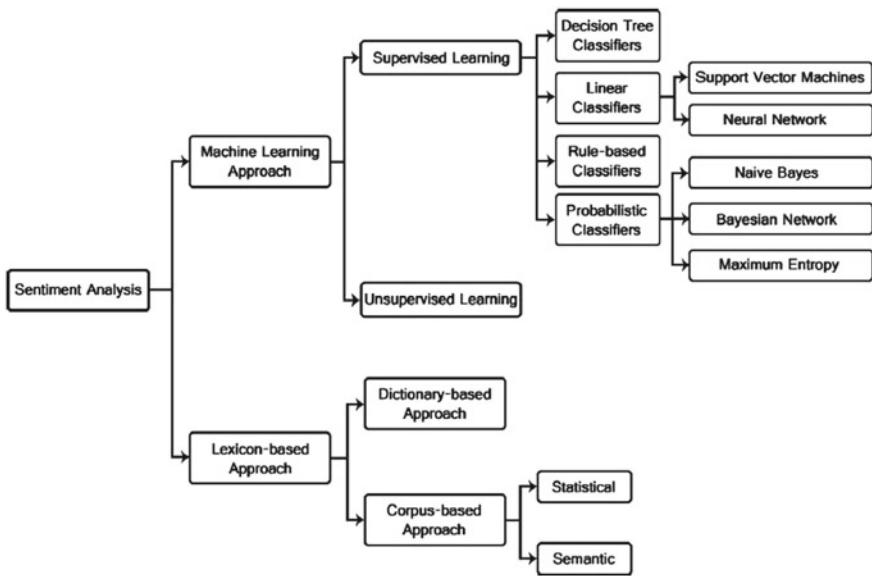


Fig. 1 Classification of techniques for sentiment analysis [10]

When it comes to complex linguistic structures and understanding the context of sentences, most existing machine learning approaches fail in this area. To counter this, Yang and Yang [9] propose to build a system which is context-aware, and hence, sentiment at individual sentences can be understood. The system makes use of the context-aware constraints which are helpful when the amount of labelled data is limited. Over the years, many techniques have been proposed for sentiment analysis. These include supervised and unsupervised learning approaches and dictionary-based or corpora-based approaches. Fig. 1 gives the taxonomy of basic techniques for sentiment analysis.

The techniques for sentiment analysis can be broadly classified into two types: machine learning-based and lexicon-based approaches.

2.1 Lexicon-Based Approach

Lexicon is a vocabulary of a language. Generally, it is considered that languages have two parts: a lexicon, which is essentially a complete catalogue of language's words, and grammar, which is a set of rules for using these words to form a meaningful sentence.

The rapid growth of technology is helping entrepreneurs to launch new products almost on a daily basis. And in this competitive field, knowing and having information about the rival product is a necessity. To make the job simpler, Kuppili et al. [11]

propose a system called as variance-based product recommendation (VPR) approach. The main aim of this approach is to find the top competitors of your newly launched product. This is done by checking the similarity in the description.

On the other hand, Bhoir and Kolte [12] found out that while giving reviews on any movies, especially Hindi movies, people used language which was generally not found in a usual lexicon dictionary. So, they proposed to develop a new lexicon dictionary with sentiment score. This own dictionary with sentiment score was very useful for getting a better sentiment score. The lexicon-based sentiment analysis again can be divided into two types: dictionary-based and corpus-based approaches.

2.1.1 Dictionary-Based Approach

In dictionary-based sentiment, a dictionary with pre-defined sentiment score is created and this dictionary is used to find the overall sentiment score. There are many dictionaries with sentiment scores for many languages; for example, SentiWordNet is a very popular Sentiment dictionary for English.

Jose and Chooralil [13] proposed a system which uses these standard sentiment dictionaries, WordNet and SentiWordNet, for the prediction of election results by using real-time tweets.

But, Kawabe et al. [14] found out that in many cases, these dictionaries could not be used. In the aftermath of tsunami, it was found out that many people were using the microblogging site for rumour-mongering and using the standard sentiment dictionary was impossible to ascertain the credibility of the tweets. So, the paper proposes a new method for sentiments of words and phrases. Also, the paper proposes substantial changes to the sentiment dictionary.

2.1.2 Corpus-Based Approach

In the corpus-based lexicon sentiment analysis, instead of a sentiment dictionary, a huge collection of sentences with a sentiment score is maintained for a particular language. The sentiment score calculated for a particular word is more dynamic in corpus-based approach. Because of this dynamism, corpus-based approach is better compared to the dictionary-based approach. But, the corpus used should be sufficiently large, else sentiment score might not be so accurate.

Abdulla et al. [15] demonstrate the development of a corpus for the Arabic language. To demonstrate the usefulness of building a corpus for sentiment analysis, they created a corpus containing 1000 positive tweets and 1000 negative tweets.

The paper demonstrates the usefulness of developing or using a corpus for a language over directly using a sentiment dictionary for the analysis of sentiments.

Riloff and Shepherd [16] have taken the idea of semantics forward. But, instead of using a corpus of a language and creating a semantic relationship for sentiment analysis, they proposed to develop a complete corpus using semantics as a base.

Motivated by this observation, Carvalho et al. [17] propose to use paradigm words to classify tweets. Here, using genetic algorithm, the paradigm words were selected and subset of such words are found. Because of which, the paper claims to improve the classification accuracy.

In corpus-based approaches, the requirement of corpus is critical and also the amount of data required is also large. But the major drawback of corpus-based approach is that no model can be trained to perform sentiment analysis on unknown data.

2.2 Machine Learning-Based Approach

The second broader classification of sentiment analysis technique is the machine learning-based approach. Machine learning deals with algorithms that allow computers to learn. It is generally seen that all data contain some kind of patterns and insights hidden inside which comes to the fore when observed keenly. This is possible because the algorithm can make a generalised rule from what it has seen in the data it was given [18].

The training data is the most important part of machine learning algorithm. Without the proper selection of training and testing data, machine learning approach will fail miserably. So, for sentiment analysis of tweeter data, Suchdev et al. [19] propose to take a minimum of 5600 tweets as training data for a specific domain.

The machine learning-based approach can be further classified into two types: supervised learning and unsupervised learning.

2.2.1 Unsupervised Learning

The problem of unsupervised learning is that of trying to find hidden structure in unlabelled data. Since the examples given to the learner are unlabelled, there is no error or reward signal to evaluate a potential solution. Because of the absence of any labelled data, the unsupervised learning approach makes use of clustering and association techniques to find the relation of the unknown data and the unlabelled training data.

Generally in unsupervised where there no labelled data clustering and association, rules can be applied with the help of bag of words technique and also called as Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF).

Also, word2vec is a system which learns to correlate words with other words in an unlabelled environment of unsupervised learning. Doc2vec, on the other hand, learns to correlate labels, and Sanguansat [20] propose an extension to this technique by adding a distributed memory to speed up the system.

Khanaferov et al. [21] are of the opinion that on the Internet, especially on the microblogging sites like twitter it is very difficult to carry out sentiment analysis

as these opinions on social platforms are very difficult to classify using supervised learning. So, clustering was considered to be the best option.

2.2.2 Supervised Learning

In supervised learning, a function is inferred from labelled training data. In supervised learning, the algorithm is given both the input and a desired output and it produces an inferred function. This function is used for mapping of new examples. This is done by the learning algorithm by generalising from the training data to unseen situations. Because of the presence of labelled data, the supervised learning makes use of classification and regression techniques.

Bouazizi and Ohtsuki [22] found that even though there had been a surge in different automated system for sentiment analysis of tweets, these state-of-the-art proposed approaches were mostly focusing on the binary and ternary sentiment classifications. They were of the opinion that it would be very interesting if a deeper classification was carried. So, they proposed to classify a text into seven classes; these classes are “happiness”, “sadness”, “anger”, “love”, “hate”, “sarcasm” and “neutral”.

Classifiers are generally used in the supervised learning approach. These classifiers can be categorised into four types.

i. *Decision Tree Classifier*

A decision tree is a decision support tool which uses a tree-like graph to map decisions to its supposed possible consequences.

Zhamagambetov and Pak [23] combine the idea of unsupervised learning and supervised learning in the proposed paper. The paper proposes to use the technique of using Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) and implement it with decision tree to get a better and accurate sentiment analysis score.

ii. *Linear Classifier*

The goal of any classifier is to try and understand the characteristics of an object and use this information to identify its class or group. In linear classification, the classifier performs some kind of linear operations in order to perform classification. The linear classification is achieved by two techniques: neural networks and support vector machines.

Duncan and Zhang [24] demonstrate the use of artificial neural networks in the field of sentiment analysis. The proposed system uses the feedforward neural network system.

Ouyang et al. [25] proposed a convolutional neural network model for sentiment analysis. A convolutional neural network is a variant of feedforward artificial neural network. This artificial neural network is inspired by the neurological structure of the animal visual cortex. That is, by tilting of the visual field helps the individual neurons to respond to the overlapping regions.

For the sentiment analysis, the proposed system uses word2vec by Google to compute vector representations of words. This vector representation is an input to the proposed neural network.

The second technique for linear classification is support vector machine (SVM). SVM analyses the data and works in a non-probabilistic manner. Generally, the SVM is used for binary classification. SVM training algorithm builds a model that assigns new unknown or unseen examples into one category or the other.

Devi et al. [26] demonstrate the effectiveness of SVM in sentiment analysis. The proposed system first performs sentence-level classification. This is followed by POS tagging, subjectivity/objectivity classification and extraction of aspects.

On the other hand, Zhao et al. [27] proposed a technique of combining of semantic and prior polarity with SVM. The proposed system incorporates semantic feature, prior polarity score feature and n -grams feature as sentiment feature set into support vector machines (SVM) model training and perform sentiment classification.

iii. Rule-Based Classifier

In rule-based classifier system, the administrator has to give an extensive and exhaustive set of rules to classify the input data.

Im Tan et al. [28] proposed a rule-based system for sentiment analysis of financial news. The system works at assigning polarity at the sentence level. The system proposes seven sets of rules of phrases.

The rules have been given for noun phrase sentiment composition, verb phrase sentiment composition, verb–noun/noun–verb phrase sentiment composition rules, the preposition phrase sentiment composition rules and rules for conjunction.

iv. Probabilistic-Based Classifier

A probabilistic-based classifier predicts the probability distribution of classes for the given sample, rather than just giving the likely class that the given sample might belong to.

There are three techniques which use probabilistic approach for classification. These techniques are Naïve Bayes classifier, Bayesian network and maximum entropy classifier.

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. It is one of the most commonly used algorithm as it is not very complex, even when the data set is large, to understand and implement as it does not contain iterative parameters.

On the other hand, Bayesian network is a probabilistic graphical model that represents conditional dependencies for a set of random variables via a directed acyclic graph (DAG), whereas when the data is the most random maximum entropy is achieved.

Wikarsa and Thahir [29] effectively demonstrate the usefulness of Naïve Bayes classifier on twitter data set, with a claimed accuracy of around 83%.

On the other hand, Yan and Huang [30] showcases the effective use of maximum entropy model for Tibetan language sentiment analysis. The maximum entropy model was implemented and tested for around ten thousand Tibetan sentences.

There are other papers which have made use of multiple classifiers together or have tested new system on multiple classifiers.

Moh et al. [31] are of the opinion that instead of using a single-tire prediction system, it would be much more beneficial to have a multi-tire prediction system. The predictive model is divided into three tires. It has been claimed that this three-tire system improves the accuracy of the classifier as the classification task complexity reduces.

Jain and Katkar [32] carry out a comprehensive comparison of different classifiers to determine which classifier gives the best result for an experimental twitter data set. The proposed experiment used the combination of different classifiers to ascertain the best possible result. The experiment shows that the ensemble of classifier approach, contrary to common perception, was outperformed by single classifiers.

Yang and Zhou [33] pitches Naïve Bayes classifier and SVM classifier against each other. After the comparison, it has been claimed that the Naïve Bayes classifier is very slow compared to SVM classifier, that the processing time required by Naïve Bayes classifier is more compared to SVM classifier for the same data set. And, for the taken data set the accuracy of both classifiers was nearly equal.

Yuan et al. [34] proposed a hybrid method for multi-class classification for sentiment analysis of microblogging site. The proposed system combines the model-based approach with the lexicon-based approach.

It has been seen that extensive research has been carried out in the field of sentiment analysis. Many scholars have also acknowledged that proper preprocessing of the extracted data is a necessary step or a prerequisite to get an accurate sentiment score.

3 Analysis

In this section, the analysis of different sentiment analysis approaches and algorithms has been given.

3.1 *Analysis of Sentiment Analysis*

Sentiment analysis can be categorised into lexicon- and machine learning-based approaches. Each of these two approaches can be further categorised into two types. Table 1 gives the analysis of these four sentiment analysis approaches.

As the unsupervised learning accepts non-annotated training set, it uses clustering algorithms such as K-means clustering and K-nearest neighbour. Considering all the above sentiment analysis approaches, it can be seen that the supervised learning approach comparatively gives one of the highest accuracies and also it uses annotated training set which helps in using classification algorithms.

Table 1 Sentiment analysis classification

	Dictionary-based approach	Corpus-based approach	Unsupervised learning approach	Supervised learning approach
Type of learning	No learning	No learning	Unsupervised	Supervised
Training set or corpus required?	No	Corpus	Non-annotated training set	Annotated training set
Preprocessing requirement	Yes	Yes	Yes	Yes
Technique Used	Using dictionary	Using corpus	K-means clustering	Using classification algorithms
			K-nearest neighbour	
–	Less complex	Relatively complex	Complex	Complex

3.2 Analysis of Supervised Learning Approaches

There are two broad approaches for performing sentiment analysis, lexicon- and machine learning-based approaches. And further these two again have two sub techniques each. Dictionary- and corpus-based approaches are for lexicon and supervised and unsupervised for machine learning-based approaches. Table 2 performs basic analysis of these four approaches based on the type of learning, complexity of the approach, etc.

Table 2 Supervised learning approaches

Parameter\models	Decision tree classifiers	Linear classifiers	Rule-based classifiers	Probabilistic-based classifiers
Type of learning	Supervised	Supervised	Supervised	Supervised
Training set or corpus required?	Training set	Training set	Training set	Training set
Preprocessing requirement	Yes	Yes	Yes	Yes
Technique used	Decision tree	Support vector machines or neural network	Rules	Naïve Bayes, bayesian network or maximum entropy
Complexity	Complex	Support vector machines are complex and neural	Complexity depends on the exhaustiveness	Naïve Bayes is less complex compared to maximum entropy

Table 3 Supervised learning algorithms

Parameter\models	Support vector machines	Neural network	Maximum entropy	Naïve Bayes
Deployment	Complex	Extremely complex	Complex	Least complex
Dealing with multiclass problem	Cannot	Cannot	To some extent	Yes
Data required for training	Average	Very large	Average	Low to average
Training time	Average	High	Average	Low

Of the four approaches in supervised learning, linear classifiers and probabilistic-based approach are the most efficient as they produce highest accuracy. support vector machine and neural network are two algorithms which under linear classifiers, and Naïve Bayes and maximum entropy are the important algorithms which use probabilistic approach.

3.3 Analysis of Supervised Learning Algorithms

In Table 3, some basic supervised learning algorithms have been analysed with respect to the deployment complexity, the amount of data required for training purpose, etc.

Of the four supervised algorithms discussed, the support vector machine and Naïve Bayes produce the highest accuracy and their deployment is not very complex. One important thing that needs to be noticed is that the support vector machine finds it difficult to deal with multi-class problems.

4 Conclusion

The adaptation of sentiment analysis and opinion mining in different domains has raised many issues. Over a period of time, many scholars have proposed many different techniques in order to solve the issues. Also, many scholars have adapted different learning approaches and learning algorithms and have further customised the algorithms to cater to the need of solving the issues that arise in different domain.

The paper conducts a review of different sentiment analysis algorithms and different approaches proposed in different papers. The review highlights that though there are different machine learning approaches which perform sentiment analysis supervised learning approaches prove to better. Further in supervised learning even though neural networks are extremely complex to implement, it has been seen that it

performs much better compared to others when the network configuration has been mastered. Also, a simple artificial, neural network can be deepened to form a deep neural network or a deep learning network which gives better accuracy and is more efficient.

Moreover with the expansion in the use of opinion mining in different fields different issues like cross-domain sentiment analysis or cross-domain adaptation, preprocessing of the data, etc., have cropped up which require in-depth and detailed investigation before using the standard machine learning approaches and algorithms.

References

1. Editorial Staff (2016, June 8) Available: <http://www.mouthshut.com/>
2. Woolf M (2014, July 4) A statistical analysis of 1.2 million amazon reviews [Online]. Available: <http://minimaxir.com/2014/06/reviewing-reviews/>. Last Accessed 4th July 2017
3. Editors (2017, July 7) <https://en.wikipedia.org/wiki/MouthShut.com>. Last accessed 4th July 2017
4. Farra N, Challita E, Assi RA, Hajj H (2010) Sentence-level and document-level sentiment mining for Arabic texts. In: 2010 IEEE international conference on data mining workshops. pp 1114–1119
5. Tungthamthiti P, Shirai K, Mohd M (2014) Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches. In: 28th Pacific asia conference on language, 2014, information and computation. pp 404–413
6. Sharma R, Nigam S, Jain R (2014) Opinion mining of movie reviews at document level. Int J Inf Theor (IJIT) 3(3):13–21
7. Khan A, Baharudin B (2011, August) Sentiment classification by sentence level semantic orientation using SentiWordNet from online reviews and blogs. Int J Comput Sci Emerg Technol 2(4):539–552
8. Seshadri S, Lohidasan A, Lokhande R, Save A, Nagarhalli TP (2015, August) A new technique for opinion mining of hinglish words. Int J Innovative Res Sci Eng Technol 4(8):7184–7189
9. Yang B Yang B (2014) Context-aware learning for sentence-level sentiment analysis with posterior regularization. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, June 23–25 2014. Baltimore, Maryland, USA, pp 325–335
10. Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J 5:1093–1113
11. Kupilli V, Kumar D, Kudchadker GP, Arora A (2015) Variance based product recommendation using clustering and sentiment analysis. In: IEEE workshop on computational intelligence: theories, applications and future directions (WCI)
12. Bhoir P, Kolte S (2015) Sentiment analysis of movie reviews using Lexicon approach. In: IEEE international conference on computational intelligence and computing research
13. Jose R, Chooralil VS (2015) Prediction of election result by enhanced sentiment analysis on twitter data using word sense disambiguation. In: International conference on control, communication & computing india (ICCC), IEEE, 19–21 November 2015. Trivandrum, pp 638–641
14. Kawabe T, Namihira Y, Suzuki K, Nara M, Sakurai Y, Tsuruta S, Knauf R (2015) Tweet credibility analysis evaluation by improving sentiment dictionary. In: IEEE congress on evolutionary computation (CEC). pp 2354–2361
15. Abdulla NA, Ahmed NA, Shehab MA, Al-Ayyoub M (2013) Arabic sentiment analysis: Lexicon-based and corpus-based. In: IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)

16. Riloff E, Shepherd J A corpus-based approach for building semantic Lexicons, <http://www.aclweb.org/anthology/W97-0313>
17. Carvalho J, Prado A, Plastino A (2014) A statistical and evolutionary approach to sentiment analysis. In: IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT). pp 110–117
18. Schrauwen S (2010) Machine learning approaches to sentiment analysis using the Dutch Netlog Corpus. CLiPS Research Center, University of Antwerp, Belgium
19. Suchdev R, Kotkar P, Ravindran R, Swamy S (2014, October) Twitter sentiment analysis using machine learning and knowledge-based approach. *Int J Comput Appl* 103(4):34–40
20. Sanguansat P (2016) Paragraph2Vec-based sentiment analysis on social media for business in Thailand. In: IEEE 8th international conference on knowledge and smart technology (KST). pp 175–178
21. Khanferov D, Luc C, Wang (George) T (2014) Social network data mining using natural language processing and density based clustering. In: IEEE international conference on semantic computing, 2014. pp 250–251
22. Bouazizi M, Ohtsuki T (2016) Sentiment analysis: from binary to multi-class classification, a pattern-based approach for multi-class sentiment analysis in Twitter. In: IEEE ICC SAC Social Networking
23. Zharmagambetov AS, Pak AA (2015) Sentiment analysis of a document using deep learning approach and decision trees. In: IEEE twelve international conference on electronics computer and computation (ICECCO)
24. Duncan B, Zhang Y (2015) Neural networks for sentiment analysis on Twitter. In: Proceedings of IEEE 14th international conference on cognitive informatics & cognitive computing ICCICC–15. pp. 275–278
25. Ouyang X, Zhou P, Li CH, Liu L (2015) Sentiment analysis using convolutional neural network. In: IEEE international conference on computer and information technology; ubiquitous computing and communications; Dependable, autonomic and secure computing; pervasive intelligence and computing, 2015. pp 2359–2364
26. Devi DN, Kumar CK, Prasad S (2016) A feature based approach for sentiment analysis by using support vector machine. In: IEEE 6th international conference on advanced computing, 2016 pp 3–8
27. Jianqiang Z, Xueliang C (2015) Combining semantic and prior polarity for boosting twitter sentiment analysis. In: IEEE international conference on smart city/SocialCom/SustainCom together with DataCom 2015 and SC2 2015. pp 832–837
28. Im Tan L, San Phang W, Chin KO, Anthony P (2015) Rule-based sentiment analysis for financial news. In: IEEE international conference on systems, man, and cybernetics 2015. pp 1601–1606
29. Wikarsa L, Thahir SN (2015) A text mining application of emotion classifications of Twitter's users using Naïve Bayes method. IN: IEEE 1st international conference on wireless and telematics (ICWT)
30. Yan X, Huang T (2015) Tibetan sentence sentiment analysis based on the maximum entropy model. In: IEEE 10th international conference on broadband and wireless computing, communication and applications, 2015. pp 594–597
31. Moh M, Gajjala A, Gangireddy SCR, Moh TS (2015) On multi-tier sentiment analysis using supervised machine learning. In: IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, 2015. pp 341–344
32. Jain AP, Katkar VD (2015) Sentiments analysis of twitter data using data mining. In: IEEE international conference on information processing (ICIP), Dec 16–19, 2015. Vishwakarma Institute of Technology, pp 807–810
33. Yang Y, Zhou F (2015) Microblog sentiment analysis algorithm research and implementation based on classification. In: IEEE 14th international symposium on distributed computing and applications for business engineering and science, 2015. pp 288–291
34. Yuan S, Wu J, Wang L, Wang Q (2016) A hybrid method for multi-class sentiment analysis of micro-blogs. IN: IEEE 13th international conference on service systems and service management (ICSSSM)

Chapter 35

Comparison of Traditional Machine Learning and Deep Learning Approaches for Sentiment Analysis



Dhvani Kansara and Vinaya Sawant

1 Introduction

Natural language processing (NLP) is the field lying at the intersection of computer science, artificial intelligence and linguistics. Sentiment Analysis, which is a sub-domain of NLP, is a process of determining the emotional tone of a comment based on words contained in it, in order to understand the attitude and opinion behind it. This process is also called opinion mining. It has multiple applications mainly in the form of social media monitoring. In recent years, there have been great advancements in this field. Various machine learning algorithms have shown to prove effective in categorizing the sentiment from a text. Recent advances have been focusing on using deep learning algorithms for this purpose. This proliferation is due to the fact that opinions are central to almost all human activities and are key influencers of our behaviors. Our beliefs and perceptions of reality, and choices we make, are, to a considerable degree, conditioned upon how others see and evaluate the world [1].

In this paper, we discuss two paradigms: traditional approaches for classification which have been in use since the past few decades and the recent breakthroughs leveraging deep learning algorithms. The models work on raw data further cleaned and pre-processed, considering removal of stop words, punctuations and mark-ups with stemming (process of reducing inflected words to their word stem) and then checking the overall sentiment by assigning polarity based on resultant cleaned sentence. The models also take into account inversion terms or negations (such as—"not bad") which reverse the polarity of the sentence as a whole. Later, this data is fed into a learning-based model that uses a supervised learning algorithm. In case of

D. Kansara (✉) · V. Sawant (✉)

Dwarkadas J. Sanghvi College of Engineering, Mumbai University, Mumbai, India
e-mail: dhvani.djk@gmail.com

V. Sawant
e-mail: vinaya.sawant@djsce.ac.in

traditional models, this text is pre-processed using the bag of words methodology. It gives a combination of types of semantic features that attempt to model the syntactic structure of sentences, intensification, negation, subjectivity or irony [2]. In this paper, we use Naive Bayes, Logistic Regression and Random Forest as traditional classification approaches. On the other hand, deep learning models learn from multiple layers of representations or features of data and produces results accordingly. For this, we use vector representations of sentiments as inputs to the deep layers. We use modified Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN) and a combination of CNN and LSTM to obtain experimental results. We then analyze the accuracies obtained in all these algorithms.

We present each paradigm along with the description of pre-processing phases and an overview of the algorithms used. We also try to analyze reason behind difference between the accuracies. We use three different datasets for this purpose and classify movie reviews, hotel reviews and tweets, thereby concluding that deep learning models prove to have an upper edge in all the cases.

2 Methodology

2.1 *The Traditional Approach*

Extensive research has been carried out in the past few years to apply basic machine learning algorithms for sentiment analysis. We leverage this research to ensure that our approach will provide the best results. In order to obtain a good performance accuracy, the pre-processing phase is carried out prior to the classification process.

Pre-processing. The reviews may be taken directly from a Web site, so it may include html mark-up. Thus, we first use the BeautifulSoup package to strip all html mark-up. We then strip the stop words using the Porter Stemming package in Natural Language Toolkit (NLTK) to stem and remove stop words like “a,” “and,” “is,” “the” because these are frequently occurring words which do not carry any significant meaning. Then, we convert this categorical data of cleaned reviews and tweets into numeric data so that we can apply machine learning algorithms on them. We use an approach called bag of words [3] for this. This model creates a dictionary by taking all words in the dataset and then assigns integers according to the count of each word of the dictionary appearing in the given text. For example, consider the following two sentences:

Sentence 1: “The movie had amazing actors”

Sentence 2: “The view from the hotel room was amazing”

So, our vocabulary will be as follows:

{the, movie, had, amazing, actors, view, from, hotel, room, was}.

To get our bag of words, we count the number of times the word from the vocabulary list occurs in each of our sentences.

In Sentence 1, “the,” “movie,” “had,” “amazing,” “actors” each appears once, so the feature vector for Sentence 1 is: {1, 1, 1, 1, 1, 0, 0, 0, 0, 0}.

Similarly, the feature vector for Sentence 2 is: {2, 0, 0, 1, 0, 1, 1, 1, 1} In this way, we obtain a sequence of integers for each review and tweet.

In the datasets, we have a very large number of reviews, which will give us a large vocabulary. To limit the size of the feature vectors, we chose a maximum vocabulary size of 5000 words, i.e., we consider the top 5000 most frequently occurring words in our vocabulary. (Considering we already removed stop words, this is enough.) Thus, we obtain a list of size 5000 with integer values mapped according to the vocabulary for each row.

The Algorithms Part I. The dataset is divided into 80% tuples as training data and remaining 20% tuples as test data and the algorithms are applied on it. Accuracy is calculated based on how correctly the model predicts the test data. Three traditional classification algorithms were experimented,

- i. Naive Bayes
- ii. Logistic Regression
- iii. Random Forests.

Naïve Bayes Algorithm. It is based on the formula that, for a label y , we have the independent data feature x_i as in Eq. (1).

$$P(y|x_1, \dots x_n) \propto P(y) \prod_{i=1}^n P(x_i|y) \quad (1)$$

where $P(y|x_1, \dots x_n)$ is the posterior probability stating that x belong to class y . To create the classifier model, the probability of given set of inputs for all possible values of the class variable y is found, the output with maximum probability is the final classification result. This can be expressed mathematically as in Eq. (2).

$$y = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i|y) \quad (2)$$

This algorithm assumes the property of conditional independence among all attributes, which may not be true in all cases thus reducing the accuracy to a certain extent [4].

Logistic Regression. It is one of the most commonly used binary classification algorithm in machine learning which gives discrete binary output between 0 and 1 [16]. It uses a logistic function, also called as sigmoid function whose equation is as shown in Eq. (3)

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Taking into consideration the weight of each input variable the output is predicted, this weighted input is passed to the sigmoid function to obtain a probability value between 0 and 1. For any real number input, the sigmoid function maps it to a value between 0 and 1. These values will then be transformed into either 0 or 1 using a threshold classifier.

Since it is a discriminative classifier which does not rely on the assumption of attribute independency as in case of Naive Bayes, thus this model may generally have a better accuracy comparatively.

Random Forest Classifier: It is formed by a collection of multiple decision tree classifiers collectively called as a “forest.” Decision tree classifier concept is based on the rule-based system [17]. Individual decision trees are generated using selection of attribute at each node to determine the split. This node is selected in a strategic manner such that the most efficient tree is formed. The flow of the tree to classify a certain tuple will be in the form of an if-then rule. The same set rules can be used to perform the prediction on the test dataset. Multiple such trees are formed with different combinations of training and testing data. Then, it takes the test features and uses the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome. Votes for each predicted result are calculated. Result which has received the maximum votes is considered as the final prediction. In this, multiple trees (different rules) are generated from given input instances or attributes and majority of the votes of the classifier output is considered as the final class [5]. Overfitting is avoided in this algorithm because every time a random set of inputs are used to build the decision tree. We use 100 estimators, i.e., 100 different decision trees to train the model and “entropy” as splitting criteria for determining the best split point.

2.2 Deep Learning Approach

Deep learning architecture consists of input layers, hidden layers and output layers. Each layer is made up of neurons, which are basically weight matrices. The higher the weight, the more influence layer has over the next. The input is passed through the hidden layers to give a certain output. Backpropagation is used to minimize the loss incurred between predicted and actual outcome. It does so by changing the weight matrices in the dense layers such that error is minimized. In this way, the learning occurs.

Pre-processing. The cleaning of the datasets is done as in the previous case, i.e., by removing html mark-up, punctuations and stop words. For input to the dense layers, we form a vector in latent space for each review, and this approach is called Word-to-Vectors. We use tokenizer for dividing sentences into tokens (or words). Unique words are identified and unified in a list, from all inputs taken into consideration collectively. We then assign an integer value for each of the unique word in the list. So now we have a dictionary of words each assigned a specific integer number. Then,

ide ▲	Type	Size	Value
0	unicode	1	stuff go moment mj start listen music watch odd documentari watch wiz ...

Fig. 1 Cleaned review

ide ▲	Type	Size	Value
0	list	219	[458, 24, 166, 6988, 85, 878, 83, 11, 793, 499, ...]

Fig. 2 Token representation of review

each review is represented by a list of these integers which form a part of that review. The transformation is as shown in Figs. 1 and 2.

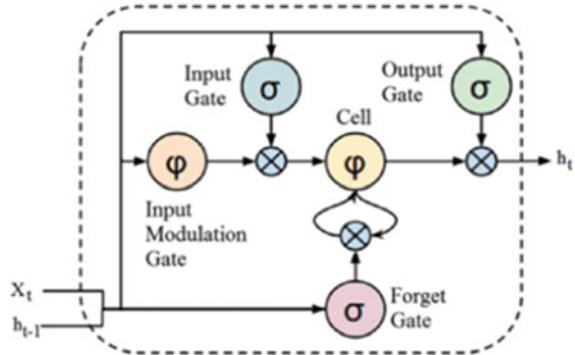
We pad the short reviews with 0s in the beginning and truncate the longer reviews. During training, the model will learn that the 0s carry no information thus preserving the content in each case and equalizing the length of each input, which is necessary for computation. We constrict a total size of 300 for each review.

The Algorithms Part II. The datasets are divided into 80% tuples as training data and remaining 20% tuples as test data and the algorithms are applied on it. Accuracy is calculated based on how correctly the model predicts the test data. The following algorithms were experimented:

- i. Recurrent Neural Network (Long Short-Term Memory)
- ii. Convolutional Neural Network
- iii. Convolutional Neural Network + Long Short Term Memory.

Recurrent Neural Network (Long Short-Term Memory). RNNs remember all the relations while training itself and are thus used particularly for sequential data [6]. The output of the previous state also goes into the input of the next state along with current input vector, this helps the model to remember context while training. That is why it is used for sentiment analysis and due to its nature it can be used to correctly predict sentiments of statements such as “Though initially I liked the movie, but toward the end it became dull and boring” this sentence has a negative sentiment but also positive words like “liked” which can confuse the system, but not in case of recurrent neural network, which can remember long-term dependencies.

In traditional recurrent neural network, during the backpropagation phase, if the values of weight matrix of the recurrent neural layers are very small (less than 1.0) that ultimately they do not have any effect on the input signal and thereby preventing the neural network from training further. The problem is called vanishing gradient problem because the gradient signal vanishes slowly when passing through different layers [7]. Thus, RNNs have problems learning long-range dependencies, i.e., relations between words that are several steps apart. Conversely, we could also face exploding gradient problem if the weight matrix values become too huge, causing training to diverge. For this reason, we use a modified version of recurrent neural network called Long Short-Term Memory (LSTM) Neural Network which is capable

Fig. 3 LSTM functioning

of learning long-term dependencies. We need to update information less chaotically and account for all the learning and the memory that it has learnt over a vast period of time, in order to make more accurate [15].

Labels in Fig. 3 and equations below represent:

X_t —current input vector | b —coefficients | h_{t-1} —Output of previous block | W —weights | h_t —output of current block | σ —sigmoid function | φ —tanh function.

Figure 3 [8] describes the functioning of a LSTM model. The cell state is the most important component which “remembers” values over arbitrary time periods thus giving LSTM a long-term memory; it represents all the learnings across time.

An LSTM cell consists of three gates as shown in Fig. 3:

1. A forget gate which is used to formulate an attention mechanism which helps filter out relevant data by knowing what to forget and what to remember. It takes current input(X_t) and previous output(h_{t-1}) and passes it through a sigmoid function and judges if the data is worth remembering, which outputs 1 if data is to be remembered or 0 if it is to be forgotten. Thus, its equation is given by Eq. (4).

$$f_t = (W_f[h_{t-1}, X_t] + b_f) \quad (4)$$

2. An input gate decides what new information we are going to store in the cell state by involving a sigmoid function to X_t and h_{t-1} . The equation of input gate is given by Eq. (5).

$$i_t = (W_i[h_{t-1}, X_t] + b_i) \quad (5)$$

The tanh layer (gives output between -1 and $+1$) is responsible for creation of vector of new candidate value, \tilde{C}_t as given by Eq. (6).

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, X_t] + b_C) \quad (6)$$

3. This new candidate value is multiplied with the input gate and forget gate output along with previous cell state, C_{t-1} is used to formulate the new cell state C_t as given in Eq. (7).

$$C_t = (f_t * C_{t-1}) + (i_t * \tilde{C}_t) \quad (7)$$

An output gate decides what information we are going to output as in Eq. (8).

$$o_t = (W_o[h_{t-1}, X_t] + b_o) \quad (8)$$

Thus, output of current block is given by multiplying output gate value with new cell state passed into a tanh layer as shown in Eq. (9) [7–9].

$$h_t = o_t * \tanh(C_t) \quad (9)$$

Thus, this model is able to remember context by storing relevant information in its long-term memory.

Embedding Layer. Above Fig. 4 shows layout of our model. Before the LSTM layer, we add an embedding layer whose role is to map semantic meaning to geometric space. This is done by associating a numeric vector to every row in the dataset, such that the distance between any two vectors would capture part of the semantic relationship between the two associated rows. For example, consider two words “potato” and “panda” which are not semantically related so their vectors would be far apart but “green” and “grass” will have closer values. The output values will be in the form of vectors with related word vectors having close by distances, thus closer values. Also, vector arithmetic is applicable to these word embeddings, for example,

$$\text{Vectorman} - \text{Vectorwoman} \approx \text{Vectorking} - \text{Vectorqueen}$$

Thus, vectors are formed based on the contexts of the words. So for example, if the machine assigns -1 for gender male and $+1$ for female, eventually learning these will enable queen getting a gender dimension of $+0.97$ and king of -0.95 and for say for mango and peach sort of genderless. Other dimensions may include whether or not it is a fruit, ages, cost, is it a noun or verb, size, loyalty, etc., thus one can come up with multiple dimensions to represent an item. In our case, vectors will be formed based on how closely related two reviews (rows in the form of tokens) are. We use



Fig. 4 Layers in training LSTM model

an embedding dimension of 100 which means that each review that is in our dataset is mapped into a 100-dimension dense vector of floating numbers. The vector will look like- (0.15,... 0.23,... -0.55).

This output is fed into the input of LSTM layer, in this layer, we apply a dropout of 20% to reduce overfitting, which drops 20% neurons in this layer. Another hidden layer is added following this which uses the sigmoid function to give a binary output.

Convolutional Neural Networks. Convolutional neural networks were designed to honor the spatial structure in image data while being robust to the position and orientation of learned objects in the scene [10]. They use pixel values and feature matrix for this. The same principle can be used to help learn structure of paragraphs of words, namely the techniques invariance to the specific position of features. This is applied over one-dimensional sequence of reviews to learn its sequence in a similar fashion. So, instead of image pixels, the inputs to CNN model are sentences represented as matrix. Each row corresponds to one word, these are word embeddings, explained previously. The width of the feature filter is same as the width of the input matrix and height is varied.

The input sentence is passed into a feature detector or a filter, which are basically vectors used to learn the sequence of the statements. As the filter is sliding (or convolving), it is multiplying its weight values with the original word embedding value matrix of the review. The multiplications are summed up to a single number, which is a representative of the receptive field. After scanning, we get an array which is called as the feature map. To scan the input, we make use of 100 filters each of size 3, which forms the first convolutional layer, which are responsible for feature selection. We also used rectifier activation function to remove the linearity (if any) present in the data. Following this layer is the max-pooling layer which is used to reduce the spatial size of the representation, which in turn reduces the computation complexity of the network, this reduces the number of dimensions. Here, care is taken that the most important information is retained. The output from here is then flattened to form a 1D matrix which is fed into input of the next dense hidden layer, which uses rectifier activation function that transforms the incoming signals of the neurons and pass these signals to the input of the next hidden layer, which uses sigmoid as its activation function to classify a binary output. CNN has a special spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. Such a characteristic is useful for classification in sentiment analysis where we expect to find a certain pattern of sequences that may be present in different parts of a sentence [11, 12]. Figure 5 shows different layers used in training the CNN model.

Convolutional Neural Network + Long Short Term Memory. This model is a hybrid of the previous two models. The CNN will gather distinct features for positive and



Fig. 5 Layers in training CNN model



Fig. 6 Layers in training CNN + LSTM model

negative sentiment and these spatial features will then be learned as sequences by the LSTM layer [13].

Accordingly, we use a one-dimensional CNN and max-pooling layer after the Embedding layer which is then fed as the consolidated features to the LSTM. We use a set of 100 filters with a filter length of 3. The pooling layer used is of the standard length 2 to halve the feature map size. The max-pooled output is then fed as the input to the LSTM layer, followed by a hidden layer, which uses sigmoid as activation function to classify a binary output. Figure 6 shows layers in the model.

3 Experiments

The following three datasets were used:

1. IMDB Movie review dataset

This dataset was taken from the Kaggle Web site. It consists of 25,000 rows each containing three attributes-A unique identifier, review for a movie and its corresponding binary sentiment-1(positive) and 0(negative).

2. Twitter Dataset

This dataset was also taken from the Kaggle Web site. It consists of three columns, tweet id, tweet text and its corresponding binary sentiment. Since we are considering only binary outcomes, i.e., 0 or 1, the accuracies obtained in testing this dataset are low comparatively because we are not considering a neutral output, which may be frequent among tweets.

3. Datafiniti's Business Database

This dataset was taken from [data.world](#) Datasets. It contains data for 1000 hotels with 35,000 rows and following attributes: hotel location, name, rating, review data and review username. From this we use hotel review data and corresponding rating. Since the rating is on a scale of 5, we consider rating >2 as positive, i.e., assign binary 1 sentiment and rating <3 as negative, i.e., assign binary 0 sentiment during the pre-processing phase.

In all three cases, the models learn from the training dataset about the words which help classify a review as positive or negative. Then, it predicts the outcome (sentiment) for reviews in the test dataset. Thus, the input is the given review/tweet and the output is predicted label- 0 or 1 of that input text. To find the accuracy, we

compare the predicted labels with the actual labels of those corresponding reviews in the test dataset. Accuracy is therefore given by the formula in Eq. (10).

$$\text{Accuracy} = \frac{\text{Correctly classified rows}}{\text{Total number of rows}} \quad (10)$$

In the deep learning approach, for training we use 1 epoch with a batch size of 32 in each case, meaning we update the weights once in each hidden layer, with 32 training tuples at a time, iteratively, till all training tuples exhaust. Before training, we configure the learning process by compiling with “adam” as optimizer function for efficiently calculating the gradient descent during the backpropagation for minimizing the loss incurred during while updating weights. To calculate this loss, we use binary_crossentropy as the loss function for all our models.

We also ensured that we chose the best methods of pre-processing in each case such that accuracy is not compromised.

4 Results

All datasets were experimented and their accuracies are as obtained in Table 1. From this, we can infer that in all three cases, deep learning algorithms prove superior over traditional machine learning algorithms. The drastic characterization of change in accuracies can be seen in Fig. 7.

From the above results, it is evident that deep learning algorithms have an edge over traditional machine learning algorithms. Elaborately because, a deep learning technique learns categories incrementally through its hidden layer architecture, defining low-level categories like letters first then little higher level categories like words and then higher level categories like sentences. Each node in the hidden layer is given a weight that represents the strength of its relationship with the output and as the model develops, the weights are adjusted.

Table 1 Accuracies (in%) of each classification algorithm on three datasets

Dataset	IMDB reviews	Hotel reviews	Twitter tweets
Algorithm			
Naïve Bayes	71.98	70.9	61.28
Logistic Regression	85.48	80.12	72.90
Random Forests	84.98	80.37	72.44
LSTM	87.58	81.29	74.54
CNN	88.98	81.28	74.44
CNN + LSTM	88.28	81.19	74.92

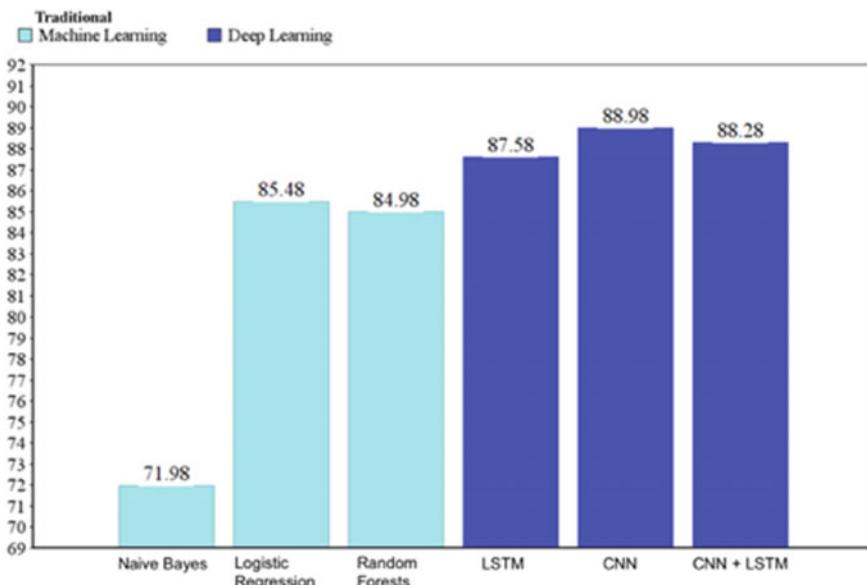


Fig. 7 IMDB movie reviews accuracies comparison for machine learning and deep learning approaches

The mascot which gives this advantage to deep learning models in NLP is the *embedding layer*, the fact that featured representation is possible using embeddings is what outsmarts them compared to traditional machine learning algorithms. Deep learning ensures that they understand the language that they are interpreting by taking word embeddings as input which retain context information and pass them to the hidden layers to learn these features during the training phase itself and thereby making more accurate predictions. Also, the ML algorithms cannot learn the sequence of words which form the sentences, thus making them less suitable for sentiment analysis tasks.

Additionally, in traditional machine learning techniques, most of the applied features need to be identified by a domain expert in order to reduce the complexity of the data and make patterns more visible for learning algorithms to work. We need to perform dimensionality reduction to find the best features to pass over the ML algorithms, which is not the case in deep learning. The biggest advantage of deep learning algorithms is that they try to learn high-level features from data in an incremental manner thus giving better performance right off the bat. This eliminates the need of domain expertise and hardcore feature extraction [14]. Manually designed features are often over-specified, incomplete and take a long time to validate whereas learned features are easy to adapt.

Though for smaller datasets traditional ML algorithms may outperform deep learning algorithms, deep learning requires sufficiently large amounts of data to work well. Also, CPU utilization is about 10 times more in case deep learning models, thus increasing the time employment.

5 Conclusion

The classification performance of the different models on movie reviews, hotel reviews and tweets gave a rough idea of the utility of these models for sentiment analysis. The performance of the deep learning models was overwhelming. Major reason for this being that the hidden layers understand the copious amount of data exceedingly well and have a deeper semantic understanding of the sequences, plus contextual relationship is retained using word embeddings which boosts the understanding of the model. In fact, they promise to perform much better as seen by the improved accuracies in the above experiments. The complementary advantage is that we can use deep learning models when there is lack of domain understanding for feature introspection, because one has to worry less about feature engineering. Thus, owing to profound contextual understanding deep learning models outperform traditional machine learning algorithms for most sentiment analysis and other NLP-based tasks.

References

1. Zhang L, Wang S, Liu BF Deep learning for sentiment analysis: a survey
2. Maite Taboada F Sentiment analysis: an overview from Linguistics. Art Annual Rev Linguistics. <https://doi.org/10.1146/annurev-linguistics-011415-04051>
3. <https://www.kaggle.com/c/word2vec-nlp-tutorial#what-is-deep-learning>. Last Accessed 30 July 2018
4. Rish I An empirical study of the naive Bayes classifier. T.J. Watson Research Center
5. Jiawei H, Micheline K, Jian P (2012) Data mining concepts and techniques. The Morgan Kaufmann series in data management systems. Morgan Kaufmann Publishers, Elsevier
6. Arras L, Montavon G, Müller KR, Samek WF Explaining recurrent neural network predictions in sentiment analysis
7. <http://deeplearning.net/tutorial/lstm.html>. Last Accessed 30 July 2018
8. <https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359>. Last Accessed 30 July 2018
9. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Last Accessed 30 July 2018
10. Hochreiter S, München, Germany, Schmidhuber J, Switzerland F Long short term memory. Neural Comput Arch 9(8)
11. Zhang Y, Wallace BCF A Sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. University of Texas, Austin
12. <http://www.joshuakim.io/understanding-how-convolutional-neural-network-cnn-perform-text-classification-with-word-embeddings/>. Last Accessed on 02 Aug 2018
13. <https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/>. Last Accessed on 30 July 2018

14. Singhal P, Bhattacharyya PF Sentiment analysis and deep learning: a survey. Singhal P, Bhattacharyya P (eds). Indian Institute of Technology, Powai
15. Timmaraju A, Khanna VF Sentiment analysis on movie reviews using recursive and recurrent neural network architectures. Stanford University
16. Cramer JJ The origins of logistic regression. Tinbergen Institute discussion paper
17. <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>

Chapter 36

Stock Price Prediction Using Grammatical Evolution



Lynette D'Mello, Aditya Jeswani and Janice Johnson

1 Introduction

Evolutionary algorithms have been shown to produce improved results in different fields, thus applying these techniques to the financial domain, specifically to stock price prediction, may provide a method to better model the market. Grammatical evolution (GE) is a grammar-based evolutionary algorithm [1] that aims to find solutions to problems by imitating the fundamental biological process of evolution. This process involves an iterative technique of evaluating the performance of a set of candidates, followed by recombination and tweaking of the best candidates in order to achieve an optimal solution. This technique of computation has relevant applications in a variety of real-world problems ranging from bioinformatics [2] to business and finance [3]. GE employs the mechanism of mapping from the search space to the solution space. The search space consists of genotypes which are essentially binary strings that represent the individuals within the population. Every genotype is mapped to a particular phenotype within the solution space. Evolution of genotype strings takes place by employing mutation or crossover. The resultant phenotype is an executable program, formatted as a tree expression. Grammatical evolution is used to radically modify the output structures by making changes to the text or employing an evolutionary strategy. This ability to easily modify the behavior by simply making changes to the original grammar supplied has made this technique highly flexible and easily applicable in various domains [4]. This flexibility provides encouragement to

L. D'Mello · A. Jeswani (✉) · J. Johnson
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: adit98@gmail.com

L. D'Mello
e-mail: lynette.dmello@djsce.ac.in

J. Johnson
e-mail: janicejohnson300@gmail.com

apply it to the problem at hand. GE is similar to Genetic Programming, having the objective to automatically generate executable programs having the best fitness value. However, GP has a drawback with regard to the representation which is restricted to single-type form in order to ensure that the property of closure is maintained throughout [5]. This drawback has successfully been overcome by GE which provides a mechanism to encode the type information, thus eliminating the type restriction. The modular design of GE consisting of the search engine, grammar and fitness function can all be viewed as simple plug-in components. This feature of modularity allows any of these components to be easily swapped in and out resulting in better usage with a range of search-algorithms preferred by the programmer.

2 Related Work

O'Neill and Ryan [1] in their paper on grammatical evolution have introduced the topic providing an insight on the benefits and the real-world applications of the technique. The paper also highlights important differences between GE and Genetic Programming, proving why GE has a broader scope in terms of handling programs with arbitrary complexities. A brief overview of the biological aspect of the algorithm has been provided for better understanding. This is followed by a detailed study of every step that takes place that is analogous to the human body. The genotype-phenotype mapping and the concept of code degeneracy have been covered in detail.

Kita et al. [6] in their paper on “Application of grammatical evolution to stock price prediction” have implemented GE with certain improvisations with respect to the search algorithm in order to improve the convergence property. This search algorithm is based on stochastic schemata exploiter, which does not require the crossover and selection operators and are still able to provide the same results as the original GE. The algorithm was applied to both GE and grammatical evolution with multiple chromosomes. The results showed a higher convergence speed as compared to that of the previous version.

Another in-depth study on grammatical evolution by O'Neill and Ryan [7] in their paper “Under the Hood of Grammatical Evolution” offers an in-depth explanation of the intricacies associated with GE. The paper focuses on the concepts of wrapping and how it can help limit the length of the genotypes. With the help of two problem domains—Symbolic Regression and Santa Fe ant trail, a study has been performed on how wrapping can affect the success rate of an optimal solution.

3 Background

3.1 Symbols and Production Rules

For any problem in GE, the initial grammar must be defined in the form of production rules. For this purpose, the Backus–Naur (BNF) notation is used [8]. This grammar comprises of a tuple— $\{N, T, P, S\}$. Here, N is the set of non-terminal symbols, i.e., the items that are a part of the language, T is the set of terminals that can be expanded using a rule, P is the set of production rules for terminals and S is the start symbol. An example of the BNF representation is:

```

N = {<expr>, <op>, <operand>, <var>}
T = {1, 2, 3, 4, +, -, /, *, x, y}
S = {<expr>}
Set of productions P:
<expr> ::= <expr><op><expr> | <operand>
<op> ::= + | - | * | /
<operand> ::= 1 | 2 | 3 | 4 | <var>
<var> ::= x(0) | y(1)

```

3.2 Mapping Process from Genotype to Phenotype

The grammar defined above specifies the syntax and constraints on the desired executable program to be produced. These programs are called phenotypes. The specifications defined could contain details of the exact programming language required or even a subset of any language.

The next requirement for mapping is a variable-length string called as genome. The string consists of genotypes; linear strings made of integers or binary values. These genotypes are initialized to a random set of numbers called a bit string. The smallest unit of this string is a chromosome and is comprised of codons (the smallest functional unit of a genotype). Each codon uses “ m ” bits to represent a single individual, making the string a list of genes.

The genotype-phenotype mapping requires the linear genome and the grammar as input. The output is the language described by the given grammar. The genome is translated as 8-bit codons. Every codon is then converted into its integer equivalent. This is analogous to the conversion of DNA to RNA during the biological process. Once the converted list of integer genotypes has been defined every genotype can be mapped to its corresponding phenotype.

3.3 Example

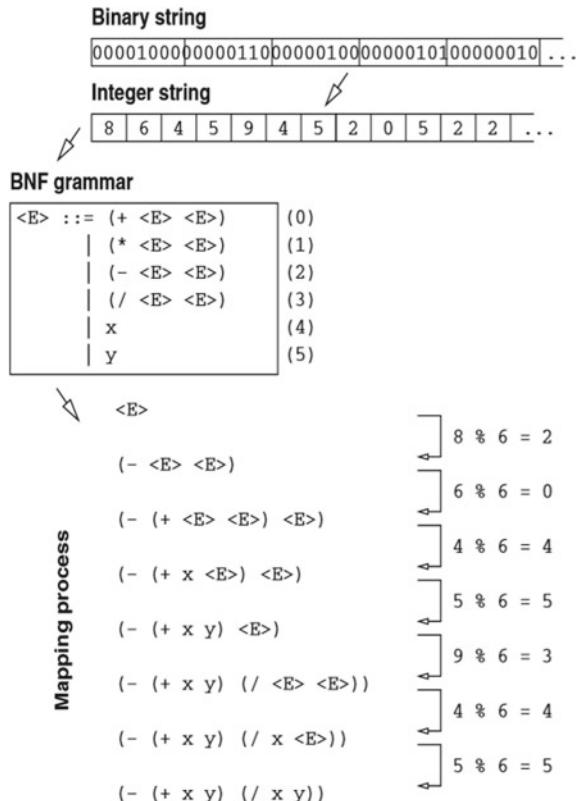
Consider the given productions for a grammar. The start symbol is considered first. With the help of the current codon “c” and the number of available productions associated with the symbol “r,” the production that will replace the non-terminal is chosen as follows.

$$\text{Rule} = c\%r \quad (1)$$

The integer value obtained is mapped to the corresponding production in the range of the number choices for the given rule.

Consider the first iteration for the grammar provided in Fig. 1. The current value of “c” will be equal to the current codon, i.e., 8. The number of productions “r” for $\langle E \rangle$ is 6. Using the formula $8 \bmod 6 = 2$, and hence $\langle E \rangle$ will be replaced with the production mapped to the number 2, i.e., ($- \langle E \rangle \langle E \rangle$). The process of mapping continues in this manner wherein at every step the leftmost non-terminal is replaced with an associated production from the rule determined by the current codon.

Fig. 1 Mapping of a genotype to the phenotype [9]



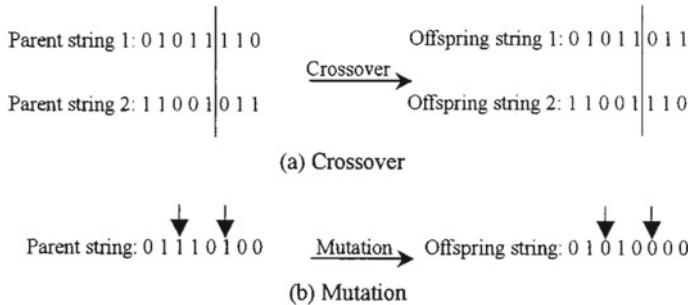


Fig. 2 Crossover and mutation to generate an offspring [10]

There could be a case where not all codons have been used for mapping, leaving the unconsumed codons as the tail of the individual. Another possibility is that codons have been consumed before the tree is entirely mapped. In this case, the reading of codons is repeated from the left-side of the genome, called wrapping.

The technique makes use of genetic operators such as mutation and crossover in order to find the optimal solution. Mutation in GE is analogous to changing one or more values in a chromosome to ensure diversity in the population. In GE, this is performed by randomly changing bits of the genetic sequence.

Crossover swaps a part of the tree within two different phenotypes. It is essentially a method that combines the genetic traits of two individuals to produce an offspring. There are different types of crossovers such as single point, two-point and uniform crossover. This ensures that the offspring is similar to each parent in some ways. An illustration of a crossover is shown below (Fig. 2).

4 Implementation

4.1 Proposed Model

The model built to apply grammatical evolution to predict stock prices is illustrated as follows (Fig. 3).

4.2 Dataset Description

The data on which the model was implemented is historical stock data of two companies—Reliance and Google. The dataset consisted of a total of seven features—Date, Open (Opening price on a day), High (Highest price on a day), Low (Lowest price on a day), Close (Closing price on a day), Adjusted Close and Volume (Volume of

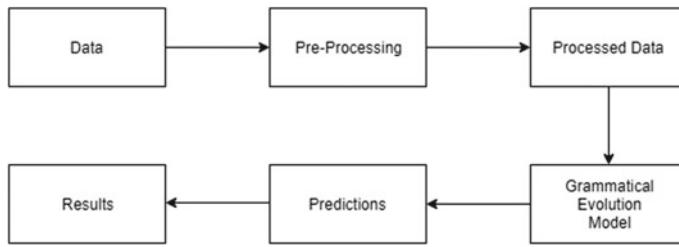


Fig. 3 Overview of the proposed system

stock traded on a day). All these features were recorded for a time period spanning 1 year.

4.3 Data Pre-processing

All the attributes which are present in the dataset do not contribute significantly to predict the price of the stock on a day. The “Volume” and “Adjusted Close” columns were dropped, after which the dataset was split into training and test sets to check efficiency of results. The split was made such that the test set consisted of 20% of the total observations, making the training dataset have the remaining 80% of the total observations.

4.4 Model Details

In order to perform grammatical evolution, we prepare two different grammars for generating the phenotype that best fits the data. The grammars defined are:

Grammar 1

```

<e> ::= <e>+<e> | <e>-<e> |
       <e>*<e> | pdiv(<e>,<e>) |
       psqrt(<e>) | np.sin(<e>) |
       np.tanh(<e>) | np.exp(<e>) | plog(<e>) |
       x[1] | x[2] |
       <c><c>. <c><c>
<c> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
  
```

Grammar 2

```

<e> ::= <e>+<e> | <e>-<e> |
       <e>*<e> | <e>/<e> |
       x[0] | x[1] | x[2] | <c><c>. <c><c>
  
```

```
<c> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

In order to implement the grammar rules and apply it to our data, we made use of the PonyGE2. PonyGE2 is primarily a Python implementation of canonical grammatical evolution, but it also includes a number of other popular techniques and EC aspects [11]. The steps involved in the evolution are as follows:

Initialization. The population is initialized to be of a fixed size (provided in the parameters passed to the algorithm). There exist different methods to perform initialization, of which, we make use of the Position Independent Grow (PI Grow) method.

Evaluation. All the individual phenotypes which are generated by the initialization process are then evaluated with the help of a fitness function defined, in this case a regression fitness function. The goal of the fitness function is to minimize the observed error between predicted and real values.

Selection. Once evaluation has been completed, we select the fittest individuals which propagate their genes to the next generation of individuals. In order to perform selection, we use the tournament selection technique.

Crossover. After selecting the individuals, One Point Crossover is done to generate new individuals. In this method, a fixed point is chosen for both the individuals at which they are split. The head of one individual is the combined with the tail of the other to generate the new population.

Mutation. While crossover provides diversity, mutation helps ensure that different scenarios are taken into account. The mutation technique used is “Int Flip Per Codon” in which we have a probability of mutation is 1 over length of the genome (Fig. 4).

5 Results and Discussion

In order to see the performance of both the grammars, we plot graphs to compare how the fitness function varies with the generations on both the datasets. Figures 5 and 6 below illustrate the reduction in fitness function values as new generations are produced.

Figures 7, 8, 9 and 10 represent the variation of the stock price predicted by the GE algorithm. The graphs show that both the grammars defined for predicting stock prices perform well on the data, making predictions to a high degree of accuracy. While both the grammars come very close in predicting values, the rules defined as a part of “Grammar 1” provides better results compared to those in “Grammar 2,” in terms of accuracy.

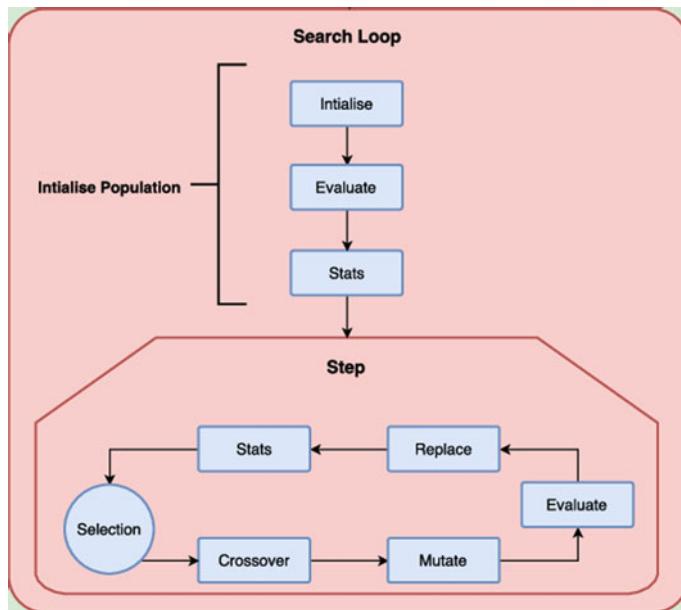


Fig. 4 Overview of the grammatical evolution process [11]

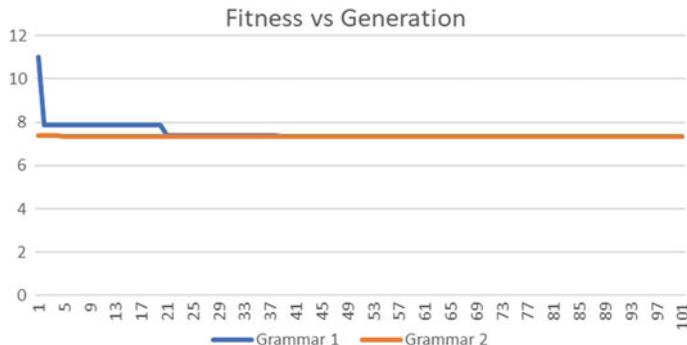


Fig. 5 Fitness evaluation on the “Google” dataset

In addition to this, “Grammar 2” takes a longer time as compared to “Grammar 1” to provide a satisfactory formula to predict the stock prices, as shown in Table 1.

Hence, due to higher accuracy as well as lesser computation time makes it feasible to use “Grammar 1” as the preferred grammar.

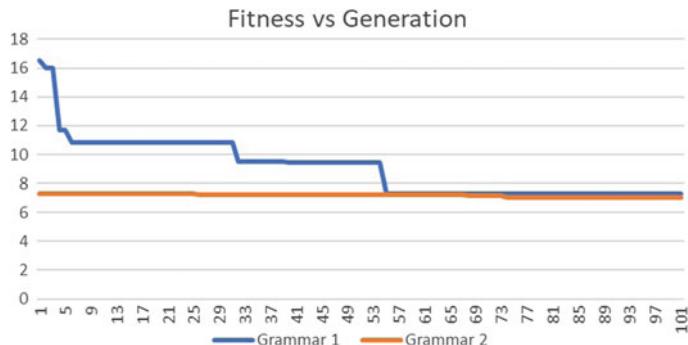


Fig. 6 Fitness evaluation on the “Reliance” dataset

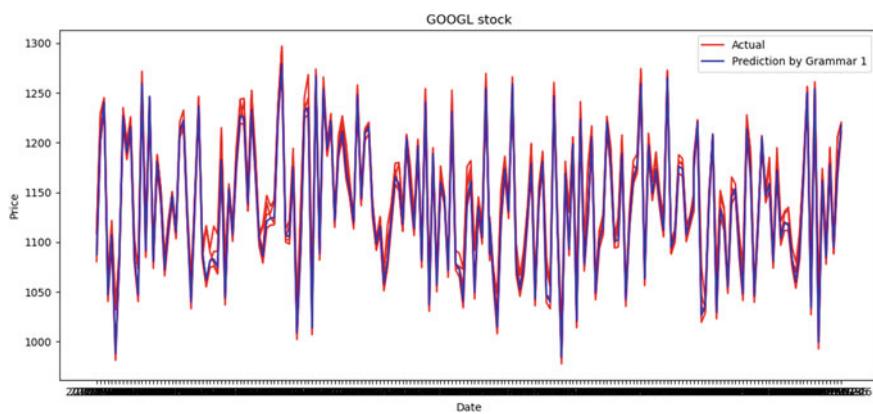


Fig. 7 Grammar 1 predictions for the “Google” dataset

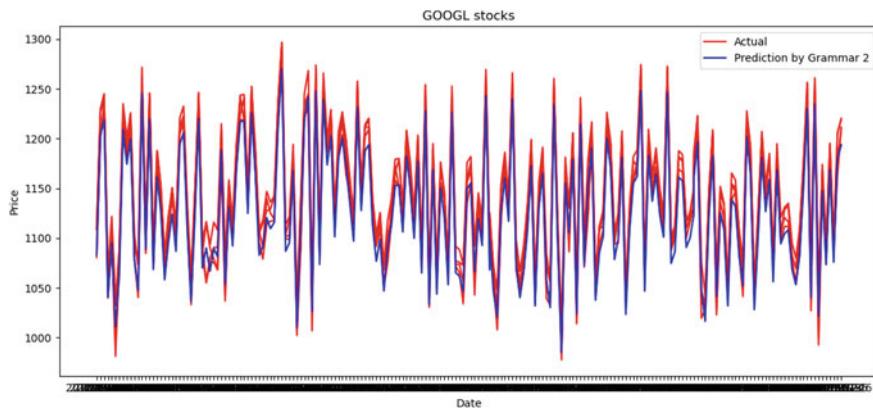


Fig. 8 Grammar 2 predictions for “Google” dataset

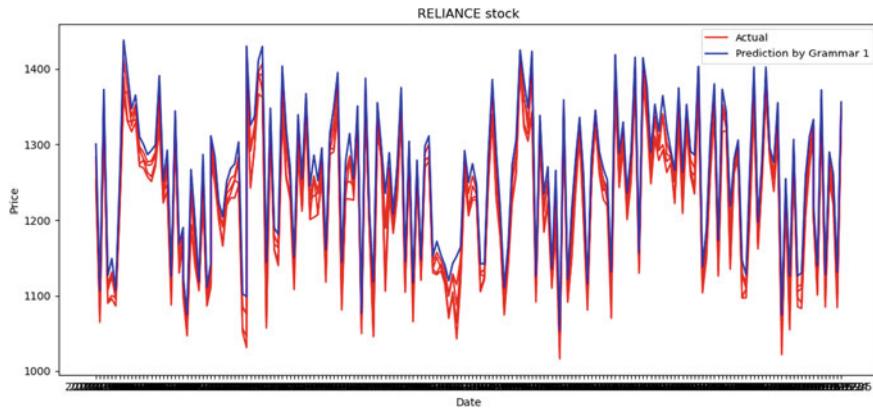


Fig. 9 Grammar 1 predictions for “Reliance” dataset

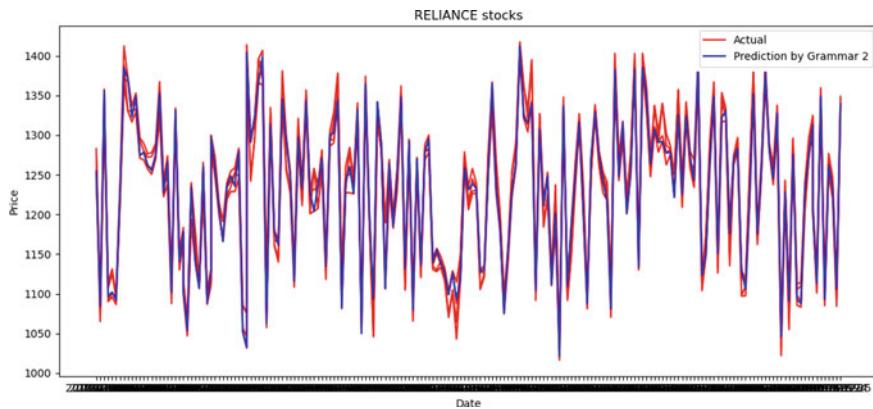


Fig. 10 Grammar 2 predictions for “Reliance” dataset

Table 1 Comparison of time performance of both grammars

	Google stock dataset	Reliance stock dataset
Grammar 1	11.07585001	12.8870542
Grammar 2	11.19103336	13.64582133

6 Conclusion

Grammatical evolution has proven to be a useful technique to obtain an optimal solution to different problems belonging to different domains. We have shown the use of this evolutionary technique in financial markets with the algorithm performing well on the datasets. The random splitting of dataset allows a variety of data to be analyzed and be tested. The GE algorithm produced satisfactory results with the

specified parameters, taking reasonable execution time and the performance can be improved by increasing the observations and providing more diverse data.

References

1. O'Neill M, Ryan C (2001) Grammatical evolution. *IEEE Trans Evol Comput* 5(4):349–358
2. Motsinger AA, Reif DM, Dudek SM, Ritchie MD (2006) Understanding the evolutionary process of grammatical evolution neural networks for feature selection in genetic epidemiology. *Proc IEEE Symp Comput Intell Bioinforma Comput Biol.* 2006:1–8. <https://doi.org/10.1109/CIBCB.2006.330945>
3. Brabazon A, O'Neill M, Dempsey I (2008) An introduction to evolutionary computation in finance. *Comput Intell Mag IEEE* 3:42–55. <https://doi.org/10.1109/MCI.2008.929841>
4. O'Neill M, Ryan C (2003) Grammatical evolution: evolutionary automatic programming in an arbitrary language. Kluwer Academic Publishers, Norwell, MA, USA
5. Whigham PA (1995, July) Grammatically-based genetic programming. In: Proceedings of the workshop on genetic programming: from theory to real-world applications, vol 16, No. 3. pp 33–41
6. Kita E, Sugiura H, Zuo Y, Mizuno T (2017) Application of grammatical evolution to stock price prediction. *Comput Assist Methods Eng Sci* 24(1):67–81
7. Neill M, Ryan C (1999) Under the hood of grammatical evolution
8. McCracken DD, Reilly ED (2013) Backus-naur form (bnf). 129–131
9. Dempsey I, O'Neill M, Brabazon A (2009) Foundations in grammatical evolution for dynamic environments. *Stud Comput Intell*
10. Yang, C-X, Tham GL, Feng X-T, Wang YJ, Lee PKK (2004) Two-stepped evolutionary algorithm and its application to stability analysis of slopes. *J Comput Civil Eng* 18(2):145–153. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2004\)18:2\(145\)](https://doi.org/10.1061/(ASCE)0887-3801(2004)18:2(145))
11. Fenton M, McDermott J, Fagan D, Forstenlechner S, O'Neill M, Hemberg E (2017) PonyGE2: grammatical evolution in Python. ArXiv, abs/1703.08535

Chapter 37

Smart Notifications Based on Total Relevancy Score



**Bhaktij Patil, Hemal Mamtora, Kunal Mandalya, Niket Parekh
and Pramod Bide**

1 Introduction

Over the duration of the last decade, it has become necessary to re-evaluate the relationship between humans and their smartphones. Although there exist multiple utilities that smartphones provide us with, they have established a firm grip on our mind that has brought along several serious problems. Some of these problems include the text neck syndrome, thumb arthritis and cubital tunnel syndrome. Prolonged smartphone usage can lead to prolonged backache and neck strain. However, smartphones themselves are not the root of the problem. It is the continuous buzzing of phones that gains user attention. A survey conducted at Deloitte in 2016 concluded that humans unlock their smartphones 47 times per day on average, and for teenagers, this number is towards 80. In 2013, Apple sent over 7.4 trillion push notifications through its servers. The intervening four years have only seen an exponential rise in this trend.

Initially, push notifications were designed with the intent of keeping users away from their phones rather than forcing them to constantly check in. The initiative for

B. Patil (✉) · H. Mamtora · K. Mandalya · N. Parekh · P. Bide

Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India

e-mail: bhaktij.patil@spit.ac.in

URL: <https://comp.spit.ac.in/>

H. Mamtora

e-mail: hemal.mamtora@spit.ac.in

K. Mandalya

e-mail: kunal.mandalya@spit.ac.in

N. Parekh

e-mail: niket.parekh@spit.ac.in

P. Bide

e-mail: pramod_bide@spit.ac.in

push notifications was led by BlackBerry when they launched push email in 2003 to reduce the amount of time users spent in constantly checking their inbox. However, app developers soon started to exploit this feature to get more user interaction. News companies started pumping out notifications to get users into their apps, so they could have an edge over their competitors. Games continuously send out notifications asking users to play more, which leads to an increase in in-app purchases. Applications whose financial model depends largely on Ad-Revenue need users to have their application open for longer periods of time. To be fair, there have been several platforms and companies that have tried to provide a solution to this ever-increasing problem over the years. Smart watches were conceived as a way to keep users from their phones, but they too turned into another device that explodes with notifications around the clock. Apple has recently made it easy to dismiss multiple notifications at a click, whilst Google has simplified the process of turning off notifications for certain applications or snoozing them for desired time. Android in its future iterations has plans to give users more control over notification management. But app developers keep figuring out exploits in the system making control over notifications extremely difficult.

At this point, notification management is a losing battle. Notification filtering is perhaps the only firm solution to this ever-increasing problem. In this paper, we propose a model that filters out notifications relevant to a user based off keywords using natural language processing and pattern mining. Since it is very difficult to limit app developers with the number of notifications being sent, we simply ignore the idea of notification control. Instead, we let all the notifications arrive at the client side and then filter out those that the user might actually find useful. The solution is not only inexpensive but also easy to implement and use.

2 Literature Survey

For the user to receive a crucial notification and to circumvent from information at involuntary times, the notification system should be such that it recognizes the user context. Awareness and Notification Service (ANS) which is based on a rule-based approach and provides notification depending on the user's context [4]. In this methodology, there is constant interaction between two entities viz ANS service and the application. The primary function of the ANS is to keep the applications aware of context and change in the environment, whereas the primary task of applications is to register monitoring rules that specify what changes in context should be notified to the users. The environment change is modelled using event condition action (ECA) rules. Awareness and Notification Service consists of a monitoring module which checks the event, and appropriate action is taken depending upon the condition. Depending upon user preference and current context, ANS provides notifications to users. The entered rules are evaluated continuously by ANS, and relevant users are notified when one of the rules evaluates to true [4]. However, this method does not include temporal event. Another approach that makes use of user context and

environmental context factor is using a modular architecture that applies machine learning algorithm to manage incoming notification. The architecture consists of three modules, viz. collector, decision maker and dispatcher. The collector collects all the user and environment context information. This information is then further sent to the decision maker module that is aware of user context, habits and environment status. Lastly, the dispatcher module adapts the information and sends the message to targeted devices. The essential characteristic of the decision maker module is that it uses a supervised learning algorithm to make decisions. Support vector machine (SVM), Gaussian Naive Bayes (GNB) and decision trees (DT) algorithms are used. SVM algorithm gave the most optimistic results in terms of prediction and accuracy [3].

Apart from context awareness, user usage pattern of the smartphone plays an essential factor in understanding and managing the notification. SmartNotify: a notification manager framework is implemented for managing notifications. For efficient management of notification, datasets are considered in two phases. One dataset consists of an analysis of the current situation of notification disruption, and another dataset consists of feedback of user for the understanding of the notification usage. How frequently user receives and checks notification, time taken to read notification and user behaviour while engaging with the notification is an important factor that assists in understanding user relevant notification. Notification manager uses supervised machine learning-based prediction engine to infer the importance of the notifications. About 87% accuracy was obtained [8].

To get active user engagement, the notifying application requires to send relevant notifications to the user. The relevance of a notification was identified by using response prediction models that predicted users engagement on notifications [5]. The following prediction models were used to predict user engagement with the two different types of notifications. pClickPush [5] model predicted the probability of a user clicking a push notification, and pClickInapp [5] model predicted the probability of user engaging with notification within the application. The relevance of the notification was determined by these predictions. These models took into account four classes of features to capture the relevance of a notification. They were actor features, item features, recipient features and edge features [5]. These class of features considered various characteristics of sender, receiver, the sent message and the relationships between the sender and the receiver. It focused on LinkedIn as a single application and tried to improve user engagement with the same application by sending near real-time notification. However, from a user's perspective, it does not consider other applications and the associated notifications.

TF-IDF is used to get the importance of each word in the document, and cosine similarity is performed based on the TF-IDF matrix value. Words are then clustered based on the similarity value [1]. However, TF-IDF has its disadvantage; it considers documents that are keyword similar. Using TF-IDF, the clustering algorithm will fail miserably because they only share one keyword even though they both talk about the same topic. Another pattern mining approach is by considering the weight of each term according to distribution of word in the document [6]. Another proposed technique uses two processes, pattern deploying and pattern evolving, to refine the

discovered patterns in text documents. The experimental results show that the proposed model outperforms not only other pure data mining-based methods and the concept-based model but also term-based state-of-the-art models, such as BM25 and SVM-based models [9]. Another approach in context awareness is detecting user context transitions that might provide recognizable and available moments and predicting these moments and providing a notification message [7].

3 Methodology

The task at hand was to notify users only when a message of certain relevance arrives. To calculate the relevance of the incoming message with respect to the user, we initially take into consideration the messages that the user had previously marked important. We then use these messages to train our model to determine the relevance scores of messages to arrive. Python 3 along with its Gensim and NLTK libraries was used for the experimental setup. The method followed can be divided into the following subsections.

3.1 Data Preprocessing

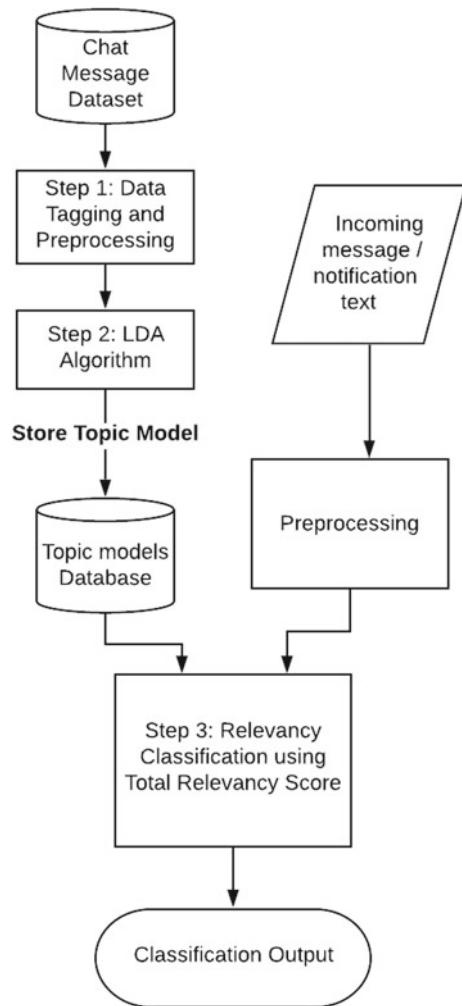
The dataset used to build the model was taken from the Free Code Camp channel of the Gitter messenger and has over five million chat messages that can be analysed. To simulate messages important to a particular user, 500 messages were selected at random and were manually tagged into categories. We then treat each category of message to be of relevance to a unique user. There is, however, a need for heavy preprocessing before this data is ready for use. The messages acquired from the dataset have several special characters, emotes, emails and other links within them. We start off by removing all the special characters and symbols, including emotes. Once we have just the text data left to use, we remove any URLs and emails present in this text along with username mentions.

This is followed by the removal of stop words and separation of the remaining words in the sentence into tokens. These tokens are then lemmatized to get the root word while still maintaining its semantics. We then create a corpus from these tokens, and the data is ready for analysis (Fig. 1).

3.2 LDA

Latent Dirichlet allocation (LDA) is a probability-based model of a corpus of documents, where documents are treated as a mixture of topics and topics are treated as mixture of words. LDA uses bag of words to form its corpus. It selects and gives

Fig. 1 Smart Notification Classification.
 (Preprocessing and Data Tagging steps give us the list of topics for specific users. LDA [2] is used to get topic models. Using smart relevancy score and the generated topic model, an incoming notification is classified as relevant or not.)



a probability distribution of topics over documents. Similarly, it gives probability distribution of words over topics. While LDA has predominantly been used in natural language processing for topic modelling purposes, it also excels at providing the terms most relevant to a topic. We have thus re-purposed LDA to obtain the most relevant terms and their relevancy scores with respect to the topic. Experimentation was also done to ensure that the obtained probability scores were realistic, and the words obtained were actually relevant. We run ten passes of LDA over the corpus and designate it to find the 50 most relevant words along with their relevancy scores for each of the topics that are pre-classified. The model can now be used for relevancy classification.

3.3 Relevancy Classification

In this module, we classify the incoming messages into a category, if its total relevancy score(TRS) is beyond a defined threshold. Every time, a new text message is received, the preprocessing steps mentioned above are followed to obtain lemmatized tokens. For each of these tokens, a relevancy score is obtained as follows. The token is cross-referenced across the corpus of top 50 words across all the categories. If it is present in one or more corpora, its probability distribution obtained from LDA is considered to be its relevancy score. Otherwise, it is set to zero. After obtaining the individual relevancy scores of each of the tokens, we calculate the total relevancy score for the message. The total relevancy score for a category is obtained via the summation of relevancy scores of all tokens in a document upon division with the square root of its length as depicted in Eq. 1

$$\text{TRS}_c = \frac{\sum_{i=1}^n R_i}{n^{0.5}} \quad (1)$$

The TRS of a message across each of the categories is calculated, and the highest amongst them is selected to be the final TRS of a message. The message is classified according to the category corresponding to the highest TRS if its TRS is beyond the defined threshold for the category.

4 Results

The TRS for messages ranged between 0 and 0.25 with an average TRS of 0.07. A sample from the classified data is mentioned in Fig. 2. along with its corresponding label and total relevancy score.

The model was tested across 4000 randomly selected text messages from the Gitter dataset. Out of these, 316 messages were classified incorrectly. Three thousand three hundred and eighty-seven messages were correctly classified, whilst the model was unable to classify the remaining 613 text messages as their TRS was below the

Message	Label	Score
@arreche Welcome to our community!	greeting	0.17
I rarely eat sugar, but when I do... I'm still chill. #FarTooLaidBack	food	0.09
I used to really dig this microwave brownie that we make sometimes	food	0.07
@Lumiras do you log in with github?	computer	0.02
Welcome Travis!	greeting	0.11
So how many different non-profit projects are there right now?	work	0.02
Good Morning Campers. Happy New Year. Watching the Rose Parade and studying.	greeting	0.06

Fig. 2 Relevancy scores and classes

required threshold. These 613 messages lie in the grey area that consists within the model and are considered as important by the system. This grey area, however, can be reduced to a large extent by increasing the amount of training data. The system performs extremely well with an accuracy of 90.67% when the 613 grey area messages are ignored. However, the performance of model is severely dependent on the amount of training data. We estimate that the model will significantly improve with additional data based on previous tests that were done on lesser data.

5 Conclusion

A smarter way of delivering notifications has become a necessity today. Fewer notifications help users maintain a healthy lifestyle and reduce stress, annoyance and time consumed. It can be realized through the results that on average, only one in every ten notifications from text messages is relevant to an user and over 90% of time can be saved with better notification management. Total relevancy score proves to be a reliable metric to calculate the relevance of a text document and can be further used over applications other than text messages and social media.

6 Future Scope

Currently, our model supports classification of the notification text that contains words only in English. This model could be developed for support of multiple languages thereby targeting a larger audience. Also, currently, it supports only text data, and image processing could be used to classify the documents and images we receive as well. A semantic-based, context-aware notification system could be developed using this model, thus alerting the users with the notifications based on their preference.

References

1. Agnihotri D, Verma K, Tripathi P (2014) Pattern and cluster mining on text data. In: 2014 fourth international conference on communication systems and network technologies. IEEE, New York, pp 428–432
2. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
3. Corno F, De Russis L, Montanaro T (2015) A context and user aware smart notification system. In: 2015 IEEE 2nd world forum on internet of things (WF-IoT). IEEE, New York, pp 645–651
4. Etter R, Costa PD, Broens T (2006) A rule-based approach towards context-aware user notification services. In: 2006 ACS/IEEE international conference on pervasive services, pp 281–284. <https://doi.org/10.1109/PERSER.2006.1652242>
5. Gao Y, Gupta V, Yan J, Shi C, Tao Z, Xiao P, Wang C, Yu S, Rosales R, Muralidharan A et al (2018) Near real-time optimization of activity-based notifications. In: Proceedings of the 24th

- ACM SIGKDD international conference on knowledge discovery & data mining. ACM, New York, pp 283–292
- 6. Gupta SD, Vasgi B (2015) Implementation of pattern discovery to retrieve relevant document using text mining. In: 2015 international conference on green computing and internet of things (ICGCIoT). IEEE, New York, pp 327–332
 - 7. Oh H, Jalali L, Jain R (2015) An intelligent notification system using context from real-time personal activity monitoring. In: 2015 IEEE international conference on multimedia and expo (ICME). IEEE, New York, pp 1–6
 - 8. Pradhan S, Qiu L, Parate A, Kim KH (2017) Understanding and managing notifications. In: IEEE INFOCOM 2017-IEEE conference on computer communications. IEEE, New York, pp 1–9
 - 9. Zhong N, Li Y, Wu ST (2010) Effective pattern discovery for text mining. *IEEE Trans Knowl Data Eng* 24(1):30–44

Chapter 38

Ensemble Method Combination: Bagging and Boosting



Jyoti Deshmukh, Mukul Jangid, Shreeshail Gupte, Siddhartha Ghosh and Shubham Ingle

1 Introduction

Machine learning using a single model is more prone to error and is less efficient. Error is disintegrated into 3 terms: bias, variance and irreducible error term. Bias is contributing error due to model simplifying the learning function and variance is contributing error due to learner making the function more complex. Ensemble learning is a better method of utilizing the notion that combining the output of different models produces more accurate results. The success in ensemble learners lies when the sub-ensemble classifiers are able to overcome the failure of each other and generate diverse predictions. Two popular ensemble learning methods: boosting and bagging are applied successfully to classification problems.

Boosting and bagging are homogenous ensemble methods. They use the same base learning algorithm but the dataset distribution is changed. Data can be distributed by sampling the instances, another approach is by dividing the data into feature subsets. Features are selected by evaluating which feature are more relevant for classifying the output. Features can be evaluated individually or as a subset. Creating subsets requires the grouping of features in different sizes. Evaluating a subset requires more computation and time.

J. Deshmukh · M. Jangid (✉) · S. Gupte · S. Ghosh · S. Ingle

Department of Computer Engineering, Rajiv Gandhi Institute of Technology, Andheri West, Mumbai, Maharashtra 400053, India

e-mail: mkljngd@gmail.com

J. Deshmukh

e-mail: jyotideshmukh11@gmail.com

S. Gupte

e-mail: shreeshail04@gmail.com

S. Ingle

e-mail: shubhamingle@yahoo.com

In the proposed method boosting is used in bagging algorithm as a base learner. Boosting itself will use its own base learner C4.5 [1]. Boosting is affected by variance error. Using multiple boosting on different data subsets makes the combined model have less variance. In this, the number of instances given to each boosting algorithm remains the same. In the results section, the results of the proposed method are compared with the combined method and individual bagging and boosting on standard datasets.

Ensembling is a process of merging at least two procedures of homogeneous or heterogeneous nature called base learners. This increases the robustness of the system which accepts the predictions from all the base learners. It is like a committee among various subcommittees to arrive at a conclusion on the problem. All of them have a different outlook on the problem thus a dissimilar mapping function from the problem statement to the favorable outcome. As a result, they have to make different predictions on the problem according to their own outlooks.

The final decision is made by accounting on all the predictions. Hence the output decision will be precise, less bias and robust. If one of these sub-committees decided this alone, the final decision might be contradictory. A single learner might not be able to predict with great accuracy. But the numerous feedbacks of multiple learners usually increase the accuracy. In the regression problem or the classification problem, the mean of probability predictions from the models are calculated by Averaging.

Majority votes consider the prediction with the highest vote count from various models predictions while predicting the results of a classification problem [2].

In weighted average, different weights are taken which are to be applied for predictions from various models, then considering the mean which results in providing high or low importance to specific model output.

2 Related Work

2.1 Bagging as an Ensemble

The paper “A Bagging Method using Decision Trees in the Role of Base Classifiers” proposed by the authors Kristna Machov, Frantiek Bark, Peter Bednr described a list of experiments with bagging—a process that can be used to enhance the results of the classification algorithm [3]. Their use of this process aims at classification algorithms implying decision trees. Bagging procedure improves the performance of classification using the minimum number of the decision trees.

2.2 *Boosting as an Ensemble*

The authors Thakkar et al. Proposed a paper on “Boost a Weak Learner to a Strong Learner Using Ensemble System Approach” which demonstrated the weak learner’s potential to reduce the rate of an error on the testing data and the ability of boosting algorithm to reduce the error rate of the weak learner [4]. In the experiment, they utilized decision stump as a weak learner (classifier) and by utilizing the boosting approach, the output denotes the enhancement in the classifier’s accuracy. The boosting meta-algorithm is a simple and well-organized model building strategy. Boosting motivates new model to become specialists, for instance, handle wrongly by earlier ones. A model’s contribution is weighted by its performance than giving equal weight to all the methods. Boosting algorithms creates various models from a dataset, consisting of some model builder techniques such as a decision tree builder which may not be a good model builder. The basic idea of boosting is that each entity in the dataset is mapped with a weight. If a model incorrectly classifies the entity, then a series of models are built and the weights are increased (boosted). The final model is then a sum of different models built from the sequence of models in which each model’s output weighted by little score.

2.3 *Techniques for Combining Bagging and Boosting*

In [5], the authors proposed a methodology in which the expected error of a learning procedure on a specific target function having training set size of 3 components: [1] A bias term computing how close the mean classifier makes by the learning procedure will be to the target function. [2] A variance term computes the amount of the every learning algorithms estimates how often they disagree. [3] A variable that measures the least classification error that is related with the Bayes optimal classifier for the target function. For enhancing the prediction of a classifier, authors suggest arranging bagging and boosting methodology with sum rule voting (Vote BB). All sub-ensemble gives a confidence value for each candidate class. In the proposed method, the voters express the degree of their preference as confidence weighs the chances of sub ensemble prediction. Later, confidence values are summed for each candidate and the candidate with the maximum sum proves to be superior.

MultiBoosting [6] is another method proposed by Webb G. I. which is acknowledged as wagging committees formed by AdaBoost. Wagging is a aberrant version of bagging, as wagging makes use of re-weighting for every training example and bagging uses resampling to receive the datasets.

BagBoo A Scalable Hybrid Boosting and Bagging Model [7] proposed by Gorodilov et al. is another method for combining bagging and boosting into a hybrid approach. The only important change from bagging and boosting is that they bagged boosted models which allowed the resulting model to be quite a bit more powerful.

3 Proposed Methodology

A comparative study on bagging [8–10] and the boosting [11, 12] algorithms is performed and combined them to eliminate both of their flaws. In the proposed method, \hat{f} features are selected from the originally present f features. Then s number of subsets is randomly sampled without replacement from the total subsets generated of \hat{f} features. The same number of samples are present in all the subsets. The boosting algorithm is fitted on each training subset giving s learned hypothesis function. Prediction is made on test tuple by all the models. For each class a confidence value which is the probability between 0 and 1 is returned, this is summed and the maximum confidence class is predicted output. Algorithm 1 presents Random attribute subset selection

Algorithm 1: Random attribute subset selection

a. **Input.**

Set all training sets $\{X_i, Y_i\}$ where $i = 1$ to n

X consists of different features i.e. (X_1, X_2, \dots, X_k) , Y is the target variable

b. **Procedure.**

Apply Base Learning Algorithm as AdaBoost

Generate all combinations of attributes of length m

Subset: Select s number of random subsets

c. **For i to s do**

$M[i] := \text{AdaBoost}(\text{subset}[i])$

end for

d. **Output.:** $\text{argmax } \sum_i M[i]$

Figure 1 describes the workflow of the proposed methodology which starts with the training of the dataset including cleaning by filtering the dataset. N bags of the dataset are created, each portraying random attributes and applying boosting on each bag and later combining the results to give the predicted value.

4 Experiment Results

Experiments are performed in Python. The experiments are carried out on 64-bit Core i5–7th generation 3 GHz processor, 8 GB RAM. In the experiment, the aim is to compare the proposed method of integrating boosting consisting random feature subset selection with bagging, boosting and sum vote combining [5] classifiers. The 22 datasets of various domains are selected from UCI repository [13] containing a different number of instances, types of features: numerical as well as categorical and

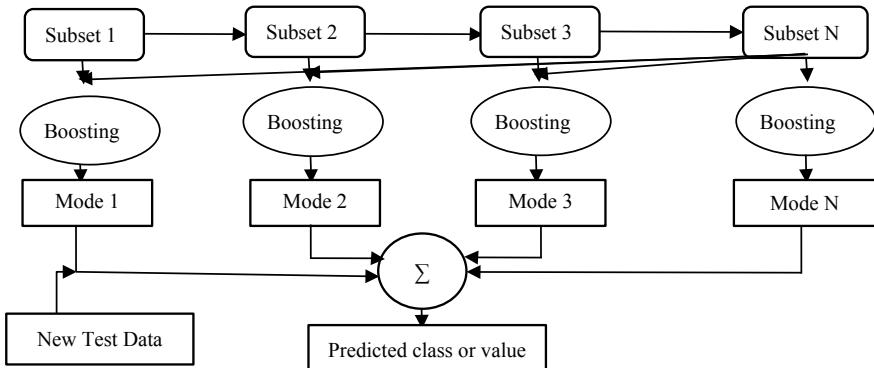


Fig. 1 Flow diagram of proposed method

class distributions of two till ten classes. The base sub-classifier CART algorithm induced decision tree is used for all the methods. In the literature [5] the authors compared several base classifiers like Decision tree, Decision stump, Bayesian algorithm and Rule Learner on various datasets and concluded that Decision tree gave the best results compared to the others. As referring to the literature, Decision Tree is used as our base estimators in the proposed method. Equal count of base sub-classifier is combined in all the models for making the results comparable equally. The accuracy is estimated by taking an average of tenfold stratified cross validation. The database is split into ten folds by sampling and rearranging without replacement. The classifier algorithm is tested on each fold and trained on the remaining nine folds. Other evaluation metrics—recall, precision, and specificity are computed for each database.

Experiment is first tested by changing the number of features selected and the number of subsets selected to decide the proportion of randomly picked feature subsets from feature vector giving the optimal result. The time taken for running the proposed algorithm takes more time than bagging and boosting due to algorithm running on additional data subspaces. Increasing the number of bags or subsets, consumes more time for training. It can be parallelized easily to decrease the running time.

Table 1 describes databases which are used to express the count of instances, categorical features, numerical features and classes. Here some well-known datasets are used for the comparison of the results.

Various experimentation is performed on all the datasets with a various count of base estimators. It is found that when the count of base classifiers is 5, the optimum result is achieved with least time and space complexity as shown in Table 2.

Bagging, boosting and combining using vote B and B are executed on the different datasets, for each method accuracy, recall, precision and specificity quality evaluation metrics are estimated and listed in Tables 3, 4 and 5.

Figure 2 visually represents the results of vote bagging and boosting ensemble on datasets.

Table 1 Description of datasets

Dataset	Instance	Categorical features	Numerical features	Classes
Anneal	898	32	6	6
Australian	690	8	6	2
Breast-cancer	699	0	9	2
Car	1748	6	0	4
Ecoli	336	1	7	8
Glass	214	0	10	7
Haberman	306	0	3	2
Heart-statlog	270	0	13	2
Heart-h	297	7	6	5
Hepatitis	155	13	6	2
Liver	583	1	9	2
Mobile	2000	5	15	3
Nursery	12,960	8	0	5
Somhappy	143	6	0	2
Teaching assistant	151	4	1	3
Tic-Tac-Toe	958	9	0	2
Travel-insurance	63,326	6	4	2
Vertebra	310	0	6	4
Wifi-localization	2000	0	7	4
Wine	178	0	13	3
Wine-red	1599	0	11	10
Wine-white	4899	0	11	10

Table 2 Comparative analysis of the number of base classifiers with Breast-cancer, Car and Haberman datasets

No. of base classifiers	Breast-cancer	Car	Haberman
5	0.9557	0.8825	0.7192
55	0.9528	0.8663	0.6960
105	0.9500	0.8674	0.6794
155	0.9514	0.8692	0.6796
205	0.9500	0.8709	0.6926
255	0.9500	0.8657	0.6991
305	0.9486	0.8663	0.6990
355	0.9486	0.8657	0.6926
405	0.9486	0.8657	0.6926
455	0.9486	0.8640	0.6958
505	0.9486	0.8634	0.7024

Table 3 Result of bagging classifier using decision tree

Dataset	Bagging			
	Accuracy	Recall	Precision	Specificity
Anneal	0.8872	0.7490	0.7608	0.9395
Australian	0.8347	0.8343	0.8352	0.8343
Breast-cancer	0.9472	0.9441	0.9428	0.9441
Car	0.8592	0.7959	0.8497	0.9312
Ecoli	0.8406	0.6792	0.7199	0.9630
Glass	0.9263	0.9163	0.9182	0.9840
Haberman	0.6446	0.5310	0.5263	0.5310
Heart-statlog	0.7704	0.7667	0.7722	0.7667
Heart-h	0.7747	0.7721	0.7829	0.7721
Hepatitis	0.8250	0.7138	0.7157	0.3138
Liver	0.6775	0.5966	0.6181	0.5966
Mobile	0.8740	0.8740	0.8765	0.9580
Nursery	0.7598	0.7541	0.7858	0.9151
Somhappy	0.4638	0.4733	0.4742	0.4733
Teaching assistant	0.6627	0.6617	0.6625	0.8307
Tic-Tac-Toe	0.8698	0.8777	0.8845	0.8777
Travel-insurance	0.9774	0.5108	0.5230	0.5108
Vertebra	0.8387	0.7911	0.8137	0.9227
Wifi-localization	0.9730	0.9730	0.9745	0.9910
Wine	0.9332	0.9398	0.9424	0.9666
Wine-red	0.5372	0.2671	0.2756	0.8762
Wine-white	0.4548	0.2424	0.2830	0.8689

Comparison of the 3 ensemble methods is performed with the proposed method whose result is shown in Table 6. It contains the evaluation metrics, number of attributes taken and number of bags which is the subsets selected randomly from different combinations of attributes.

Figure 3 visually represents the results of proposed method on different datasets.

The proposed method has higher accuracy in 10 out of 22 datasets for bagging, 15 for boosting and 8 for vote B and B method. Comparing the Tables 4 and 6 it is seen that presented classifier gives better precision in 15 out of 22 datasets than boosting alone. Precision is better in 4 datasets for both bagging and combine vote B and B method. It states that positive predictions made by the proposed classifier are more relevant. The accuracy values also depend upon the random features selected. A feature can contribute more to the classifier prediction than others.

Table 4 Result of Adaboost classifier using decision tree

Dataset	Boosting			
	Accuracy	Recall	Precision	Specificity
Anneal	0.7947	0.3970	0.3663	0.8223
Australian	0.8579	0.8582	0.8593	0.8582
Breast-cancer	0.9400	0.9287	0.9409	0.9287
Car	0.7991	0.4544	0.3857	0.9345
Ecoli	0.6407	0.3831	0.2758	0.9089
Glass	0.7773	0.4820	0.4395	0.9411
Haberman	0.7160	0.5876	0.5643	0.5876
Heart-statlog	0.8148	0.8108	0.8232	0.8108
Heart-h	0.8142	0.8127	0.8208	0.8127
Hepatitis	0.8250	0.7204	0.7215	0.3204
Liver	0.6705	0.5459	0.5850	0.5459
Mobile	0.6225	0.6225	0.7195	0.8742
Nursery	0.7334	0.5395	0.4575	0.9063
Somhappy	0.5433	0.5407	0.5430	0.5407
Teaching assistant	0.4749	0.4739	0.4998	0.7376
Tic-Tac-Toe	0.7111	0.6420	0.7010	0.6420
Travel-insurance	0.9854	0.5000	0.4927	0.5000
Vertebra	0.6742	0.6489	0.6876	0.8438
Wifi-localization	0.8750	0.8750	0.8873	0.9583
Wine	0.8955	0.9015	0.9269	0.9466
Wine-red	0.5304	0.2132	0.1792	0.8684
Wine-white	0.4276	0.1769	0.1431	0.8540

5 Conclusion and Future Scope

Authors compared the ensemble methodologies of bagging and boosting with the proposed methodology. Further, authors also compared the proposed methodology to that of combining bagging and boosting using VOTE B and B method. After the comparison, it reveals that in almost all the cases it got better results than boosting alone. In 8 datasets (Australian, Breast-cancer, Haberman, Heart-statlog, Heart-h, Liver-IND, Somhappy, Travel Insurance) the proposed methodology of combination gave better results, whereas the combination using VOTE B and B gave better results in the remaining 14 datasets. Further, the authors stated that, boosting works better compared to bagging on noiseless dataset, whereas bagging is more efficient compared to boosting on the data containing noise. The methodology the authors proposed has been proved to give better accuracy and achieve a lower error rate in general than bagging and boosting considering Decision Tree as the base classifier.

Table 5 Result of vote bagging and boosting ensemble

DAtaset	Vote B and B			
	Accuracy	Recall	Precision	Specificity
Anneal	0.8872	0.7315	0.7844	0.9302
Australian	0.8405	0.8402	0.8415	0.8402
Breast-cancer	0.9429	0.9358	0.9406	0.9358
Car	0.8626	0.7880	0.8455	0.9350
Ecoli	0.8343	0.6657	0.7211	0.9616
Glass	0.9263	0.9163	0.9182	0.9840
Haberman	0.6478	0.5332	0.5272	0.5332
Heart-statlog	0.7889	0.7850	0.7913	0.7850
Heart-h	0.8012	0.7983	0.8104	0.7983
Hepatitis	0.8375	0.7700	0.7657	0.2700
Liver	0.6879	0.6039	0.6299	0.6039
Mobile	0.8685	0.8685	0.8740	0.9562
Nursery	0.7648	0.7585	0.8043	0.9170
Somhappy	0.4705	0.4760	0.4721	0.4760
Teaching assistant	0.6961	0.6950	0.6985	0.8474
Tic-Tac-Toe	0.8698	0.8777	0.8845	0.8777
Travel-insurance	0.9808	0.5062	0.5259	0.5062
Vertebra	0.8355	0.7911	0.8305	0.9206
Wifi-localization	0.9750	0.9750	0.9766	0.9917
Wine	0.9617	0.9663	0.9671	0.9811
Wine-red	0.5516	0.2739	0.2774	0.8807
Wine-white	0.4652	0.2344	0.2922	0.8691

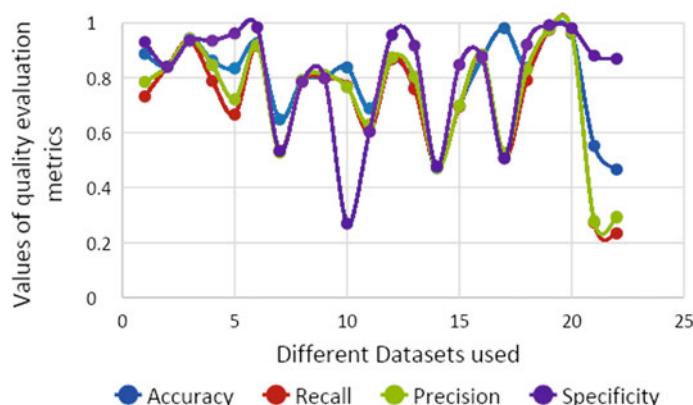
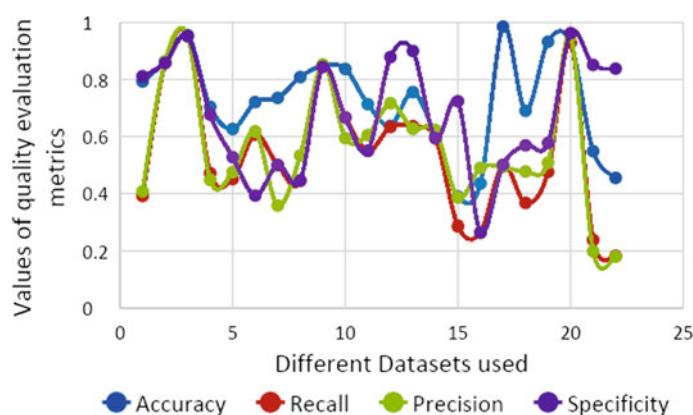
**Fig. 2** Visualization of results of vote bagging and boosting ensemble

Table 6 Result of proposed method on datasets

Dataset	Accuracy	No. of bags	No. of attributes	Recall	Precision	Specificity
Anneal	0.7933	50	5	0.3915	0.4070	0.8117
Australian	0.8623	1201	8	0.8581	0.8638	0.8581
Breast-cancer	0.9557	50	5	0.9524	0.9529	0.9524
Car	0.7041	8	3	0.4713	0.4479	0.6768
Ecoli	0.6263	28	4	0.4506	0.4750	0.5275
Glass	0.7206	84	6	0.6063	0.6179	0.3926
Haberman	0.7355	1	1	0.5000	0.3577	0.5000
Heart-statlog	0.8084	686	7	0.4453	0.5333	0.4453
Heart-h	0.8481	686	7	0.8444	0.8529	0.8444
Hepatitis	0.8369	775	15	0.6667	0.5942	0.6667
Liver	0.7133	50	5	0.5499	0.6051	0.5499
Mobile	0.6370	387	16	0.6332	0.7165	0.8786
Nursery	0.7559	28	4	0.6360	0.6268	0.9000
Sonhappy	0.6022	8	3	0.5940	0.6226	0.5940
Teaching assistant	0.3896	4	3	0.2867	0.3857	0.7236
Tic-Tac-Toe	0.4355	50	5	0.2651	0.4900	0.2651
Travel-insurance	0.9854	50	5	0.5000	0.4927	0.5000
Vertebra	0.6903	8	3	0.3672	0.4771	0.5689
Wifi-localization	0.9325	14	4	0.4758	0.5081	0.5773
Wine	0.9287	57	10	0.9302	0.9507	0.9629
Wine-red	0.5485	184	6	0.2391	0.1992	0.8512
Wine-white	0.4545	184	6	0.1831	0.1805	0.8381

**Fig. 3** Visualization of results of proposed method on datasets

Acknowledgements The authors intend to convey gratefulness to David Aha and co-founders, for providing various UCI datasets for our work. Authors would also like to thank Anthony Goldbloom and Ben Hamner for providing Kaggle datasets.

References

1. Quinlan JR (1993) C4. 5: program for machine learning. Morgan Kaufmann, San Francisco
2. Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Mach Learn* 36(1–2):105–139
3. Machová K, Barcák F, Bednár P (2006) A bagging method using decision trees in the role of base classifiers. *Acta Polytech Hung* 3(2):121–132
4. Vaghela VB, Ganatra A, Amit Thakkar A (2009) Boost a weak learner to a strong learner using ensemble system approach. In: 2009 IEEE international advance computing conference, IEEE. pp 1432–1436
5. Kotsiantis SB, Pintelas PE (2007) Combining bagging and boosting. *Int J Mathe Comput Phys Electr Comput Eng* 1(8):372–381 (2007)
6. Webb GI (2000) Multiboosting: a technique for combining boosting and wagging. *Mach Learn* 40(2):159–196
7. Pavlov DY, Gorodilov A, Brunk CA (2010) BagBoo: a scalable hybrid bagging-the-boosting model. In: Proceedings of the 19th ACM international conference on Information and knowledge management. ACM
8. Breiman L (1996) Bias, variance, and arcing classifiers. Tech. Rep. 460, Statistics Department, University of California, Berkeley, CA, USA
9. Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140
10. Breiman L (1984) Classification and regression trees, 1st edn. Routledge, New York
11. Freund Y, Schapire RE (1996, July) Experiments with a new boosting algorithm. In: International conference on machine learning. pp 148–156
12. Schapire RE, Freund Y, Bartlett P, Lee WS (1998) Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann Stat* 26:1651–1686
13. Blake C, Merz CJ (1998) UCI repository of machine learning databases [Machine-readable data repository]. University of California, Department of Information and Computer Science, Irvine, CA

Chapter 39

YOLO Based Recognition of Indian License Plates



Jimit Gandhi, Purvil Jain and Lakshmi Kurup

1 Introduction

Automatic License Plate Recognition (ALPR) has been a frequent topic of research due to its applications in intelligent transportation and surveillance systems, automatic traffic law enforcement, toll violation, detection of stolen vehicles, parking and border control.

We divide the ALPR problem into two subtasks: License plate detection (LPD) and optical character recognition (OCR). Most existing ALPR systems focus on either of the two tasks [1].

First, we use transfer learning by using the pre-trained weights for YOLO, using the DarkNet framework. Transfer learning involves using the knowledge obtained while solving a problem into a different problem but related to the original problem. Since we have a small dataset, this approach comes in handy. We use the DarkNet as our pre-trained model and since it is already trained on a large dataset, we can keep the weights of the initial layers of the neural network fixed, and then train on our custom dataset.

DarkNet is a neural network framework written in C and CUDA which supports GPU computation. It is installed with GPU dependency which we use with Google Colab which allows us to perform computations on a Tesla K80 GPU with 12 GB of RAM. We first download the pre-trained weights, train our dataset on it and then run the detector. The detector outputs the objects, the confidence and the time it took to detect the objects.

J. Gandhi (✉) · P. Jain · L. Kurup
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: jimitgandhi@outlook.com

P. Jain
e-mail: jainpurvil98@gmail.com

L. Kurup
e-mail: lakshmi.kurup@djsce.ac.in

We first use YOLOv2 and then YOLOv3 as our base object detector to properly understand each model. YOLOv2 uses DarkNet-19 as its feature extractor which has shortcut connections and its object detector use feature map upsampling and concatenation. YOLOv3 uses the DarkNet-53 as its feature extractor.

Optical character recognition is used to obtain the text from the detected license plates. It converts the image into a two-color version (black and white). It then analyzes light and dark areas, where if the dark areas are letters then the light areas are identified as background and vice versa [2].

2 Literature Review

This section provides a revision of various topics like deep learning, object detection, character segmentation, character recognition that are relevant to ALPR. It also discusses several methods used frequently for ALPR.

2.1 Deep Learning

Convolutional Neural Networks and Deep Learning have recently revolutionized the field of computer vision. Many papers have used deep learning for ALPR [1, 3] and have yielded promising results. While Masood et al. [3] uses traditional CNNs for a proprietary software, Lee et al. [1] uses R-CNNs with a focus on detection of license plates only. Yosinski [4] provides much insight into using transfer learning to transfer pre-trained weights to make a deep learning model perform better.

Multi-oriented and scale invariant license plate detection technique is another approach based on convolution neural networks [5]. They do so by parameterizing three edge points and then mapping region proposals to infer the fourth point of the bounding parallelogram. They make use of anchor boxes to achieve scale invariance enabling them to recognize true images of multiple scales.

2.2 Object Detection

CNNs have been used for object detection which have remarkably improved the accuracy of detecting objects [6, 7]. One approach uses semantic region proposals for adaptive license plate detection [8]. They use semantic segmentation convolution network followed by a classification and regression network for region verification [9]. Redmon proposed YOLO (or You Only Look Once) which is a single shot object detection algorithm that significantly improved the precision and recall on several datasets [10]. Successive improvements to it were YOLOv2 [11] and finally YOLOv3 [12] which is currently considered state of the art in object detection.

2.3 Optical Character Recognition

In most papers, optical character recognition is done in multiple phases. It first starts with the pre-processing phase which includes various techniques like binarization, denoising, skew correction and thresholding. This phase is followed by the segmentation phase, normalization and feature extraction [13].

The tesseract OCR engine [14] is initiated by finding lines and words from the image. This is followed by fixed pitch detection and chopping of joined characters. It then associates broken characters to extract features and classify characters to easily recognize words.

3 Dataset

Several datasets are available for ALPR but most of them have license plates of different countries [15–17]. Considering that Indian license plates follow a unique set of syntax, and the relative complexity involved in detecting vehicles and license plates on Indian roads, we decided to generate our own dataset (Fig. 1).

Our dataset involves over 300 images of vehicles with Indian license plates. To diversify and improve the quality of our dataset we have included several types of license plates: black-over-yellow plates for tourist vehicles, black-over-white for private vehicles, white-over-red for temporarily registered vehicles, white-over-black for diplomatic plates and several other types.



Fig. 1 Images of vehicles with Indian license plates, after manual annotation. Examples of vehicles with front and rear license plates are shown, along with pictures captured from CCTV footage. Several types of license plates are also shown [18]

While limited number of night-time images are available due to reduced visibility, we have made sure to include images from different times of the day and under different weather conditions. Since ALPR from CCTV footage poses its separate set of challenges, we have also included car footage from CCTV feeds. Both parked vehicles as well as moving vehicles are included in our dataset.

All these images were manually annotated by us to generate the coordinates of the bounding boxes containing the license plates. The coordinates of the top-left and bottom-right edge of the bounding box were stored.

The dataset is split as: 80% for training and 20% for testing. The images are randomly shuffled before splitting to evenly represent each type of license plate in both the sets.

4 Implementation

We approach the problem in two stages. First stage includes transfer learning with our YOLO object detection algorithm and then fine tuning it on our dataset. The second stage includes improving the quality of the image, and then applying character recognition techniques to obtain our final output license plate in text form.

4.1 Proposed Model

See Fig. 2.

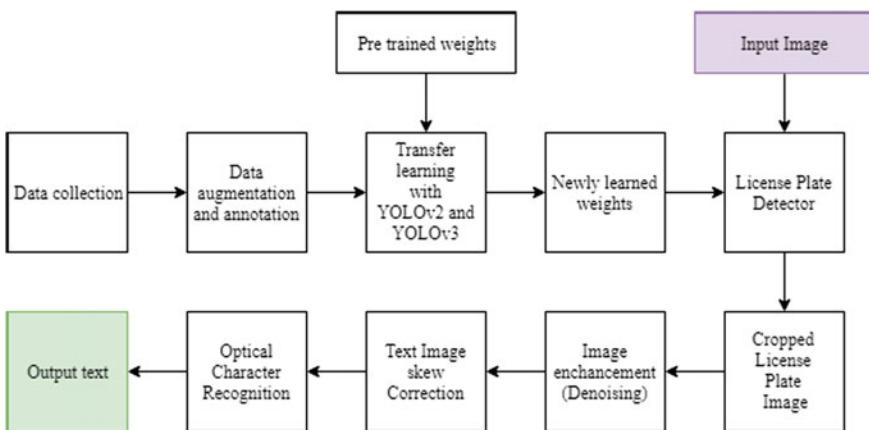


Fig. 2 Flowchart of proposed implementation model

4.2 Data Preprocessing

To augment the data, we first flipped all the images horizontally and then converted all images to grayscale. Not only does this increase the diversity of the data, but also prevents any kind of bias from affecting the training. Hence, we could now train our dataset on 1200 images instead of just 300, thus, significantly improving the quality of the data (Fig. 3).

Our annotated data gave us the location of the license plate in the form of pixels (x_1, y_1, x_2, y_2) where (x_1, y_1) is the location of the top-left edge and (x_2, y_2) is the location of the bottom-right edge of the bounding box. But to train the data using YOLO, we needed normalized points in the form (w, h, x, y) where w is the width of the bounding box, h is the height of the bounding box, and (x, y) is the location of the center point of the box. All these values are normalized. These are given as:

$$w = \frac{x_2 - x_1}{(\text{width of image})} \quad (1)$$

$$h = \frac{y_2 - y_1}{(\text{height of image})} \quad (2)$$

$$x = \frac{x_1 + x_2}{2x(\text{width of image})} \quad (3)$$

$$y = \frac{y_1 + y_2}{2x(\text{height of image})} \quad (4)$$

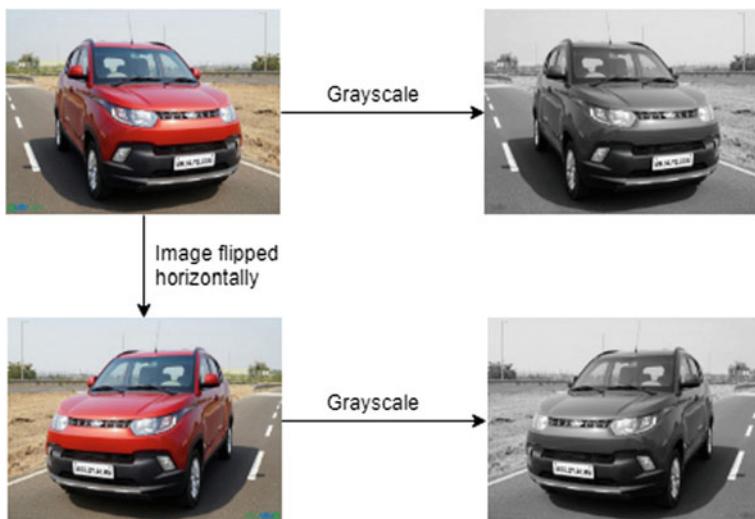


Fig. 3 Data augmentation of image using flipping and grayscaling

Finally, to generate the train and test files, we wrote a python script to randomly shuffle the images and split the data into training and test set, performing an 80/20 split.

4.3 Transfer Learning

Instead of training our own CNN from scratch, we chose to take advantage of the publicly available pre-trained weights of YOLOv2 and YOLOv3 [19]. These pre-trained weights are trained over thousands of images from ImageNet and hence provide substantial knowledge before the model has started training itself.

The base architecture of YOLOv2 that was used by us is DarkNet-19 and for YOLOv3 we used DarkNet-53. We modified this architecture to properly fit against our data.

As shown in Fig. 4a, the YOLOv2 architecture has 19 convolutional layers with different number of filters, with the last one having 30 filters. The number of filters in the last layer is calculated for YOLOv2 as:

$$\text{filters} = (\text{no. of classes} + 5) \times 5 \quad (5)$$

For YOLOv3, the architecture has 53 convolutional layers and the last convolutional layer has 18 filters. The number of filters in the last layer is calculated for YOLOv3 as:

(a)

Type	Filters	Size	Output
Convolutional	32	3x3/1	416 x 416
Maxpooling		2x2/2	208 x 208
Convolutional	64	3x3/2	208 x 208
Maxpooling		2x2/2	104 x 104
Convolutional	128	3x3/1	104 x 104
Convolutional	64	1x1/1	104 x 104
Convolutional	128	3x3/1	104 x 104
Maxpooling		2x2/2	52 x 52
Convolutional	256	3x3/1	52 x 52
Convolutional	128	1x1/1	52 x 52
Convolutional	256	3x3/1	52 x 52
Maxpooling		2x2/2	26 x 26
2x			
Convolutional	512	3x3/1	26 x 26
Convolutional	256	1x1/1	26 x 26
Convolutional	512	3x3/1	26 x 26
Maxpooling		2x2/2	13 x 13
2x			
Convolutional	1024	3x3/1	13 x 13
Convolutional	512	1x1/1	13 x 13
Convolutional	1024	3x3/1	13 x 13
1x			
Convolutional	30	1x1/1	13 x 13

(b)

Type	Filters	Size	Output
Convolutional	32	3x3/1	416 x 416
Convolutional	64	3x3/2	208 x 208
1x	Convolutional	32	1x1/1
1x	Convolutional	64	3x3/1
			208 x 208
Convolutional	128	3x3/2	104 x 104
2x	Convolutional	64	1x1/1
2x	Convolutional	128	3x3/1
			104 x 104
Convolutional	256	3x3/2	52 x 52
8x	Convolutional	128	1x1/1
8x	Convolutional	256	3x3/1
			52 x 52
Convolutional	512	3x3/2	26 x 26
8x	Convolutional	256	1x1/1
8x	Convolutional	512	3x3/1
			26 x 26
Convolutional	512	3x3/2	13 x 13
4x	Convolutional	1024	3x3/1
4x	Convolutional	512	1x1/1
4x	Convolutional	1024	3x3/1
			13 x 13
1x	Convolutional	18	1x1/1
1x	Residual		13 x 13

Fig. 4 Architectures used **a** YOLOv2 architecture **b** YOLOv3 architecture



Fig. 5 Image enhancement by denoising, followed by skew correction of image

$$\text{filters} = (\text{no. of classes} + 5) \times 3 \quad (6)$$

Since we have only one class, that is license plate, we get 30 filters for YOLOv2 and 18 filters for YOLOv3.

4.4 Image Enhancement

Image Enhancement is done to improve the accuracy of our optical character recognition system. We first remove noise from the image which is termed as denoising. We assume the noise to be Gaussian white noise, and then perform OpenCV's non-local means denoising algorithm. We then apply OpenCV's background subtraction technique to enhance our image quality.

4.5 Image Skew Correction

The license plate image that we obtain is not always straight and it needs to be corrected if we need better results with OCR. We need to determine both the direction (clockwise or anticlockwise) and angle of rotation, while correcting our image. We de-skew our input image by setting appropriate threshold values. After determining the skew angle, we apply an affine transformation to correct for the skew (Fig. 5).

4.6 Optical Character Recognition

The optical character recognition is done on our license plate images using the pytesseract package in python [20]. The pytesseract module, instead of providing python bindings, provides an interface to tesseract binary and writes the image to a temporary file on disk and then calls the tesseract binary on the file, and captures the resulting output.

5 Experimental Results

See Fig. 6.

5.1 License Plate Detection

We observed the changes in loss function as training progresses, as shown in Fig. 7. We noticed that in YOLOv3 the loss function becomes fairly stable after 200 iterations, while in YOLOv2, the loss function stabilizes after 300 iterations. The difference between the loss functions in YOLOv2 and YOLOv3 is that the last three terms in YOLOv2 are the squared errors, whereas in YOLOv3, they have been replaced by cross-entropy error terms. Object confidence and class predictions in YOLOv3 are predicted through logistic regression. Hence for YOLOv3 the loss function starts at a comparatively high value and quickly converges to 0.

We use Intersection over Union (IoU) as one of our evaluation metrics for our object detector on our dataset. It is the ratio of the area of overlap to the area of union

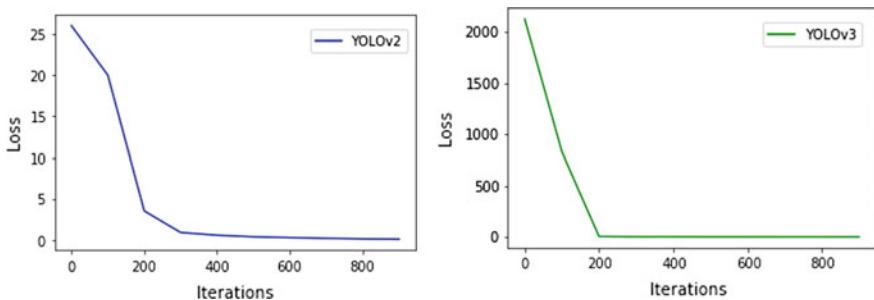


Fig. 6 Behavior of loss function across number of iterations for YOLOv2 and YOLOv3

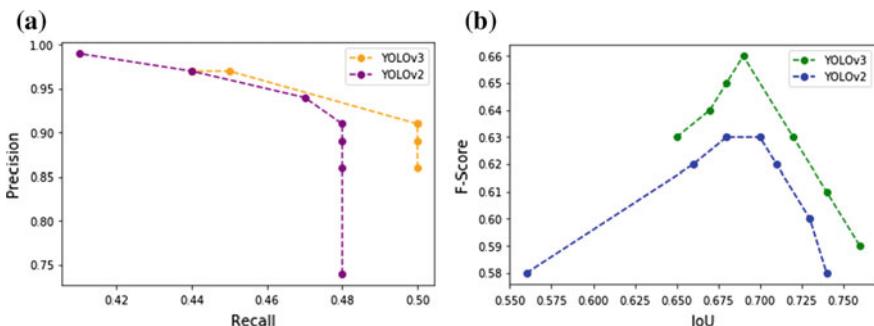


Fig. 7 Graphical Representation **a** Precision versus Recall Graph **b** F-Score versus IoU Graph

between the predicted bounding box and the ground truth bounding box. It is given as:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (7)$$

We see that YOLOv3 performs better than the YOLOv2 version in terms of precision and recall. The precision for YOLOv3 is greater than the precision for YOLOv2. We get a similar observation for our Intersection over Union measure (IoU) as we get a value of 0.58 for YOLOv3 and 0.57 for YOLOv2.

Another metric that we use is mean Average Precision (mAP) for our object detector. Average Precision computes the average precision value for recall values over 0–1. The mAP is the mean of the Average Precision (AP) over all classes.

$$\text{AP} = \int_0^1 p(r)dr \quad (8)$$

$$\text{mAP} = \sum_{q=1}^Q \frac{\text{AP}(q)}{Q} \quad (9)$$

Just like previous metrics, YOLOv3 yields a better mAP value of 53.30% than YOLOv2 that yields 51.28%.

5.2 Character Recognition

After analyzing our optical character recognition's performance, we obtained a fairly high accuracy in character recognition. As shown in Fig. 8, most of the characters are correctly labeled, with minor amount of incorrect predictions, mainly when two characters have similar visual representations (like the letter 'O' and the number '0').



Fig. 8 Performance of OCR on license plates detected by LPD

6 Conclusion

In this paper, we implemented transfer learning from two algorithms for license plate detection and we see that YOLOv3 performs better than YOLOv2 on our dataset. YOLOv3 generates a better IoU value at constant F-1 score and a better mAP value at a constant threshold as compared to YOLOv2. However, we found that the values of precision, recall, Intersection over Union, mean Average Precision and other evaluation metrics are not ideal because of the limited size of our dataset. We got good accuracy while doing optical character recognition, by using the tesseract OCR module with python.

As future work, we intend to collect more data and annotate them to increase the size of our dataset. We plan to explore the vehicle's registered information to see if we are able to collect data with relevant information. Additionally, we intend to further our scope to license plates of not only India, but other countries as well, specially covering those countries where such ALPR systems have not been implemented.

References

1. Lee D, Yoon S, Lee J, Park DS (2016) Real-time license plate detection based on faster R-CNN. *KIPS Trans Softw Data Eng* 5:511–520. <https://doi.org/10.3745/kttsde.2016.5.11.511>
2. Usmankhajaev S, Lee S, Kwon J (2020) Korean license plate recognition system using combined neural networks. In: Herrera F, Matsui K, Rodríguez-González S (eds) *Distributed computing and artificial intelligence, 16th international conference. DCAI 2019. Advances in intelligent systems and computing*, vol 1003. Springer, Cham
3. Masood SZ, Shu G, Dehghan A, Ortiz EG (2017) License plate detection and recognition using deeply learned convolutional neural networks
4. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? 3320–3328
5. Han J, Yao J, Zhao J, Liu Y (2019) Multi-oriented and scale-invariant license plate detection based on convolutional neural networks. *Sensors* 19:1175. <https://doi.org/10.3390/s19051175>
6. Girshick R (2015) Fast R-CNN. In: 2015 IEEE international conference on computer vision (ICCV). Santiago, pp 1440–1448
7. Ren S, He K, Girshick R Sun J (2017, June) Faster R-CNN: towards real-time object detection with region proposal networks. In: *IEEE transactions on pattern analysis and machine intelligence*, vol 39, no. 6. pp 1137–1149
8. Tian J, Wang G, Liu J (20, March 20) Semantic region proposals for adaptive license plate detection in open environment. *J Electron Imaging* 28(2):023017
9. Tian J, Wang G, Liu J (2019, March 9) Semantic region proposals for adaptive license plate detection in open environment. *J Electron Imaging* 28(2):023017
10. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, NV, pp 779–788
11. Redmon Joseph, Farhadi Ali (2016) YOLO9000: Better. Faster, Stronger
12. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement
13. Hamad KA, Kaya M (2016) A detailed analysis of optical character recognition technology. *Int J Appl Mathe Electron Comput* 4:244–244. <https://doi.org/10.18100/ijamec.270374>
14. Smith R (2007) An overview of the tesseract OCR engine. In: Ninth international conference on document analysis and recognition (ICDAR 2007), Parana, pp 629–633

15. Izidio DM, Ferreira AP, Medeiros HR, Barros EN (2018) An embedded automatic license plate recognition system using deep learning. In: 2018 VIII Brazilian symposium on computing systems engineering (SBESC). Salvador, Brazil, pp 38–45
16. Abdullah S, Hasan MM, Islam SM (2018) YOLO-based three-stage network for Bangla license plate recognition in Dhaka Metropolitan City. In: 2018 international conference on Bangla speech and language processing (ICBSLP). Sylhet, pp 1–6
17. Silva SM, Jung CR (2017) Real-Time Brazilian license plate detection and recognition using deep convolutional neural networks. <https://doi.org/10.1109/sibgrapi.2017.14>
18. <https://www.kaggle.com/dataturks/vehicle-number-plate-detection>
19. Sun H, Fu M, Abdussalam A, Huang Z, Sun S, Wang W (2019) License plate detection and recognition based on the YOLO detector and CRNN-12. In: Sun S (eds) Signal and information processing, networking and computers. ICSINC 2018. Lecture notes in electrical engineering, vol 494. Springer, Singapore
20. <https://pypi.org/project/pytesseract/>

Chapter 40

Efficacy Analysis of Technology Approaches Toward Auto-assignment of Clinical Codes to the US Patient Medical Record



Milind Godbole and Anuja Agarwal

1 Introduction

The hospitals generate a huge volume of data which may be the medical records of the patients, notes of the doctors, reports obtained from laboratory tests. The volume of this data is only increasing with time and with technological advancements the hospitals are opting to maintain these records electronically [1]. The electronic patient record is an abundant source of clinical data that could be utilized for a wide scope of computerized applications to enhance the process of health care like cautioning for potential therapeutic errors, generating the patient issues, and evaluating the seriousness of a condition [2].

The healthcare systems are employing a substantial number of frameworks for classification and categorization to aid the management of data for various tasks such as patient care, storage and retrieval of the records, billing, insurance, and statistical analysis. International Classification of Diseases (ICD) is one such official system for assigning the codes for diagnosis and processes linked to the hospital use. Once the clinical treatment of a patient is complete, the additional process involves assigning of diagnosis codes to the patient medical record. These codes are manually allocated by the trained and certified medical coder and these codes will be used to generate claim and as well as for any further diagnostic. The process is prone to errors and time-consuming due to the large set of health information that is available in the patient medical chart. The best example for this was the case of a Turkish hospital where the human experts on examining the 491 pre-labeled records of the patients found that more than 50% of the documents were assigned wrong codes of ICD [3].

M. Godbole (✉) · A. Agarwal

SVKM's Narsee Monjee Institute of Management Studies V. L, Pherozeshah Mehta Road, JVPPD Scheme, Vile Parle West, Mumbai, Maharashtra 400056, India

e-mail: milindgodbole@hotmail.com

URL: <http://www.nmims.edu/>

Due to the transition of storing the patient records from paper chart to electronic health record, the techniques of machine learning and NLP are immensely helping the assembly of the clinical data and driving the insights and meaningful use of the data. These techniques allow the generation of statistics, alerts, and trends; however, is normally applied and limited to the structured data elements of the patient's records.

However, in the area of risk adjustment coding, loads of the clinical data still remain in the form of semi-structured or unstructured data in the patient medical chart. Also, a lot of clinical data is documented in the form of free text [4]. Development of automated tools to auto-assign codes for the free-text patient records is an area that is being explored in recent times by both the academic researchers and healthcare practitioners. This study develops an approach for the automatic assignment of clinical codes to the semi-structured data available in -patient charts. This experimental study is conducted by adapting and combining the individual methods like OCR for text reading, statistical text mining, machine learning, and rule-based NLP algorithm. Traditional "bag-of-words" method used by earlier researchers was adopted to create classifiers as anchor keywords using statistical text mining and rule-based NLP algorithm is constructed.

2 Literature Review

Boytcheva [5] developed a technique for automatic mapping of ICD-10 codes for diagnosis derived from the discharge letters. Ailments that are frequently depicted in the medical records as free-ext utilizing words, paraphrases, and phrases contrast fundamentally from those utilized in ICD-10 classification. Along these lines, the task of recognizing the diseases (which for all intents and purposes implies allocating CMS institutionalized ICD codes to names of the diseases) is a significant challenge in NLP. This algorithm was particularly developed to process the text documents in the Bulgarian language. The technique is based on the multiclass SVM, wherein every ICD-104-character classification code is accounted to be a single class. The system provided an accuracy of 97.3% along with the recall and F-measure of 74.68% and 84.5%. Perotte et al. [6] studied the ICD9 codes and discharge summaries that were available in the MIMIC repository. They experimented the coding with two classifiers, viz. the flat classifier and hierarchy-based classifier. The flat classifier treats each code of ICD9 independent of each other while the other leverages the code's hierarchical nature during its modeling. The experimental results proved that the hierarchy-based classifier performed better than the flat classifier with an F-measure of 39.5% and 27.6%, respectively, when they were trained and tested for 20,533 and 2282 records. There is a requirement for benchmark datasets for research to progress and for the network to evaluate the estimation of various approaches dependably.

Scheurwegen et al. [7] investigated if integrating unstructured and structured data improved the prediction accuracy when compared to the use of these data types separately. In this technique, both structured and unstructured data are combined for

allocating the ICD9-CM clinical codes to patients stays. Two distinct data integration techniques were evaluated. The early data integration combined the features of various sources in a single model while the late data integration learned the separate model per source of data and combined the predictions with the aid of a meta-learner. These evaluations were conducted using the Dutch language EHR of a single hospital. The results proved that late data integration improved the performance algorithms across all the medical specialties. Kavuluru et al. [8] evaluated supervised learning method for automatic assignment of disease classification codes ICD-9-CM with the aid of a realistic EMR consisting of huge data. The main aim of this study was to determine the approaches which performed better when such a large volume of data is under consideration. They used a dataset of 71,463 EMRs that corresponded to the in-patient visits with a failed discharge date in the period from 2011 to 2012 obtained from UKY Medical Center. The experiments clarify that a bigger dataset improves small-scale scores over bigger code sets regardless of whether most codes have few precedents. In any case, better-named recognition of entity and increasingly precise methodologies for mark alignment are required to gain further ground in computerized code allotment particularly for those names with not many training models compared with the size of the whole dataset.

Koopman et al. [10] developed a framework for automatic identification and characterization of cancers from huge volumes of free-text death certificates. This aided the cancer registries in monitoring and reporting the rate of cancer mortality with reliable accuracy. The developed framework had two components: the NLP pipeline which extracted the features from the death certificates and SVM which use the extracted features to classify them. This model effectively determined if the cause of death was cancer with an F-measure of 0.942 and determined the various types of common cancers with an average F-measure of 0.7. The architecture of two levels first identified the presence of cancer and then the kind of cancer. Zhang et al. [11] addressed the issue of data imbalance in ICD-9-CM code automatic assignment task. They solved the data imbalance problem by strategically drawing data from PubMed to enrich the training data when required. An SVM classifier was used and the model was validated using the CMC dataset. The results of the evaluation indicated that the technique can significantly enhance the performance of the code assignment classifier at the macro-averaging level. However, for the classifiers which initially had enough data for training, additional data did not have any considerable effect.

Jackson et al. [9] used NLP to develop a suite of language model that could capture the critical symptoms of severe mental illness (SMI) from the clinical text. This technique was tested for the 23,128 discharge summaries obtained from the CRIS database. It could be determined from the results that 46 symptoms were extracted with an F1 score of 88%. The symptom models that did not perform well were excluded. From the discharge summaries, 87% and 60% of the patients were diagnosed with and without SMI, respectively. This framework demonstrated the possibility of automatic extraction of a wide scope of symptoms for English text. Descriptive information additionally demonstrated that most indications cut across

diagnoses, instead of being confined to specific gatherings. Baumel et al. [12] investigated four models to allocate the ICD codes to discharge summaries and validate using the MIMIC II and III clinical datasets. They developed a Hierarchical Attention-bidirectional Gated Recurrent Unit (HA-GRU) for tagging the document by the identification of the sentences relevant for every label. HA-GRU accomplished state-of-the-art results, and along with its best performance, it provided an insight into the future extensions of this work. The HA-GRU accomplished 7% improvement in absolute F1, implying that it needs lesser data for training to accomplish high performance.

Healthcare information includes a variety of data on medicine, clinical, admin and staff measured readings. Machine will later store such data in the systems for further analysis, which was previously stored in outdated platforms. Due to inadequate integration within heterogenous platforms used in healthcare systems results into poor healthcare management and thereby impacting the IT transformation in healthcare sector.

The gap in this proposal is the fragmented healthcare space, which is still using the traditional methods of storing and managing highly complex data. Usage of conventional process management technology is not practicable. The automation of the processes through business process modeling in health care will find its solution through predictive data analytics and techniques resulting in the digitization in the industry. The journals cited and referred in this proposal stated the need for new data and modeling techniques for the evolution of management. This will further facilitate the practices seamlessly. The current practice continues with the loss of data and high cost as it remains robust. The literature related with examining the inner and outer context affecting hospitals' financial performance was also reviewed. Revenue cycle management details were broken down into certain distinct categories. Literature does not focus directly on the issues related with RCM or healthcare information process (notable exceptions include Rauscher and Wheeler).

Big data is an accumulation of datasets which are abundant and intricate in character. They comprise both structured and unstructured data that evolve abundant, so speedy they are not convenient by classical relational database systems or current analytical tools. Medical staff including the practitioners and healthcare service providers will benefit with the proposed techniques by reducing their time and increasing the accuracy. This states to be critical information for the patients.

The technique will also facilitate in:

- Retrieving past visits and data information through systems;
- Collecting and storing data through a channeled medium as manual data collection and analysis are not feasible or sustainable;
- Business process modeling will focus on identifying hidden patterns more effortlessly, as traditional practices is time-consuming, challenging, inefficient, and costly;
- Smooth decision-making process as identification and extraction of contextual data are critical in providing the meaningful information needed to raise awareness and empower informed decision making.

However, all the departments work in a systematic format and have different needs. Considering this, the proposed technique will help in a modern way of decision making and functioning of the environments dealing with predictive analytics, reporting, and clinical information.

3 Proposed Methodology

The primary challenge in discovering the knowledge is the handling of structured and unstructured data which is available in the medical records of the patients. The techniques of data analytics include conventional statistics, machine learning, and text mining. There are various techniques for automatically coding and classifying for converting clinical information into structured format. The model developed in this study is illustrated in Fig. 1; it is a combination of three techniques statistical text mining, machine learning, and NLP which is developed to form computer-assisted coding (CAC) technique.

The structural text mining applied in this study has three phases, viz. putting some structure to the unstructured text, pattern identification, clinical text assessment, disease interpretation and code assignment. After the extraction of important concepts, machine learning algorithms like decision tree induction and association rule mining discover the rules of classification for targeted text words. Utilizing these anchor keywords for the disease categories and the traditional “bag-of-words” method as the classifiers with acceptable performance and rules are created. The application of conjunctive STM and NLP techniques are combined to implement as a data reduction technique as increased levels of data and multiple combinations will introduce machine doing loads of extra coding without taking into consideration of CMS guidelines

The primary objective of text mining is sifting through huge volumes of text for extraction of models and patterns so that they can be adopted in the automatic computing application. The decision rules and techniques based on decision trees are particularly attractive from learning perspective, since they provide detailed insights to the developers and the end users. The study has focused on applying these techniques categorizing the text and increasing the performance quality of the machines. The “Decision Tree” technique has been used in the study for statistical text mining. The patient data for a period of one year which has been manually processed by human intelligence is considered for benchmarking. From these processed patient records, the unique ICD codes are collected to determine the occurrence of each code. Simple Pareto chart technique was then used for listing the frequently appearing diseases and bag of words. To test the algorithm, top 20 ICD codes carrying maximum weightage was considered to identify the respective “anchor word” for each ICD code along with the disease family. Then, the “bag of word” for each “anchor word” is identified and with this combination several set of rules are formed. After the rule creation, they are run on a sample of 6 sets consisting of 500 charts each. The obtained results are evaluated, and root cause analysis (RCA) is conducted

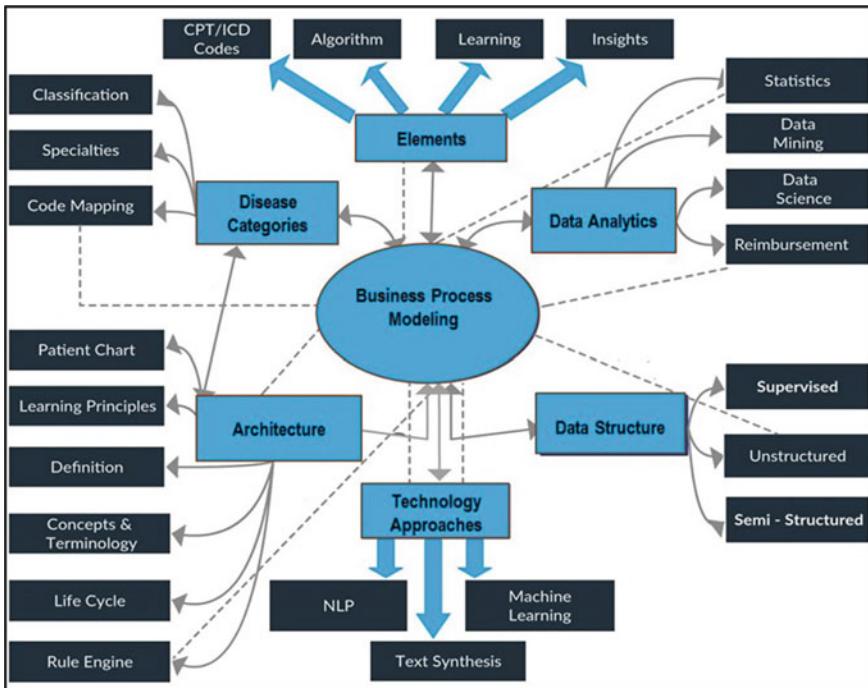


Fig. 1 Depicts the elements involved in this experimental study and their inter-dependencies as regards to variables and classifications

by medical coding subject matter experts, and based on this analysis, the existing rules are rectified along with the addition of new ICD code rules. This process is conducted for several iterations, and for each iteration, base lining of the rule-based NLP framework is done. Each of the baseline version accuracies is measured with the manually processed data to determine the accuracy benchmark (Fig. 1).

In the rule-based NLP, the training data is split into two sets of growing set and pruning set, the former aids in learning the rules while the latter helps in the removal of overfitting rules. Every rule will start empty conjunction to which new condition will be added based on the criteria of information gained. When all the positive samples are covered, then the rule will be tested on the pruning set in order to remove the conditions of over fitting. Depending on the error obtained from the rules applied to the test set in the descriptive rule set algorithm, it will be determined whether pruned rule set can be included, continued or stopped. Finally, the post-processing step will be performed on the basis of heuristic optimization of the rule set.

The approach of machine learning uses a cascade of two classifiers trained on the same data for predicting the codes. Both these classifiers will perform the multi-label classification by decomposition of the task into a fixed set of binary classification problems one for every code. Hence, the classifier can predict the impossible or empty combination of the codes. These kinds of recognizable mistakes will be utilized for

triggering the cascades, i.e., when the first classifier makes an error known to kit, then the second classifier output is the final prediction.

The flowchart in Fig. 2 illustrates the flow of the developed technique. The OCR process is carried out on the patient charts sent by the client to search for keywords, bag of words, negative words, encounter date, etc. Then the searched data is extracted, and the rule NLPengine is applied to this data. If the match is not found, then diagnosis codes are not applied for that word and the process moves on to the next word to find a match. If the match is found, then the respective diagnosis code is applied, and they are auto-populated on the coder screen.

Patient Chart Sample:

1. Patient chart text information

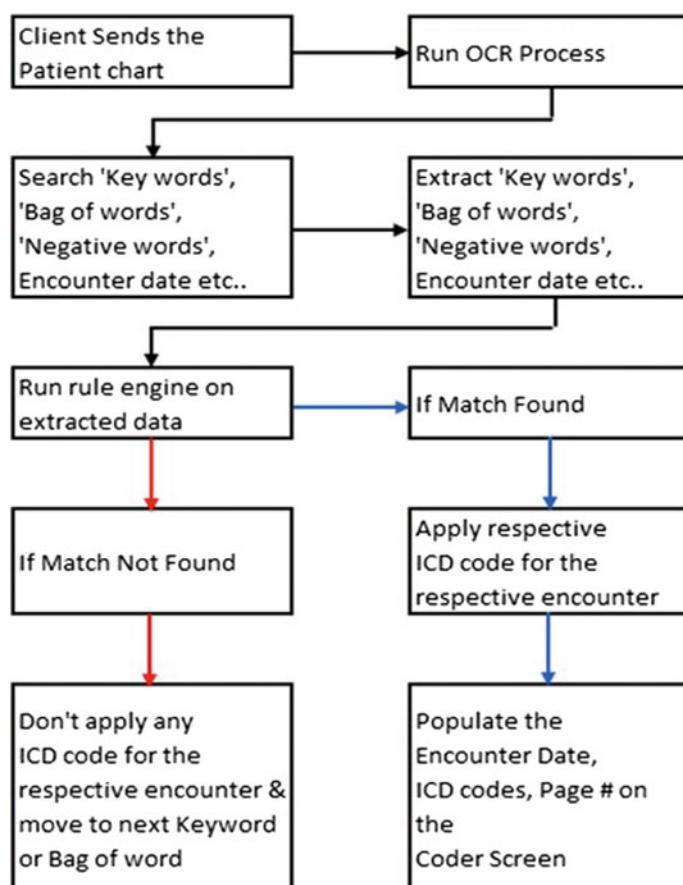


Fig. 2 Flow of the proposed model

<i>Discharge Summary</i>	
<p>during this hospitalization.</p> <p>4. Incidentally found to have a right-sided hydronephrosis with dysuria and possible urinary retention. Seen by Dr. Williams in consultation during this hospitalization. Followup advised. The patient's symptoms have improved and the right-sided hydronephrosis is resolved by CT urogram. At this time the patient and wife understand this might be a potential source for infection if continued urinary retention recurs.</p> <p>5. Rheumatoid arthritis by history. Resume outpatient therapy.</p> <p>6. Hypertension. Resume outpatient therapy.</p> <p>7. Gastroesophageal reflux disease, on therapy.</p> <p>8. Obstructive sleep apnea. Stable during this hospitalization.</p> <p>9. Hypothyroidism. Resume outpatient therapy.</p> <p>10. Leukemia underlying dementia although at the present time the patient has remained stable with family next of kin support.</p> <p>11. Antibiotic therapy, as stated above to complete the full course until 03/17/2017.</p>	

2. “Keywords,” “bag-of-words” sample:

Diagnosis code	Anchor keyword	Bag of words (1)	Bag of words (2)	Bag of words (3)
E11.69	AODM	AODM with other	Arteriosclerosis	Coronary
I13.0	Hypertension	Benign essential hypertension	Cardiac Asthenia	Chronic kidney disease, unspecified CKD stage
C61	Prostate	Prostate	Cancer	
E03.4	Thyroid	Acquired	Thyroid	Atrophy
E89.0	Hypothyroidism	Post-irradiation	Hypothyroidism	
E05.90	Myxedema	Circumscribed	Myxedema	
E00.1	Hypothyroid	Cretin	Hypothyroid	
I13.10	HBP	HBP	Myocardium degeneration	CKD
I13.2	High blood pressure	High blood pressure	Chronic Kidney Disease 5	Myocardial insufficiency
C61	Prostate	Prostate	Adencarcinoid	

Challenges faced during running the proposed model:

- Unstructured formats of patient charts
- Poor quality of PDF files
- Hand-written data
- No clear indications of encounter date in the charts
- Different formats of “Dates,” without header
- Read data from different patterns like tables and columns (tabular information and columnar information)
- Reading sequence of data
- Negative phrases and exclusions
- Identification of codable sections

- Client-specific coding guidelines.

Algorithm for Set 1:

- Date of service (DOS) scoping: Identification of the DOS basis date header/keyword by the machine, so that the scope at the encounter level is defined. This date identification is extremely critical as patient got serviced on that particular day and billing claim will have this date of service captured.
- After the DOS scoping, the machine identifies the codable section, i.e., to look for the H&P notes, progress notes, encounter notes, and consultation notes. The entire information is available on the patient medical chart, but we need to identify the areas from which the text needs to be searched, identified, compared, and then go through NLP algorithm. If the codable section is not identified, then it will result into machine coding from any unstructured text which may or may not be useful as per CMS guidelines and will result into extra codes.
- Checking for anchor keywords and marking them.
- Listing out the distinct “bag of words” found against anchor keywords.
- Checking for negative words and marking them as removed, that presides any bag of the keyword.
- If a negative word is found in either anchor keyword or respective “bag of word,” then the respective rule is ignored, else the rule will be run.
- Then “BMI” anchor keyword-related rules are checked.
- If the Family Dx is checked and it is coded only once even if it is found many times within the scope of the respective DOS (Encounter).
- The data is stored against the respective chart.

Algorithm for Set 2:

- DOS scoping: Identification of the DOS basis date header/keyword by the machine, so that the scope at the encounter level is defined.
- After the DOS scoping, the machine identifies the codable section, i.e., to look for the H&P notes, progress notes, encounter notes, and consultation notes.
- Checking for provider credentials (e.g., acceptable: ANPC, APN, MD, and MBBS; non-acceptable: OA, RN, pharma, and registered nurse) from the codable section.
- Checking for distinct anchor keywords and marking them.
- Checking for negative words that preside any bag of the keyword and marking them.
- Listing out the distinct “bag of words” found against anchor keywords.
- Checking for negative words that preside any bag of the keyword and marking them as removed.
- If a negative word is found in either anchor keyword or respective “bag of word,” then the respective rule is ignored, else the rule will be run.
- Then “BMI” anchor keyword-related rules are checked.
- If the Family Dx is checked and it is coded only once if it is found many times within the scope of the respective DOS (Encounter).
- The data is stored against the respective chart.

Algorithm for Set 3:

- DOS scoping: Identification of the DOS basis date header/keyword by the machine, so that the scope at the encounter level is defined.
- After the DOS scoping, the machine identifies the codable section, i.e., to look for the H&P notes, progress notes, encounter notes, and consultation notes.
- Checking for provider credentials (e.g., acceptable: ANPC, APN, MD, and MBBS; non-acceptable: OA, RN, pharma, and registered nurse) from the codable section.
- Checking for distinct anchor keywords and marking them.
- Checking for negative words that preside any bag of the keyword and marking them.
- Listing out all of the distinct “bag of words” found against anchor keywords.
- Checking for negative words that preside any bag of the keyword and marking them as removed.
- If a negative word is found in either anchor keyword or respective “bag of word,” then the respective rule is ignored, else the rule will be run.
- Checking the bag of words in any sequence for the respective anchor keyword.
- Then “BMI” anchor keyword-related rules are checked.
- If the Family Dx is checked and it is coded only once if it is found many times within the scope of the respective DOS (Encounter).
- The data is stored against the respective chart.

4 Results and Discussion

The steps mentioned below are followed for obtaining the results.

- The data considered for the study by the coder are already coded charts (PDF files) and data of the year 2018.
- A separate CAC database instance is created, and it is loaded with the already coded charts and data.
- From the charts, the “Readable” and “Non-readable” are identified to create 4 sets of 500 charts each.
- Then the “anchor keywords” and “bag of words” are identified, and the algorithm is created for executing on the charts.
- The OCR is run on the identified charts and stored in the OCR output within the database against every chart.
- The auto-coding algorithm (CAC) is run on each set.
- The CAC algorithm is baselined after every 4 sets run.
- Next 4 sets (5, 6, 7, and 8) of 250 charts each are identified and created.
- The baselined CAC algorithm is run and the above steps 6–8 are followed.

Below mentioned are the results and the regression analysis obtained for the three sets of data. It also specifies the changes that are recommended for each set of data.

Set 1 Summary:

Run date	Iteration	Set	Template	Sample size	Applied rules
28-Dec-18	1	Set 1	Patient Charts	185	3333

Set 1 Analysis:

Coder coded	Machine coded—matched	Machine coded—extra	Machine not coded	Current accuracy percentage	Machine extra coding %	Machine not coded %
8243	480	8115	4055	4%	12%	49%

The set 1 results are analyzed using Poisson regression analysis to understand the relationship between the set of predictors and response that describes the number of times the event occurred in a defined observation space. The regression analysis establishes the relationship between set of predictors (match codes, system extra coded codes, and system not coded codes) with a response. From the analysis, it can be inferred that the accuracy of the machine (Dx codes matched) is affected by the system extra coded and the system not coded. Graph in Fig. 3 shows the impact percentage of matched found, system extra coded, and the system not coded on the Dx codes.

The relationship between the total charts coded and the match found, system not coded, and system extra coded codes was verified using the scatter plot test, the results of which can be observed in the scatter plot of Fig. 4. The inference drawn from the results are:

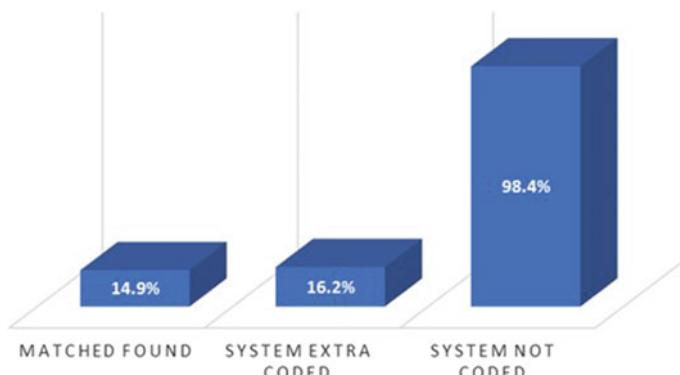


Fig. 3 Regression analysis of set 1 results

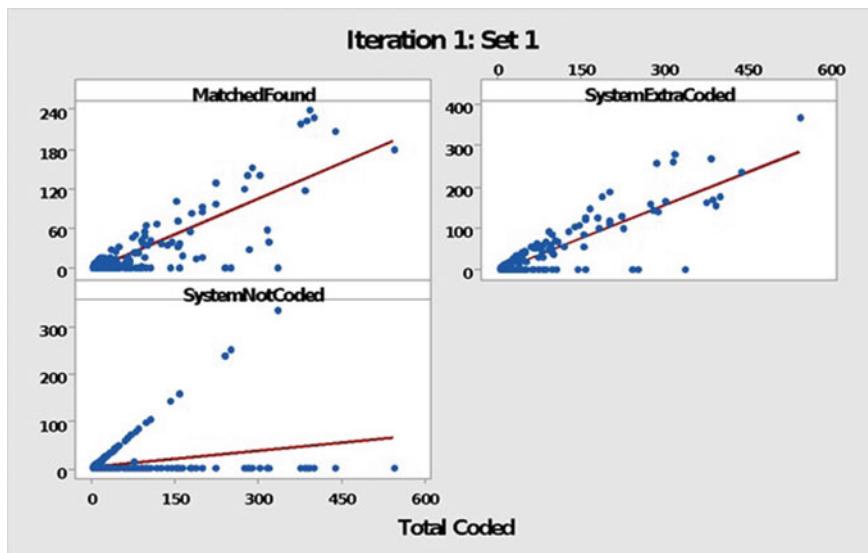


Fig. 4 Results of the scatter plot test on set 1

- Match found: A positive co-relationship can be observed indicating that the machine accuracy will increase with the coding of a higher number of charts. This implies that the applied rules were successful.
- System extra coded: A positive co-relationship can be observed indicating that all of the rules need to be relooked at since the machine is coding extra hence there is a requirement to take corrective measures.
- System not coded: A weak positive co-relationship can be observed indicating that there are some rules which were unsuccessful as a result of which the machine was unable to code the charts appropriately, implying the need to check the rules applied in machine learning.

The pie chart obtained for set 1 shown in Fig. 5 indicates that there is a need to revise the rules applied on set 1 of Iteration 1, since the success percentage of the rules is 23% (match found). From the pie chart, 30% of the rule was coded extra, thus Pareto analysis was carried out to find the reasons for extra coding (Fig. 6).

The rules must be revalidated based on the reasons mentioned below which were identified by Pareto analysis.

- Unacceptable Provider Credential
- Unacceptable headers
- Family DX need to update
- Negative keywords
- 2 Progress note for single DOS
- Need to update rules.

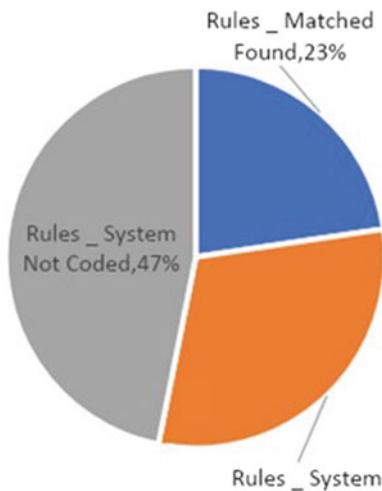


Fig. 5 Pie chart obtained for set 1

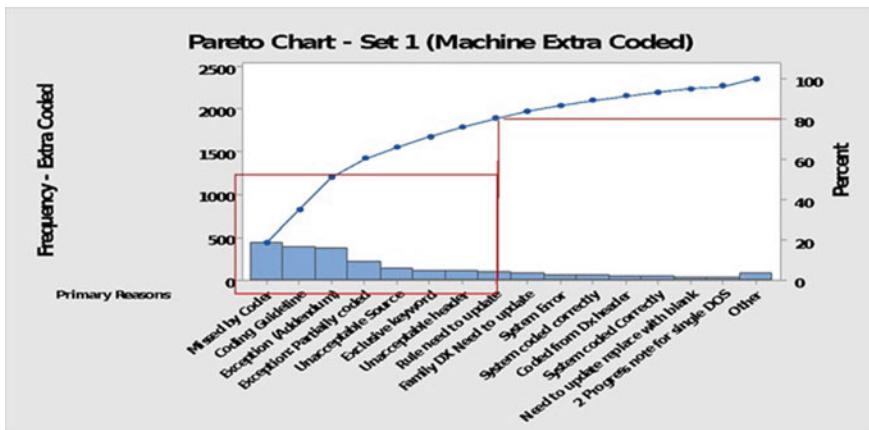


Fig. 6 Pareto chart obtained for set 1

Once the top contributors of machine extra coded were identified, 84 new rules were added to increase the accuracy of matched codes increases. The summary and analysis of this are presented below.

Set 2 Summary:

Run date	Iteration	Set	Template	Sample size	Applied rules
31-Jan-19	1	Set 2	WellCare Charts	500	3417

Set 2 Analysis:

Coder coded	Machine coded—matched	Machine coded—extra	Machine not coded	Current accuracy percentage	Machine extra coding %	Machine not coded %
9438	2364	2857	7074	25%	23%	75%

Figure 7 above compares the machine extra coded of set 1 and set 2, from which it can be observed that the “Incorrect Dos Captured by System” has increased from 3 to 35%, thus there is a requirement to check “bag of words” to address the machine extra coded codes.

The changes recommended based on the above analysis of the pie charts are given below:

- Need to update rules: Check the bag of words in any sequence within DOS scope

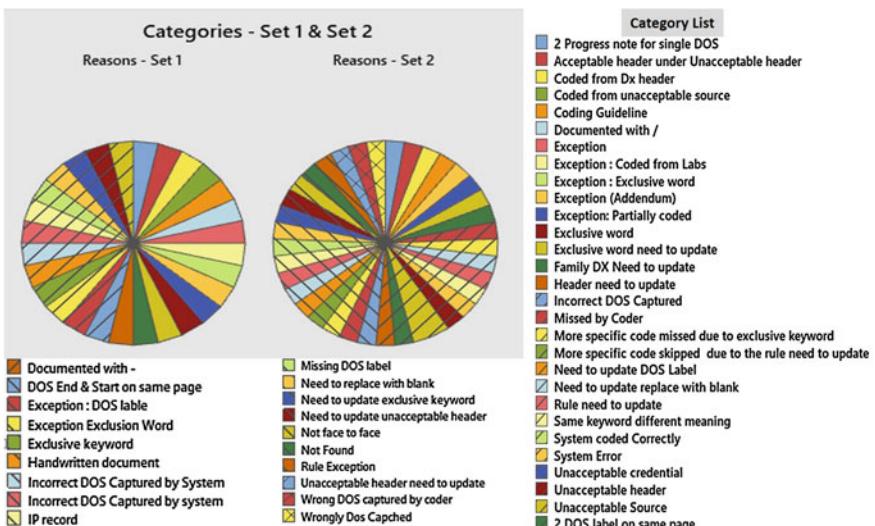


Fig. 7 Pie charts comparing the performance of (machine extra coded) set 1 and set 2

Set 3 Summary:

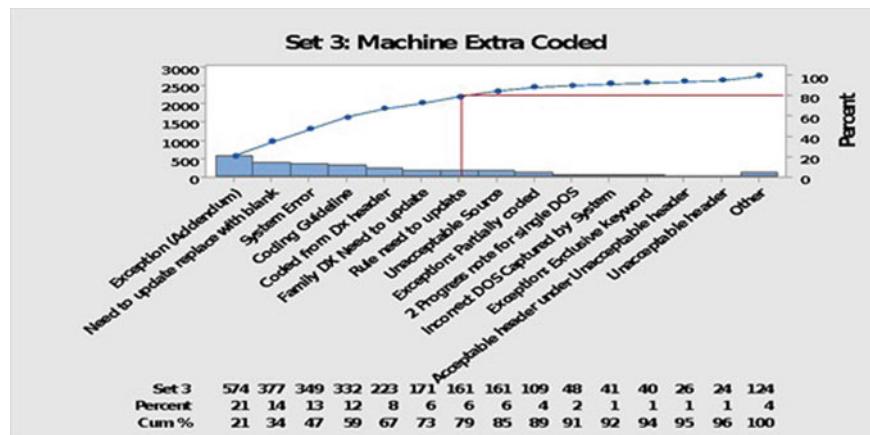
Run date	Iteration	Set	Template	Sample size	Applied rules
26-Feb-19	1	Set 3	WellCare Charts	500	4250

Set 3 Analysis:

Coder coded	Machine coded—matched	Machine coded—extra	Machine not coded	Current accuracy percentage	Machine extra coding %	Machine not coded %
12,698	4901	5712	7797	39%	31%	61%

Based on the outcome analysis of the set 3, the changes recommended are given below:

- Unacceptable provider credential—Add more credentials
- Unacceptable headers—Add more unacceptable headers
- Family DX need to update—More Family codes to be updated
- Negative keywords—Include more negative key words
- Need to update rules—Few rules to be updated and new rules to be added.



The Pareto analysis identified 33 reasons for machine extra coded among which 8 reasons, i.e., 24% of these reasons contributed to 85% of the overall reason.

Table 1 Summary of iterations

	Iteration 1				CAC Accuracy %	Iteration 1			
	Set 1	Set 2	Set 3	Set 4		Set 1	Set 2	Set 3	Set 4
Unique Dx Codes	309	346	396	404	CAC Accuracy %	14.90%	15.30%	14.60%	33.40%

	Iteration 2						CAC Accuracy %	Iteration 2					
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6		Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Unique Dx Codes	349	396	436	413	313	337	CAC Accuracy %	29.80%	49.60%	37.10%	47.50%	50.00%	58.00%

	Iteration 3						CAC Accuracy %	Iteration 3					
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6		Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Unique Dx Codes	392	472	495	521	441	387	CAC Accuracy %	55.60%	56.00%	58.00%	60.00%	62.00%	65.50%

These 8 reasons are mentioned below:

- Exception (Addendum)
- Need to update replace with blank
- System error
- Coding guideline
- Coded from Dx header
- Family DX needs update
- Unacceptable source
- Rule needs update.

The machine algorithm did not address the above reasons hence increasing the rework, i.e., the manual deletion of the extra coded codes. The summary of the three iterations is tabulated in Table 1.

The graph in Fig. 8 plots the accuracy of match found for all the iterations in Table 1. From the graph, it can be observed that the accuracy has increased from 14.9 to 65.5%.

5 Conclusion

In this study, we developed an effective text analytic process by adapting and combining individual techniques. The text is preprocessed using NLP and STM. Combining the STM and NLP helps in reducing the data. Then, the traditional “bag of words” method is used for creating the classifiers with acceptable performance and the rules are constructed using this model. The method was assessed on a set of real clinical

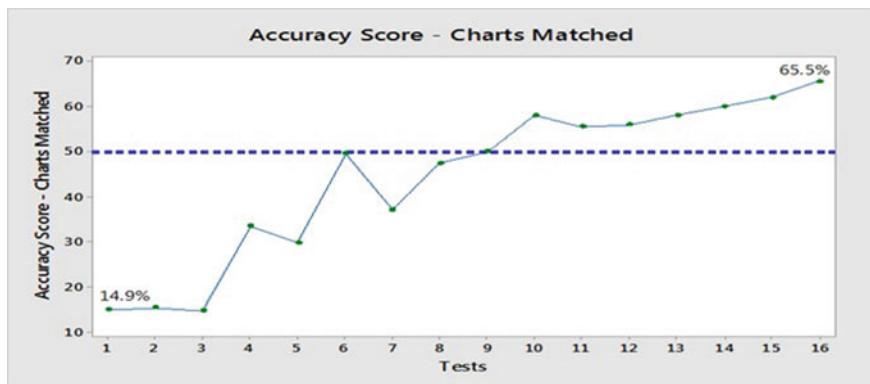


Fig. 8 Accuracy of matched found

notes annotated by the experts for the year 2018. From the results, it can be observed that the accuracy of matched found increased from 14.9 to 48.8% implying a positive trend for the rules developed for the machine auto-coding.

References

- Crammer K, Dredze M, Ganchev K, Talukdar PP, Carroll S (2007, June) Automatic code assignment to medical text. In: Proceedings of the workshop on bionlp 2007: biological, translational, and clinical language processing. Association for Computational Linguistics, pp 129–136
- Friedman C, Shagina L, Lussier Y, Hripcsak G (2004) Automated encoding of clinical documents based on natural language processing. *J Am Med Inform Assoc* 11(5):392–402
- Arifoğlu D, Deniz O, Aleçakır K, Yöndem M (2014) CodeMagic: semi-automatic assignment of ICD-10-AM codes to patient records. In: Information Sciences and Systems 2014. Springer, Cham, pp 259–268
- Suominen H, Ginter F, Pyysalo S, Airola A, Pahikkala T, Salanterä S, Salakoski T (2008, July) Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: a method description. In: Proceedings of the ICML/UAI/COLT workshop on machine learning for health-care applications
- Boytcheva S (2011) Automatic matching of ICD-10 codes to diagnoses in discharge letters. In: Proceedings of the second workshop on biomedical natural language processing. pp 11–18
- Perotte A, Pivovarov R, Natarajan K, Weiskopf N, Wood F, Elhadad N (2013) Diagnosis code assignment: models and evaluation metrics. *J Am Med Inform Assoc* 21(2):231–237
- Scheurwegs E, Luyckx K, Luyten L, Daelemans W, Van den Bulcke T (2015) Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *J Am Med Inform Assoc* 23(e1):e11–e19
- Kavuluru R, Rios A, Lu Y (2015) An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artif Intell Med* 65(2):155–166
- Jackson RG, Patel R, Jayatilleke N, Koliakou A, Ball M, Gorrell G, Roberts A, Dobson RJ, Stewart R (2017) Natural language processing to extract symptoms of severe mental illness from clinical text: the clinical record interactive search comprehensive data extraction (CRIS-CODE) project. *BMJ Open* 7(1):e012012

10. Koopman B, Zuccon G, Nguyen A, Bergheim A, Grayson N (2015) Automatic ICD-10 classification of cancers from free-text death certificates. *Int J Med Informatics* 84(11):956–965
11. Zhang D, He D, Zhao S, Li L (2017, August) Enhancing Automatic ICD-9-CM code assignment for medical texts with PubMed. In: BioNLP 2017. pp 263–271
12. Baumel T, Nassour-Kassis J, Cohen R, Elhadad M, Elhadad N (2018, June) Multi-label classification of patient notes: a case study on ICD code assignment. In: Workshops at the thirty-second AAAI conference on artificial intelligence

Chapter 41

Survey of Sentiment Analysis on Social Media



Suyash Chavan, Jai Puro, Sarthak Kawade and Pramod Bide

1 Introduction

Sentiment analysis is the computational study of people's sentiment, emotions, attitude and feeling towards a particular entity, the entity being an individual, events, topics. The main target of sentimental analysis is to detect the expressed sentiments from the text form and then classify their polarity. Sentiment analysis finds its application as an important market research tool, employed across numerous domains. The most common application is to identify the feedback generated from customers through the form of reviews. Often feedback exists in a pure textual format, such as reviews [1, 6, 5], which needs to be evaluated. Sentiment analysis also plays its role in the identification of social trends [4, 10], especially in trying to identify the socio-political mindsets of the masses [15]. Sentiment analysis also finds its use in usually non-conventional setups such as stock market evaluation/predictions [7] and in the education domain to identify student psyche [8].

The main source of data is either reviews or customer feedback of personal posts on social media. Such data is very crucial for business holders as they can take necessary and important decisions based on the reaction of consumers to their products

S. Chavan (✉) · J. Puro · S. Kawade · P. Bide

Department of Computer Engineering, Bharatiya Vidya Bhavan's Sardar Patel Institute of Technology, Mumbai, Maharashtra, India

e-mail: suyash.chavan@spit.ac.in

J. Puro

e-mail: jai.puro@spit.ac.in

S. Kawade

e-mail: sarthak.kawade@spit.ac.in

P. Bide

e-mail: pramod_bide@spit.ac.in

and services. The best source for data gathering is social media and microblogging Websites as people generally express their views freely.

Many sentiment analysis methods were proposed in the past years. This survey aims to identify the various enhancements done and to summarize some papers published in the field of sentiment analysis. The authors have collected around 15 articles trying to cover the major methods such as support vector machine [3, 8] and Naive Bayes model [14, 9] which are used for sentiment analysis and categorize them on the basis of algorithm or technique used to do the sentiment analysis. The sentiment analysis classification techniques shown in Fig. 1 are discussed in detail illustrating related articles and originating references as well.

Many papers are published in the sentiment analysis field, and hence, a need of summarizing all the common techniques has emerged which will highlight the trend and changes in this growing field. This survey will be useful as it will discuss almost all the most used techniques for sentiment analysis. So, newcomers to this field can get a broader idea about the enhancements and techniques of sentiment analysis by reading one survey paper. This will help them in their research work in this field as well.

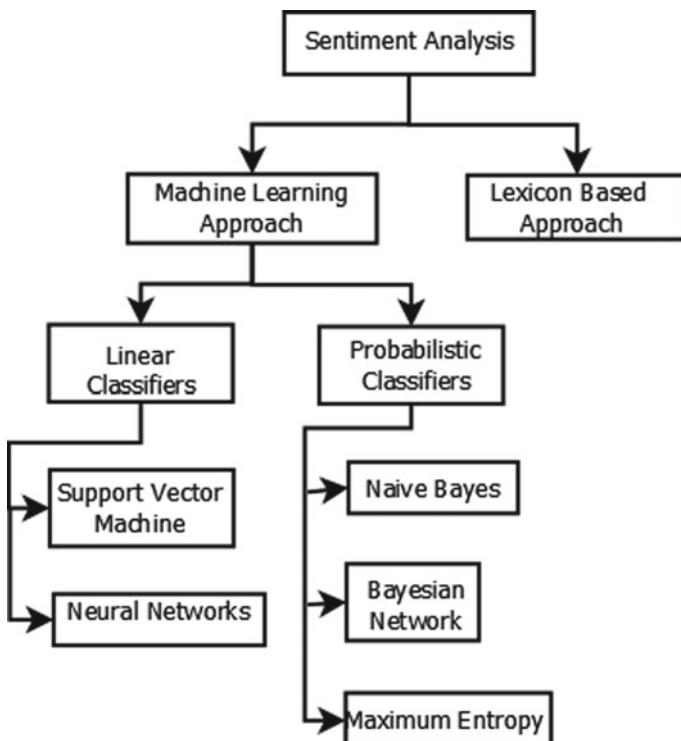


Fig. 1 Classification of sentiment analysis

2 Literature Survey

A lexicon-based approach uses a dictionary having words along with their sentiment score, which denotes how positive or negative the word is. These scores can be used directly to provide sentiments based on sentence structures. Lexicon-based approaches can be further enhanced by considering sentence structures, such as enhancers, which are words which magnify the sentiment of subjective words (“very” in “very beautiful”). There are many established lexicon-based resources, which can be used for sentiment analysis, like R’s sentiment package. A dictionary is used to map words to their subjectivity as well as polarity scores. The dictionary was compiled using sentences from social media and internally uses a Naïve Bayes like approach at calculating the sentiment. Sentiment scores are assigned by at a word level, and then, the total positive sentiment score and total negative sentiment score are obtained. These scores can then be used to extract sentiment score [1]. SentiWordNet is an important lexical resource available which can be used for sentiment analysis. It uses an internal dictionary where each word is tagged to three scores: objectivity, positive and negative. The net sentiment value using positivity score and negativity score was calculated. A weighted sum of the net sentiment value was taken as final sentiment score for the topic [7].

Lexicon uses a hybrid method for sentiment analysis of words. The lexicon-based approach is used to label the words, and then, this dictionary is used as the training set for a machine learning model. The paper does extensive preprocessing on the data. Some of the unique preprocessing technique includes conversion of emoticons and recognition of idioms as a single token. Trinh et al. [13] use a lexicon-based approach. Here, a new dictionary specific to Vietnamese language called Vietnamese emotional dictionary or VED based on the SO-CAL dictionary is created. Based on this dictionary, the emotional score for a sentence is calculated by calculating the emotional scores for adjectives, adverbs, nouns and verbs separately and then summing them. The obtained feature vector is then plugged into SVN.

Naive Bayes is one of the most popular methodologies used for processing of natural language. It classifies the text into positive or negative sentiments. The Naive Bayes classifier works on the principle of conditional probability, that is, it calculates the probability of a text belonging to a certain category (positive or negative) given the frequency of the positive or negative words occurring in that text.

Troussas et al. [14] employ a Naive Bayes classifier to determine if a certain status update is positive or negative. The basic formula is

$$P(\text{Sentiment}|\text{Sentence}) = \frac{P(\text{Sentence}|\text{Sentiment})P(\text{Sentiment})}{P(\text{Sentence})}$$

The methodology further uses Laplace smoothing to avoid 0 s in multiplications. The probability of a given status being positive/negative before it has been evaluated is always 0.5. The incoming text is tokenized, and each token the likelihood of each

word being seen in that class is multiplied together. The final result is sorted, and the text is classified into the highest scoring class.

Parveen and Pandey [9] again make use of the Naive Bayes approach. The tweets are collected using the Twitter API. The data is pre-processed by removing the hyperlinks, special symbols and usernames and by converting the emoticons to a word that best matches the sentiment conveyed by it. Then, it goes through a map phase that creates a hashmap containing the polarity of each word and then processes the overall polarity of the tweets by applying Naive Bayes algorithm. Then, it goes through a reduce phase that retrieves the polarity of the overall tweet and classifies it accordingly. The SentiWordNet dictionary is used with Hadoop for this method.

In [2], the data is obtained with the help of Twitter API, and the tweets that were in other languages were translated to English. An ensemble of Naive Bayes and SVM is used to analyse the sentiments of the tweets. The log-ratio vector calculated by the Naive Bayes algorithm with the help of the average word extracted from the positive and negative documents. This multiplied by the binary pattern of each word is supplied as the input to the SVM algorithm.

Support vector machines (SVM) are a machine learning approach at tackling sentiment analysis. It tries to identify the optimum boundary (also called hyper-plane) in the domain space which can properly separate out different classes. The hyper-plane can be understood as ($N-1$) dimensional structure in the domain space, which separates out data points belonging to different classes. Textual data needs to be pre-processed into a more suitable form.

Most common approach is to use a bag of words model which does not consider the semantic structure of the original text. One concern that arises here is how to avoid noise due to frequent but non-contributing words such as “the”, “and”, etc. One possible approach is using the TF IDF vectorizer, which takes in a corpus of data and returns corresponding vectors. Advantage of this method is that these vectors can be used directly as input to the SVM [11]. Bag of words model can be utilized without compromising on the semantic structure of data through n-gram features. Each n-gram (n consecutive words) is considered as one token. Apart from using features based on frequency of words, features can be utilized based on the structure of the text, namely POS tagging and emoticon occurrences. Feature-related POS tagging could relate to number of nouns, verb, adjectives or the ratio of nouns to adjectives or vice versa [3]. Ortigosa et al. [8] have used a two-step process in achieving reduced dimensionality. First, all the words having less than two occurrences in the corpus were not considered as dimensions. Secondly, they employed correlation-based feature selection (CFS), which is a method of identifying the most significant dimensions. This massively reduced the number of dimensions from 8000 to 94. A hybrid method which used lexical features as feature vectors was also explored.

Stanford CoreNLP is a freely available text processing software which can be used directly for analysis. The inbuilt sentiment analysis of Stanford CoreNLP is based off a recursive neural network which provides sentiment scores in an incremental fashion, starting from smaller phrases which combine to give larger phrases whose scores are derived from the scores of the constituents and so on [4, 10]. The score is

in the form of an integer in the range of 0–5, with 5 being the most positive and 0 being the most negative.

Maximum entropy is a probability distribution estimation technique that is extensively used for natural language processing tasks. The essential principle of maximum entropy is that the probability distribution should be uniform when there is no pre-knowledge. Maximum entropy classification is used to estimate the polarity of data. There is no different from other learning technique, and the outputs of machine learning techniques are relied on the given training dataset of input. When using maximum entropy classification, the first step is to get the constraints that characterize the class-specific expectations for the distribution from labelled training dataset for the model distribution.

Phand [10] used Stanford NLP tool which follows based on maximum entropy which classifies the Twitter data into five categories of sentiment namely positive, somewhat positive, neutral, somewhat negative, negative, respectively. Mehra et al. [5] extracted words using the Chi-square method which identifies the words having skewed distribution. These words served as features that were served into a frequency of the maximum entropy classifier and noticed that in cases where no independent assumptions are made about the recurrence of words or the frequency of words, the models outperform Naive Bayes and nearest neighbour method in sentiment classification.

While an effective classifier on its own, a MaxEnt classifier can be augmented using various features. This method has been explored and compared to the basic method of using purely the classifier [12]. LDA and part-of-speech tagging can enhance the performance of the classifier. After modifying the traditional Gibbs method, the Labelled LDA methodology is achieved. This LDA is used alongside the MaxEnt classifier. POS tagging is done through Stanford NLP Core functionalities, and the classifier + POS model showed the best promise of all the combinations.

Habernal et al. [3] suggested a method to identify the sentiment of Czech input data. Using methods like stemming, lemmatization, the mistakes and misspells were reduced to standard International Phonetic Alphabet (IPA). N-gram, n character gram, two lists of emoticons were the main features considered by the classifier. Using these features along with information gain, the MaxEnt classified the input. In this case too, the MaxEnt classifier showed more consistent and better performance when compared to other methods like SVM classifier and Naive Bayes.

It is used often as maximum entropy classifier makes minimum assumptions about the previous distributions and when such assumptions cannot be made. Further, the MaxEnt classifier is used when no assumption about the conditional independence of the features can be made. Generally, the MaxEnt classifier has a higher training time as compared to Naive Bayes mostly due to its optimization problem that has to be solved for estimating the model parameters. But MaxEnt classifier does provide reliable results and is better in terms of memory and processing power consumption.

3 Discussion and Comparative Study

The following table presents summarized details of the papers studied as part of the survey.

4 Performance Analysis

Sentiment analysis can mainly be divided into two categories: Lexicon-based and machine learning-based. Advantage of such methods lies in the fact that there exist multiple sentiment dictionaries which are easily available, as compared to the training corpus required for the supervised learning approach. As suggested before, these methods do not fare as well as the supervised learning approach due to their inability to extract the context. One thing to be noted is that this fact makes the lexicon-based approach more robust for multiple applications. Supervised learning methods would have to be trained with a more application appropriate training data to be able to pattern from these sentences. For a long time, lexicon-based approaches have been used for sentiment analysis on their own. However, recent trends in the field of sentiment analysis have shown that a hybrid model could be much more accurate. Lexicon-based approaches can be made more accurate with better preprocessing of data. Some techniques include detecting idioms as one word, detecting emoticons, detecting interjections and language-specific semantic analysis.

A simple Naive Bayes classifier can be used to identify sentiments in a text. But as seen from Table 1, even a simple NB classifier can be made more accurate by preprocessing the data better. Since the focus is on data from social media, one way preprocessing could be conversion of emoticons into relevant words. This improves the accuracy of the classifier. Naive Bayes algorithm is sensitive to feature or parameter selection. It means that if the input features supplied to the algorithm are changed, the result varies significantly. To counter this, an ensemble of Naive Bayes with other algorithms can be used. One such algorithm is support vector machine. This makes the classifier less sensitive to feature selection. Maximum entropy is a probability distribution estimation technique that is extensively used for natural language processing tasks. MaxEnt classifier which uses prior results as a part of training dataset is more effective in NLP application. MaxEnt classifier has shown higher accuracy than Naïve Bayes in some cases where the features of training data are not conditionally independent of each other. Further, the classifier is more flexible of having many different types of data in a unified platform and does the classification. In aspect of social media, emoticons can be considered as a feature in the MaxEnt classifier thus not neglecting any data and increasing the accuracy.

Table 1 Discussion and comparative study

Methodology adopted	Dataset used	Accuracy	Research results	Shortcomings
Using the R sentiment analysis package by CRAN [1]	An adapter is used to connect to Facebook API in order to collect data in JSON format	Achieved an average accuracy of 67.7% across brands	Algorithm developed which gives varying accuracy	The result is not a very high accuracy for the developed model The assumption is made that the data that is scraped is purely in English
A system involving Web crawling, semantic and lexical analysis, sentiment analysis, and classification [6]	Data is obtained from Facebook via crawling. Only data relevant to Rail and La7 news programs has been considered	Not addressed	A complex system involving semantic and linguistic approaches to sentiment analysis was developed	Not much information is given on the actual method of analysis
Maximum entropy classification is used to maximize the entropy of classification system [5]	Obtained from Rec, arts, movies, reviews, from www.imdb.com	Addressed indirectly, through comparisons with varying parameters	A system based on statistical features of data and a MaxEnt model for sentiment classification	Features extracted did not use semantic nature of data
Stanford CoreNLP tools to do the actual sentimental analysis [4]	REST API is used to extract data from Twitter	Not addressed	An application was developed which extracted tweets in any required domain to identify the total number of positive and negative tweets	CoreNLP training API is difficult-to-use for not off-the-shelf training
Max entropy machine learning method with Stanford NLP [10]	Not addressed	The accuracy is around 80–90%	A tool is created that takes a word as an input. It obtains the Twitter data and subsequently the sentiment for the word entered	Sentiment search is done using tag words like “Amazon”, “shopping”, “India versus Pakistan”. Hence, limiting the search scope and the GUI was confusing

(continued)

Table 1 (continued)

Methodology adopted	Dataset used	Accuracy	Research results	Shortcomings
Analysis done using custom dictionary. Custom formula taking input as popularity on Twitter based on keywords [15]	Data is collected via Twitter API. Filters put on the data include words related to French election and date	The algorithm correctly predicted the winner of the election candidates was proposed which used certain keywords to determine whether the tweet was positive, negative or neutral towards a certain candidate	A new method for analysing the popularity of election candidates was proposed which used certain keywords to determine whether the tweet was positive, negative or neutral towards a certain candidate	neutral tweets were not really considered, and popularity was purely based on the frequency of keywords used. Some positive and negative weights were added to the keywords, but the actual grammar structure was not considered
SVM, Stanford CoreNLP, LDA, JST-based method [7]	Data is collected from 18 different message boards on Yahoo Finance Message Board for a period of one year	An accuracy of 54.4% was obtained	Three different approaches for stock market prediction are proposed. Results observed for a long term (1 year) were better than other models	The number of topics and sentiments for LDA and JST-based models are specified beforehand
Hybrid classification using lexicon-based approach followed by a ML-based classifier [8]	Not addressed	Achieved an average accuracy of 63.7%	Application which uses Facebook status data to analyse student sentiment and change	The duration for which sentiment change is analysed is fixed at 1 week. Words having a polar sentiment are given the same weight regardless of their actual emphasis

(continued)

Table 1 (continued)

Methodology adopted	Dataset used	Accuracy	Research results	Shortcomings
Supervised learning model, using MaxEnt and SVM [3]	Data is obtained from Facebook using Facebook Graph API and Java language detector	F-measure of 0.69 is obtained	An in-depth analysis of supervised methods which lead to the identification of the most optimum pipeline	Not a lot of research was available in sentiment analysis in the Czech language. The solutions for POS tagging were custom made for ease of use
Simple Naive Bayes approach is used on Facebook status data [14]	Data is collected by using Facebook Query Language (FQL) in combination with Graph API	Precision of 0.77 with a recall of 0.68 is obtained	A comparative study of Naive Bayes, Rocchio and Perceptron classifiers was conducted. Naive Bayes had the least recall and performed almost as well as Rocchio classifier	Assumes mutual independence of words in the sentence. Works on Facebook data because of large character limit. Emoticons are not handled
Naive Bayes with probabilistic model for determining polarity [9]	Data is collected via Twitter API and stored in HDFS	Accuracy in mapping of sentiment increases	A system of tweet extraction and sentiment analysis based on a Naive Bayes classifier	The methodology used simple Naive Bayes algorithm. Better and more accurate methods are available
Lexicon-based dictionary creation followed by support vector machine classification to identify sentiments [13]	Data is collected via Facebook API	Average precision of 89.5% is obtained	A custom made Vietnamese emotional dictionary was created, which was used to analyse data from different domains such as education and sports	the emotional dictionary created was based on an existing emotional dictionary for English language. Very little of natural language processing was done. Emoticons were not handled

(continued)

Table 1 (continued)

Methodology adopted	Dataset used	Accuracy	Research results	Shortcomings
Naive Bayes—support vector machines (NB-SVM) to compute a log-ratio vector between the counts of average word extracted from positive documents [2]	Data is obtained by using Twitter API and filtering the data using the hashtags relevant to the URI attack	Not addressed	A word cloud was created to show the words that were repeated most often. Tweets are also analysed based on other factors such as number of likes, retweets and favourites to determine the survival of tweets in the network	No support for other languages or emoticons
Machine learning model for classification of tweets followed by preprocessing and TfIdfVectorizer [11]	Data extracted from available datasets: Stanford sentiment 140, polarity dataset, University of Michigan dataset	F1 measures and accuracy tabulated for different algorithms	SVM, AdaBoosted decision tree and decision tree algorithms are combined, and the output obtained is more accurate than the individual algorithms	The result table is not easily understood and needs further clarification
MaxEnt classifier and its variations including POS tagging and LDA topic modelling [12]	Data is collected using iFeel, a PHP-based app that uses Facebook Graph API.	F1 score for binary classification is 0.85 whereas that for multi-class classification is 0.65	Evaluation of how binary classifiers scale to a multi-class problem, based on a Facebook status dataset	The raw data was not labelled by human annotators but was labelled on the basis of the emoticons. Improper filtering of emoticons caused error rate to be influenced. The context was unused or not extracted

5 Conclusion

Social media is the perfect platform to gather public sentiment about a certain product or event or even person. This work exhibits the various developments in the field of sentiment analysis focusing on the data obtained from social media. This data might not always be in a grammatically correct format or might not always be in a textual format. Hence, correct preprocessing of the data makes a lot of difference when it comes to efficient sentiment analysis of social media content. Another recent development in this area is using a hybrid approach, which combines both lexicon-based and machine learning-based approaches.

References

1. Dasgupta SS, Natarajan S, Kaipa KK, Bhattacherjee SK, Viswanathan A (2015, October) Sentiment analysis of facebook data using hadoop based open source technologies. In: 2015 IEEE international conference on data science and advanced analytics (DSAA). pp 1–3
2. Garg P, Garg H, Ranga V (2017, May) Sentiment analysis of the URI terror attack using twitter. In: 2017 International conference on computing, communication and automation (ICCCA). pp 17–20
3. Habernal I, Ptek T, Steinberger J (2013) Sentiment analysis in czech social media using supervised machine learning. In: Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis. pp 65–74
4. Kisan HS, Kisan HA, Suresh AP (2015, December) Collective intelligence sentimental analysis of twitter data by using standford nlp libraries with software as a service (SAAS). In: 2016 IEEE international conference on computational intelligence and computing research (ICCIC). pp 1–4
5. Mehra N, Khandelwal S, Patel P (2002) Sentiment identification using maximum entropy analysis of movie reviews
6. Neri F, Aliprandi C, Capaci F, Cuadros M, By T (2012, August) Sentiment analysis on social media. In: 2012 IEEE/ACM international conference on advances in social networks analysis and mining. pp 919–926 (Aug 2012)
7. Nguyen T, Shirai K, Velcin J (2015) Sentiment analysis on social media for stock movement prediction. *Expert Syst Appl* 42
8. Ortigosa A, Martn JM, Carro RM (2014) Sentiment analysis in facebook and its application to e-learning. *Comput Human Beh* 31:527–541, <http://www.sciencedirect.com/science/article/pii/S07475632130017519>.
9. Parveen H, Pandey S (2016, July) Sentiment analysis on twitter data-set using naive bayes algorithm. In: 2016 2nd international conference on applied and theoretical computing and communication technology (ICATCCT). pp 416–419
10. Phand SA, Phand JA (2017, October) Twitter sentiment classification using stanford nlp. In: 2017 1st international conference on intelligent systems and information management (ICISIM). pp 1–5
11. Rathi M, Malik A, Varshney D, Sharma R, Mendiratta S (2018, August) Sentiment analysis of tweets using machine learning approach. In: 2018 Eleventh international conference on contemporary computing (IC3). p. 1–3
12. Sushmitha RHV (2016) Sentiment analysis: facebook status message. In *J Eng Res Technol*
13. Trinh S, Nguyen L, Vo M, Do P (2016) Lexicon-based sentiment analysis of face-book comments in vietnamese language. *642:263–276*

14. Troussas C, Virvou M, Espinosa KJ, Llaguno K, Caro J (2013, July) Sentiment analysis of facebook statuses using naive bayes classifier for language learning. In: IISA2013. pp 1–6
15. Wang L, Gan JQ (2017, September) Prediction of the 2017 French election based on twitter data analysis. In: 2017 9th computer science and electronic engineering (CEEC). pp 89–93

Chapter 42

Survey on Detection and Prediction Techniques of Drive-by Download Attack in OSN



Madhura Vyawahare and Madhumita Chatterjee

1 Introduction

In this era of technology, the popularity of Internet is increasing not only among technologically strong people but also non-technical people. Criminals use multiple techniques to perform attacks. Attacker can use OSN platform to get crucial and private information of user for multiple purposes. Increased popularity of OSN and continuous availability of users are attracting more criminals and results in high rate of insecurity. Among few very dangerous attacks, drive-by download is one dominating attack. SANS institute conducted a survey where it was found that almost 48% of attacks were originating through drive-by download [1]. In this attack, the cybercriminal can use social media platforms to attract users, who unintentionally download virus or malicious software on their device, by clicking on a link deployed by the cybercriminal. This can result in obtaining remote access, steal user information or even create an entry point to other serious problems [2]. Taking control of the victim's machine is the final goal of drive-by download attack. Author Amir Javed et al. have mentioned the working in detail using four steps [1]: 1. Redirection: client is taken through multiple redirection processes to deliver him to the malware distribution site. 2. Obfuscation: attackers use obfuscation as the second step to hide the malicious scripts under several layers of obscurity. 3. Environment preparation: the attacker creates the environment to control the memory in order to inject malicious code in the browser's memory and jumps to execute the injected code. 4. Exploitation: this is the last step to perform the attack and compromise the vulnerabilities in browser plug-ins. Working of drive-by download attack is summarized in Fig. 1.

M. Vyawahare (✉)

Pillai College of Engineering, Mumbai University, Navi Mumbai, India

e-mail: madhura.vyawahare@gmail.com

M. Chatterjee

Pillai HOC College of Engineering and Technology, Mumbai University, Rasayani, India

e-mail: mchatterjee@mes.ac.in

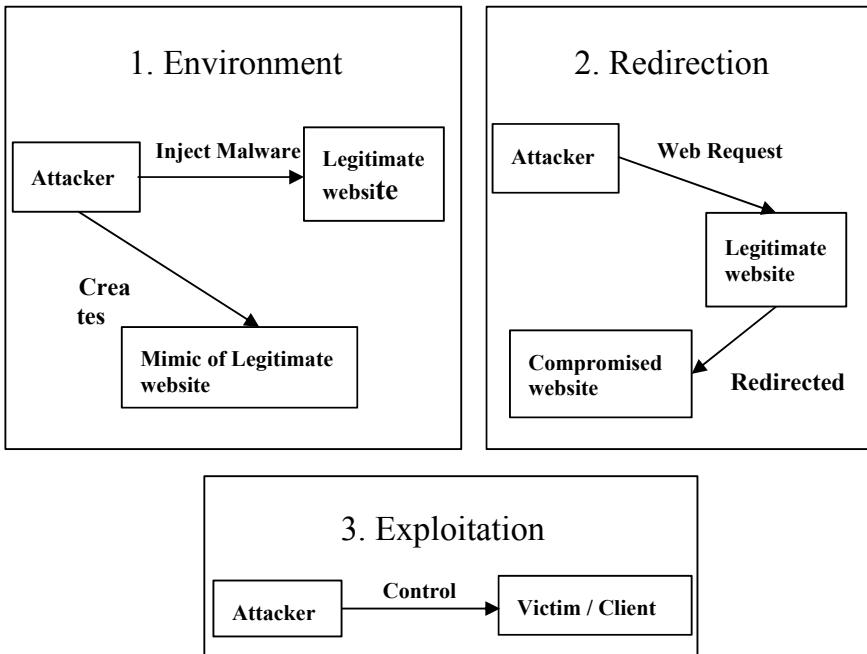


Fig. 1 Working of drive-by download attack

Many real-time examples explain why detection of drive-by download attack is important. In April 2012, the attacker used drive-by download to attack on personal computer systems running Mac OS. For this attack, malware writers created a fake toolkit for WordPress-based blogs, which created a backdoor and infected victim's blogs. Browsers visiting those blog pages were redirected to malware sites, which tried to install a “downloader”, the first part of the Flashback Trojan. Another piece of malware used a more traditional technique: It asked the user for permission to install (fake) Apple software, which was in fact the downloader. Once installed, the downloader would install more malware. The backdoor did not install anything except fake ads. It could have instead stolen the users' identities, emptied their bank accounts or used the infected machines to pump out spam and malware [3].

However, only detection of such an attack is not sufficient, prediction is also necessary. Few predictive models have been developed in the past which are able to predict the future occurrence of drive-by download attack. To improve the detection and prediction, monotonous traditional approach is not sufficient. Adaptive approach is required for the detection of such continuously adapting attack.

2 Detection of Drive-by Download Attack

The detection process of drive-by download attack started with the offline methods which are static in nature like blacklisting, traditional anti-virus tools and IDS. Multiple semi-real-time tools were designed to identify the attack through the code in web page. Detection process was then improvised to real-time detection system by using honeypots and then further enhanced using machine learning approaches.

The detection at the initial stages was done by using offline server infrastructure, and producing malware signatures and blacklists. Continuous update of the blacklist was becoming a hectic job and automation was needed for achieving this. Better detection techniques were need of time as spammers were also becoming smart [4]. To observe system's functionality, IDS was used. Hardware-based IDS provides efficient and fast detection system but they were costly and complex. Software-based IDS was not as fast as hardware-based one [5]. A semi-real-time system Cujo was designed to extract static and dynamic features of a web page and identify malicious contents [6]. Cujo was embedded in a web proxy, and Cujo transparently inspects web pages and blocks delivery of malicious JavaScript code. JavaScript is largely used for implementing and deploying context-rich web applications, but it is not a security-conscious language which makes JavaScript an obvious candidate for targeted drive-by download attacks [7]. Hence, many techniques focused on it.

Another application was designed named SpiderMonkey which was implemented for real-time detection of malware [8]. It is the first JavaScript engine that was developed and managed by Mozilla Foundation. Firefox browser includes SpiderMonkey which parses and executes JavaScript. SpiderMonkey was virtual machine (VM)-based honeypots called Strider HoneyMonkeys. Attractive feature of the application was the ability to search zero-day exploit-URL of the javaprxy.dll vulnerability. Another famous system using honeypot was ARROW which was using high-interaction client honeypot and able to determine these malware distribution networks from the secondary URLs and redirect chains [9]. ARROW gave noticeable results by identifying malicious signatures and determining malware distribution network and blocks it, but it is not completely automated. A tool JSAND was used by Wepawet service and many other researchers. It is a tool to identify malicious JavaScript code by combining anomaly detection with HTMLUnit emulation [10]. Authors Cova designed an approach for analysis and detection of malicious JavaScripts but they could not examine the zero pixel objects which are a recent and common techniques used by hackers.

2.1 *Detection of Drive-by Download Attack Using Machine Learning*

Researchers started using machine learning approaches to get better accuracy in feature matching and detection of attack. Highly precise, low-overhead and mostly

static detector for malware “Zozzle” was developed by authors Curtsinger et al. [11]. It is an application written in JavaScript and is able to prevent JavaScript malware by using Bayesian classification for hierarchical features of the JavaScript. It was deployed in a commercial browser. Zozzle classifies the URL as malicious or benign by checking from blacklist. The paper also gives performance comparative study of Cujo, JSAND and prophiler [11].

Static and semi-dynamic detection techniques are easy to bypass by exploiting attacks as compared to dynamic anomaly detection approaches, but dynamic techniques are resource intensive hence used less often as a real-time solution on the individual systems. A semi-real-time approach to detect drive-by download using dynamic analysis is presented by authors Jayasinghe et al. [7]. It dynamically monitors the bytecode stream. The automatic feature extraction approach dynamically evaluates browser Opcode called sequences, extracted from the JavaScript engine and uses supervised machine learning for classification. It detects previously unseen drive-by download attacks at run-time. Authors Ryo K. and Shigeki G. proposed a new defence method which can detect drive-by download attacks in real time because it judges an attack during HTTP communications [12]. It uses common features of HTTP request and response header fields and provides more detailed clarification of discrimination criteria. The method proposed by authors Takata et al. analyse JavaScript code for redirection and extracting the destination URL in the code [13–14]. Browser emulator called MineSpider is used for this. Code relevant to redirection is extracted independent of the analysis environment.

Aldwairi [15] proposed a system which was analysing the HTML tags for identifying 15 different features. MATLAB code was written and used for feature extraction from HTML page without visiting it. The system uses machine learning approach to classify these pages into malicious and trustworthy. Twenty-three different classifiers were used and tested from which five top classifiers were selected by the authors. The paper introduced three important features: onload() functions, zero-width i-frame and zero pixel objects. These features are useful in detecting drive-by download attack and are not previously used by any researchers.

When some researchers were focusing on web page content analysis [16], others were using a different approach of analysing the properties of web page URLs [17]. Authors Justin et al. have proposed a method to classify URLs as malicious and non-malicious [17]. Authors have considered two different features for classifying URLs: lexical and host-based features. In lexical feature analysis, hostname and path of URL were analysed using bag-of-words. In host-based features analysis, different features of host were analysed like: IP address properties, domain name properties, geographic properties, etc. The paper only focuses on URL-based features and does not consider other kinds of potentially useful sources web page content, and the context of the URL.

Authors Hiroaki K. and Hiroaki M. have analysed the redirection of the URL and object size to detect the malicious connections using decision tree. Redirection can happen with a legitimate site or due to drive-by download attacks. The authors analyse that drive-by download exploit typically involves redirection in conjunction with shellcode download [18].

While many researchers were working on detecting the drive-by download attack few started with predicting it.

3 Prediction of Drive-by Download Attack

Detection of drive-by download attack has been broadly investigated by many researchers. Detection of drive-by download attack is not sufficient to deal with, and prediction is required to completely save the OSN users from this attack. Detection is studied from a number of perspectives similarly for prediction few models are developed.

Authors Burnap et al. has explained a machine learning model for Twitter metadata to predict whether the URL is malicious or not [19, 20]. The classification model was trained for four different machine learning algorithms. Machine activity metrics were generated and logged during interaction with a URL endpoint. The URL was passed to a sandbox environment called client-side high-interaction honeypot system. The Capture HPC toolkit was selected, to dynamically take snapshots of machine activity at periodic time intervals. The classifier was trained to do prediction by analysing these snapshots. Intention to build a model was to predict malicious behaviour and to incorporate potential zero-day attacks for which the code may be previously unseen, but the behaviour is clearly malicious. These methods contributed in detecting and preventing drive-by download attack. Prediction of drive-by download is not achieved directly through this.

To predict the drive-by download attack Amir Javed et al. have designed a predictive model for Twitter platform [1]. Three main components of model are: feature extraction, persistent storage and machine learning. The feature extractor module was used for creating snapshots of machine activity at time interval “t” for a period of “p” by passing each opened URL to a sandbox environment. Each snapshot was written to a database for persistence, which includes (i) machine activity and (ii) metadata of the tweet containing the URL. Training of predictive model was done using four different machine learning algorithms: Decision tree, Naive Bayes, Bayes Net and Neural Network. Weka toolkit has been used to compare the predictive accuracy. The exclusion list of the honeypot gets periodically updated once every 14 days to include new techniques employed by cybercriminals to carry out a drive-by download attack. When the model was tested using an unseen dataset, it achieved an F-measure of 0.833 from log files generated at 2 s—that is 1 s after launching the URL it allows to stop the execution. It predicts malicious behaviour and hence does not execute such payload completely. Overhead of detecting the malicious action retrospectively and having to repair the system is avoided.

Authors Takashi Adachi and Kazumasa Omote have proposed two approaches to predict drive-by download attack [21]. The prediction of malware downloading during drive-by download has been done. The first approach evaluated vulnerabilities in malicious page to predict the malware downloading. Wepawet as the analysis tool was used to identify CVE-IDs in the page. Then, clustering was performed to

each CVE-ID using K-means++ algorithm which extracted features, based on these features prediction probability was calculated. It could not predict malware downloading without detecting vulnerabilities in malicious pages in advance. Approach-I was then enhanced using in-between language (Opcode) with JavaScript. The limitation of Opcode approach was it could only detect attacks which employ JavaScript. Approach-II was then proposed which combined approach-I and Opcode approach. This includes preparation phase to extract features from malicious pages and training phase to build prediction model for drive-by download attacks using supervised machine learning approach. This approach calculates the prediction probability of malware downloading. Experimental results show that approach-I achieves accuracy of 92%, FNR of 15% and FPR of 1.0% using Naive Bayes. Enhanced version approach-II embeds Opcode analysis and uses dynamic analysis. Approach-II has the prediction rate of 92% and improves FNR to 11% using random forest. Comparison of approach-I, II and Opcode approaches is presented in the paper.

4 Classification of Detection Techniques

To detect drive-by download attack, robust detection frameworks are required. It is possible only when detection solutions are enhanced using next-generation technologies. As drive-by download attacks are becoming increasingly sophisticated, the detection solutions should also become more promising. From the literature survey, we found out several techniques are designed in the direction to detect the drive-by download attacks.

The existing techniques can be classified in two different categories: using code and content and using URLs. Including the third parameter, “analysis of the characteristics of user account”, will help in not only detection but also will contribute in predicting the attack. Hence, to have a more efficient model of detection and prediction of drive-by download attack, the following categories can be taken into consideration: 1. Detection via analysing code and content of a web page, 2. Detection via analysing characteristics of URL, and 3. Detection via analysing characteristics of user account (Fig. 2).

5 Analysis and Discussion

The literature survey has already focused on the detection techniques considering code and content of web pages and also techniques using analysis of URL. The comparative chart makes it clear that not many techniques are developed for prediction of drive-by download attacks. Researchers have been focusing on machine learning techniques more in recent years. Social media is becoming a very huge platform for multiple purposes, and hence the availability of users is also attracting more criminals. Many of the recent methods are considering detection and prediction of drive-by download on various social media. Mostly researchers have considered any one

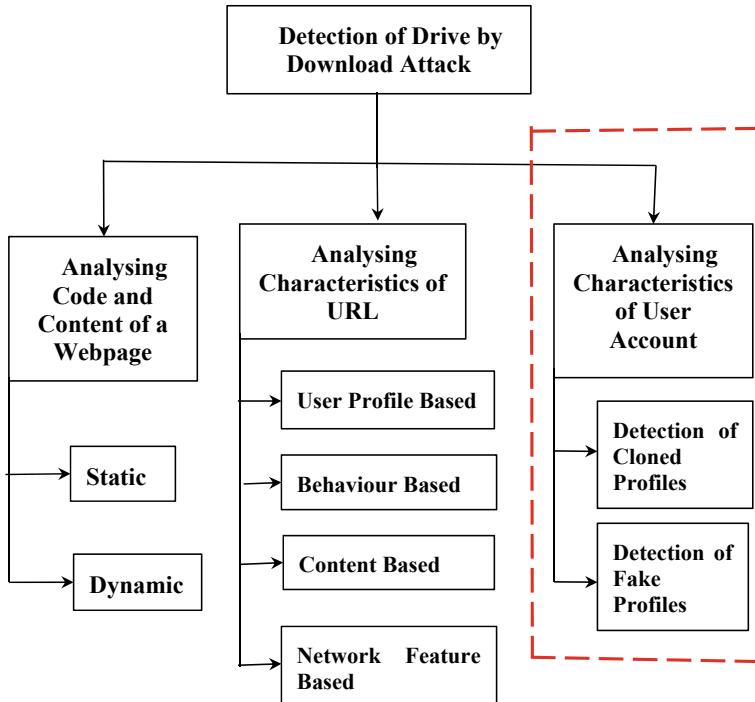


Fig. 2 Classification of detection techniques

social media, e.g. Facebook and Twitter. An adaptive, real-time framework is not yet defined for detecting, preventing and predicting these attacks happening in any social media platform. All the various ways are used for detecting attack where detection of drive-by download via analysing characteristics of user account is not considered directly by any researcher (Table 1).

Authors Cao et al. mentioned the characteristics of the user who is posting the link also helps in identifying the nature of the link [22]. If the previous history of the user and URL are not trustworthy then URL can be classified as malicious. To predict if the URL posted in OSN is malicious or not, it is very important to know who is posting this URL and who is clicking on this URL [22].

In terms of identifying the legitimacy of user in OSN, his/her OSN user profile behaviour is very important. This directs the focus on fake profile or Sybil attack detection. Profiles in OSN can be a real profile or it can be either cloned or fake. Identity cloning or cloned profile can be created on the same site or cross-site where attacker creates a duplicate account of a user and wisely uses it for malicious actions. Another attack is fake profile attack where the person does not actually exist in real and attacker creates fake identity to perform some kind of attacks [23]. Detection of the fake or cloned profile can contribute and improve accuracy of prediction of drive-by download attacks too. This will be our direction of future work.

Table 1 Comparative chart of detection and prediction techniques for drive-by download attack

Technique	Parameter						Honeypot used	Browser plug-in
	Real time	Detection technique	Machine learning technique used	Detection	Prediction	Programming language		
Cujo [6]	Semi-real time and static	Analysing web contents	No	Yes	No	JavaScript	No	Yes
SpiderMonkey [8]	Real time	Analysing URL char	No	Yes	No	JavaScript	Yes	Yes
ARROW [9]	Real time	Analysing URL char	No	Yes	No	HTTP content	Yes (high interaction)	No
Zozzle [11]	Static	Analysing web contents	Yes	Yes	No	JavaScript	No	Yes
Opcode analysis [7]	Semi-real time and dynamic	Analysing Web Contents and code	Yes (supervised)	Yes	No	browser Opcode	No	No
Fine-grained analysis [12]	Real time	Analysis HTTP header content	No	Yes	No	HTTP content	No	No
MineSpider [17]	Semi-real time, static and dynamic	Analysing URL char	No	Yes	No	JavaScript and HTML code	No	Yes
Vulnerability evaluation Opcode [21]	Semi-real time and dynamic	Analysing web contents	Yes	Yes	JavaScript	No	No	No
Detector using ML [15]	Real time	Analysis HTTP header	Yes	Yes	No	HTML code	No	No

(continued)

Table 1 (continued)

Technique	Parameter							
	Real time	Detection technique	Machine learning technique used	Detection	Prediction	Programming language	Honeypot used	Browser plug-in
Automated detector [18]	Semi-real time and dynamic	Analysing URL char	Yes	Yes	No	HTTP content	Yes	No
Real-time predictor [1]	Semi-real time and dynamic	Analysing URL char	Yes	Yes	Yes	Machine logs and HTML code	Yes (high interaction)	No

6 Conclusion

Rate and type of attacks and crimes in online social network are growing with the popularity of OSN. Drive-by download attack is one of the most popular and dangerous attacks among many other attacks. Drive-by download can act as an entry point to many other attacks like ransomware and cryptolocker. Many techniques have been designed for detecting drive-by download attacks in the past. An adaptive, real-time framework is required to detect prevent and predict these attacks more efficiently and accurately. Profiles with anonymous identity are most important because of several crimes in OSN. Drive-by download also originates from cloned or fake profiles most of the times. By enhancing the detection through analysing user characteristics of the account via which the URL is posted, prediction and prevention of drive-by download attacks can be improved.

References

1. Javed A, Burnap P, Rana O (2018) Prediction of drive-by download attacks on Twitter. Published by Elsevier Ltd. 12 Feb 2018
2. Sood AK, Zeadally S (2016) Drive-by download attacks: a comparative study. IEEE Comput Soc
3. <https://www.tomsguide.com/us/driveby-download-news-18329.html>
4. Chen C, Zhang J, Xiang Y, Zhou W (2016) Spammers are becoming “Smarter” on Twitter. In: IEEE computer society 1520-9202/16/\$33.00 © IEEE
5. Aldwairi M, Alansari D (2011) Exscind: fast pattern matching for intrusion detection using exclusion and inclusion filters. In: 7th international conference on next generation web services practices. Salamanca, Spain
6. Rieck K, Krueger T, Dewald A (2010) Cujo: efficient detection and prevention of drive-by-download attacks. In: ACSAC’10 Proceedings of the 26th annual computer security applications conference
7. Jayasinghe GK, Culpepper JS, Bertok P (2014) Efficient and effective realtime prediction of drive-by download attacks. J Netw Comput Appl 38:135–149
8. Wang YM, Beck D, Jiang X, Roussev R, Verbowski C, Chen S, King S Automated web patrol with strider HoneyMonkey: finding web sites that exploit browser vulnerabilities. In: Proceedings of network and distributed system security symposium
9. Zhang J, Seifert C (2011) ARROW: generating signatures to detect drive-by downloads. In: WWW 2011. Hyderabad, India, 28 Mar–1 Apr 2011
10. Cova M, Kruegel C, Vigna G (2010) Detection and analysis of drive-by download attacks and malicious JavaScript code. In: Proceedings of the 19th international conference on world wide web. ACM, New York, pp 281–290
11. Curtsinger C, Livshits B, Zorn B, Seifert C (2011) Zozzle: fast and precise in-browser javascript malware detection. In: Proceedings of USENIX security symposium
12. Kiire R, Goto S (2016) Detecting drive-by-download attacks based on HTTP context-types. In: Proceedings of the APAN-research workshop ISBN
13. Takata Y, Akiyama M, Yagi T, Hariu T Goto S (2016) MineSpider: extracting hidden URLs behind evasive drive-by-download attacks. IEICE Trans Inf Syst E99-D(4)
14. Takata Y, Akiyama M, Yagi T, Hariu T, Goto S (2015) MineSpider: extracting URLs from environment-dependent drive-by download attacks. In: IEEE 39th annual international computers, software and applications conference

15. Aldwairi M, Hasan M, Balbahaith Z (2017) Detection of drive-by download attacks using machine learning approach. *Int J Inf Secur Priv* 11(4):16–28
16. Priya M, Sandhya L, Thomas C (2013) A static approach to detect drive-by-download attack on webpages. In: IEEE international conference on control communication and computing (ICCC)
17. Ma J, Saul LK, Savage S, Voelker GM (2009) Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the SIGKDD conference. Paris
18. Kikuchi H, Matsumoto H (2015) Automated detection of drive-by download attack. In: 9th international conference on innovative mobile and internet services in ubiquitous computing
19. Burnap P, Javed A, Rana OF, Awan MS (2015) Real-time classification of malicious URLs on twitter using machine activity data. In: Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining, NY
20. Javed A, Burnapa P, Ranaa O (2017) Real time prediction of drive by download attacks on twitter. *J Inf Process Manage* (preprint submitted)
21. Adachi T, Omote K (2015) An approach to predict drive-by-download attacks by vulnerability evaluation and opcode. In: 2015 10th Asia joint conference on information security, IEEE. <https://doi.org/10.1109/asiajcis.2015.17>
22. Cao C, Caverlee J (2015) Detecting spam URLs in social media via behavioral analysis. In: Advances in information retrieval. Springer International Publishing Switzerland
23. Bilge L, Strufe T, Balzarotti D, Kirda E (2009) All your contacts are belong to us: automated identity theft attacks on social networks, In: Proceedings of the 18th WWW 2009, pp 551–560

Chapter 43

System to Fight Counterfeit Drugs



Soham Tendulkar, Alban Rodrigues, Keval Patel and Harshal Dalvi

1 Introduction

The current supply chain management (SCM) system of pharmaceutical industry is obsolete, and it is not transparent. “India’s pharmaceutical market is the world’s third largest in terms of volume” [1]. However, according to the World Health Organization (WHO), “35% of fake drugs sold all over the world come from India” [2]. The problem of counterfeit drugs within the supply chain costs the pharma industry billions; also it puts the patient’s life in danger. Particularly in developing nations where the WHO estimates that one in ten medical products (e.g., pills, vaccines, and diagnostic kits) is substandard or fake. Globally, more than a million people die every year after consuming counterfeit medicines [1, 3]. “India is one of the leading global producers of low-cost generic medicines due to its high domestic demand and inexpensive manufacturing costs” [1].

Drug manufacturers have been struggling to find a solution to the problem of counterfeit drugs. They want to trace securely, transparently, and rapidly the origin of these drugs. The government is also looking at a mechanism to get real-time visibility of the drug distribution that originates from one country and ends in some other.

S. Tendulkar (✉) · A. Rodrigues · K. Patel · H. Dalvi
Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, India
e-mail: ssten223@gmail.com

A. Rodrigues
e-mail: albanrod98@gmail.com

K. Patel
e-mail: kevalptl8097@gmail.com

H. Dalvi
e-mail: hddalvi.hd@gmail.com

Another motivation for the project is the unexplored specialized treatment in the field of medical science called as “Gene Therapy.” Patients have to provide samples of their DNA (or stem cells are withdrawn), and then, the drug is created specifically for that individual patient. But throughout that entire process, samples have to move from the patient to the drug manufacturing facility, and then from the facility back to the hospital, where it is stored and finally administered. With that many steps, manually checked, there are plenty of opportunities for mix-ups. Even right now, although there are only a handful of treatments utilizing this specific case-by-case approach, in the next five to ten years, that number will increase exponentially. So whatever amount of risk is being experienced now will only continue to build, meaning more and more personal data will need to be stored and analyzed appropriately. The systems we have in place currently are not sufficient to withstand real life-or-death scenarios that accompany these risky procedures. That is where blockchain comes in.

Why Blockchain?

For the prevention of counterfeit drugs, pharmaceutical industry needs an efficient supply chain management system, and the best available solution to develop a perfect SCM system is the blockchain technology. Blockchain is a distributed ledger system that has shown widespread adaptability in recent years, and a variety of market sectors sought ways of incorporating its abilities into their operations. Although so far most of the focus has been on the financial services industry, now projects in other service-related areas, such as health care, energy, and legal firms also started using this marvel. Whenever there is a product that is subject to a sensitive production process, the benefits of blockchain are evident. Blockchain is the best fit in those scenarios where privacy protection and data security are the highest priority [4].

2 Literature Survey

The following is the comparison of drug distribution in developed and developing countries:

1. In developed countries, there is a strong presence of insurance companies and limited out-of-pocket expenditure, whereas in developing countries, payments are made out-of-pocket.
2. In developed countries, there are strong well-defined laws, whereas in developing countries, there is weak regulatory and ill-defined laws.
3. In developed countries, prescribed drugs can only be dispensed with a formal prescription, whereas in developing countries, retail drug shops often dispense medicines and often act as the first point of healthcare contact.

The modern pharmaceutical supply chain is complex. Medicines are made from ingredients sourced from different countries. Final formulations are then exported.

Drugs change hands many times between the manufacturer and patient; every transaction is an opportunity for falsified or substandard products to infiltrate the market. FDA has promoted the Serial Number Identification (SNI) primarily as a way of preventing drug counterfeiting. The SNI for most prescription drug packages should be a serialized National Drug Code (sNDC). The sNDC is composed of the National Drug Code (NDC) (as set forth in 21 CFR Part 207) that corresponds to the specific drug product combined with a unique serial number, generated by the manufacturer or repackager for each individual package. Serial numbers should be numeric (numbers) or alphanumeric (include letters and/or numbers) and should have no more than 20 characters (letters and/or numbers) [3]. An example is shown below with a 10-character NDC. However, the transaction of multiple hands cannot be traceable by this SNI system [3] (Fig. 1).

The existing system has many loopholes which are out of reach of the traditional approach. This loophole includes Diversion of Drugs, Unregistered Pharmacies, and Pilfering and Heists of Drugs. Apart from these, the traditional system also fails due to its drawbacks, such as the traditional system uses individual database for all the nodes, there is no immutable storage, i.e., any node can change the data at its end, no common database for all nodes as each node preserves its own data, there is no common truth, difficult to track through customs as it involves lots of trusted third parties, too many trusted intermediaries are involved which increase the overall cost, and lastly, data is filled manually, mainly receipts are maintained, thus there is a chance of human error.

Blockchain technology is one of the most trending technologies in the world right now. It is considered to be the most secure way to store transactional data. Blockchain brings transparency, and thus, it is the best solution to implement in a supply chain where extensive liabilities are at stake. There are several open-source blockchain projects available, namely Ethereum, Hyperledger, etc. We are planning to design an Ethereum-based private blockchain network as it is best suited for B2C projects. Ethereum blockchain also has the feature called Smart Contracts, which allows us to write business logic that is executed automatically in the blockchain. MedShare is another proposal that aims to use blockchain technology in health care to share medical data from one entity to another in a trustless environment [5, 6].

Example of a serialized National Drug Code (sNDC)		
NDC		SERIAL NUMBER
55555 666 77	+	11111111111111111111
labeler code + product code + package code		unique, up to 20 characters

Fig. 1 Explanation of sNDC [3]

3 Proposed System

The proposed system aims to make the drug supply chain transparent enabling quality control throughout the supply chain. Blockchain provides a solution to the challenge of interoperability across suppliers and systems, as well as providing a verifiable change log, held in a secure electronic environment (Fig. 2).

In the proposed solution, we will be using Ethereum blockchain as it provides Smart Contracts feature. A permissioned (private) blockchain is best suited in this case as it allows a central authority to add or remove participants. Only the participants that have permission can access data in a private blockchain. Blockchain uses public key cryptography. That means each participant in the blockchain has a private key and a public key pair. The registration of each of the participants would be done by a central authority. This would ensure that only authorized participants can access and modify data on the blockchain.

Although a medicine is a physical asset, it has to be treated as a digital asset in the blockchain. Thus, every single medicine produced by the manufacturer will have a unique hash or unique identifier. Every time the medicine changes hands, this hash needs to be verified. Each transaction is sent to the blockchain for verification, and after verification, it is added to the blockchain which remains there as a permanent entry. Every transaction before getting added to the blockchain is validated by the Smart Contracts which is nothing but a business logic. It will verify the details such as date of manufacture and date of expiry.

As the amount of data stored on the blockchain is very critical to its performance, it is important to choose what data to store on the blockchain wisely. In our proposed solution, every participant has a unique ID that is registered by the central authority. Every transaction uses this unique ID of participants and the unique ID of the

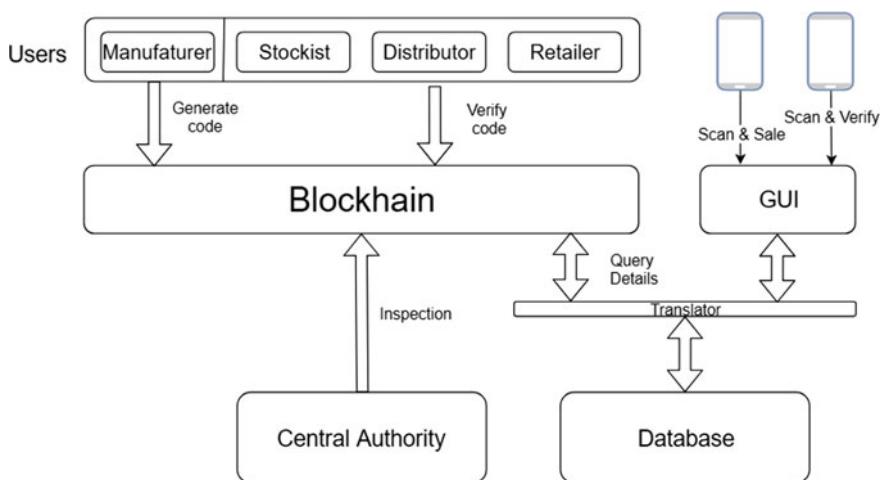


Fig. 2 System architecture

medicine to be transferred. This data will be stored on-chain, whereas all the other data related to medicine or manufacturer/distributor/retailer will be stored off-chain. The off-chain data may include name of the drug, place of manufacture, and name of manufacturer/distributor/retailer. The hash of this off-chain data will be stored on the blockchain, so that it can be verified for its authenticity.

The database unit is a traditional relational database that keeps track storing all the data in relational form. Data from the blockchain is translated and stored onto the database. A GUI provides user interface for common users to make transactions and payments. The Ethereum blockchain uses Web3 interface for Web applications and Web3j interface for android applications to create a light client. A user can scan the QR code on the medicine and get the complete trace of that medicine till its origin.

4 Future Scope

Cold chain is a temperature-controlled supply chain. Some specific drugs are very sensitive to temperature and require extensive care to be taken during their transport. Blockchain application can be extended to cold chain management as well. It requires integration of IoT with blockchain. A temperature sensor can be used per batch of medicines to constantly monitor the temperature of that batch while it is being transported. In case if the temperature deviates too much from the threshold, then the batch of medicine should be invalidated in the blockchain, which means that batch cannot be further sold to anyone in the supply chain and needs to be discarded [7, 8].

The drug being transported is a physical asset; we use the hash of the drug manufactured to verify at every stage. Blockchain brings transparency and traceability; however, a solution is required to ensure that tampering of a medicine does not take place and if it happens it needs to be updated in the blockchain immediately.

5 Conclusion

Thus, in this paper we reviewed all the literature work related to the problem of counterfeit drugs all across the world. We saw how a blockchain-based solution can be used to counter this problem. We discussed the loopholes in the current distribution system, and we also saw how blockchain, i.e., distributed ledger technology can help to overcome these problems. In the proposed system, we discussed how data can be stored in on-chain and off-chain forms. The system will be able to detect which drugs have been expired and alert accordingly so as to prevent repackaging of those drugs. We also highlighted in brief what can be the possible future scope of this system.

References

1. Sylim P, Liu F, Marcelo A, Fontelo P (2018) Blockchain technology for detecting falsified and substandard drugs in distribution: pharmaceutical supply chain intervention. In: JMIR research protocols. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6231844/>. Accessed 13 Sep 2019
2. (2018) How blockchain can help fight counterfeit drugs in India: forbes India blog. In: Forbes India. <http://www.forbesindia.com/blog/technology/how-blockchain-can-help-fight-counterfeit-drugs-in-india/>. Accessed 14 Sep 2019
3. Commissioner of the standardized numerical identification for prescription drug packages. In: U.S. food and drug administration. <https://www.fda.gov/RegulatoryInformation/Guidances/ucm125505.htm>. Accessed 13 Sep 2019
4. Mettler M (2016) Blockchain technology in healthcare: the revolution starts here. In: 2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom), IEEE
5. Xia QI, Sifah EB, Asamoah KO, Gao J, Du X, Guizani M (2017) MeDShare: trust-less medical data sharing among cloud service providers via blockchain. IEEE Access 5:14757–14767
6. Ethereum (2019) In: ethereum.org. <https://www.ethereum.org/>. Accessed 5 Sep 2019
7. Bocek T, Rodrigues BB, Strasser T, Stiller B (2017) Blockchains everywhere-a use-case of blockchains in the pharma supply-chain. In: 2017 IFIP/IEEE symposium on integrated network and service management (IM), IEEE, pp 772–777
8. Tian F (2017) A supply chain traceability system for food safety based on HACCP, blockchain and Internet of things. In: 2017 international conference on service systems and service management, IEEE, pp 1–6

Chapter 44

Comparative Analysis of Hand Gesture Recognition Techniques: A Review



Parth Shah, Raj Shah, Maulik Shah and Kiran Bhowmick

1 Introduction

A hand gesture recognition system is a vital component of any human–computer Interaction (HCI) system. Without the machine sensing, understanding and recognizing the gesture it would be impossible to create a system that would serve to fulfill a real-life need such as sign language translation or any other vision or gesture-based system.

1.1 Challenges for a Vision-Based System

The system must generalize over users and variation in the performance of the gestures.

First, the algorithm must be robust to varying global illumination changes and shadow artifacts along with tackling the commonly occurring problem of self-occlusion of hand gestures. Perfect gesture assessment (as in [7, 8]) is difficult and has not been studied before in settings of varying self-occlusion and different ambient conditions, yet many approaches rely on such information for their classification models.

Secondly, the computation speed of the system should be as fast as possible so that it can mirror and be applied to real-time use cases.

Thirdly, various other factors such as position and orientation of hand, camera placement and blurring of images can be the other factors that can add secondary layers of difficulty to the problem statement.

P. Shah (✉) · R. Shah · M. Shah · K. Bhowmick
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: Shahp98@gmail.com

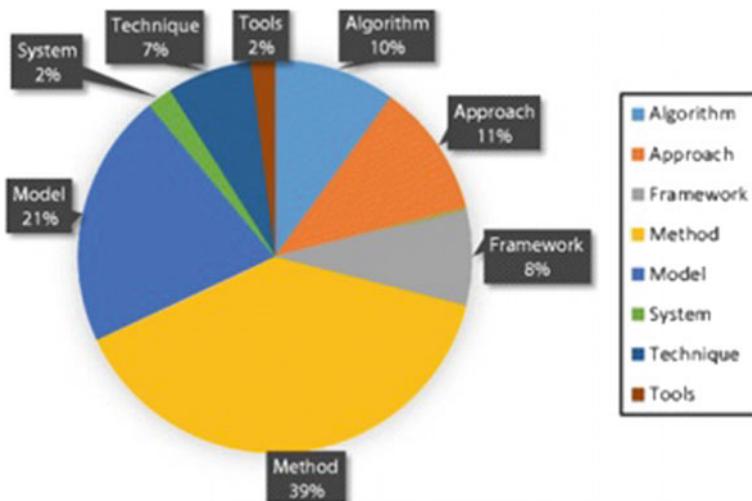


Fig. 1 Statistical analysis

All the algorithms in this paper have been implemented on the Senz3D hand gesture recognition dataset that contains gestures performed by four different people, each performing 11 different gestures repeated 30 times each, for a total of 1320 samples [4, 5].

In this paper, we implement and compare various hand gesture classification systems that utilize intensity channels comparing various machine learning and deep learning techniques such as logistic regression, k-nearest neighbors, support vector machine (SVM) and convolution neural networks. All of these algorithms have been independently implemented and further optimized before their results have been considered.

Figure 1 shows a statistical analysis of hand gesture recognition solutions according to various contribution types that have been historically presented.

There are a lot of incredible applications of this system. The primary one being, integrating our system with a voice recognition system, such as Alexa we can embed it in robots and make an autonomous sign language system for breaking down the barriers of communication between mutes with the rest of the populace.

2 Background and Related Work

The literature survey provides different methods and approaches for implementing hand gesture recognition. These provide insights to advantages and disadvantages of different approaches and techniques. The previously used methods which caught attention were as follows:

In [9], detection of hand region was achieved by template matching which uses random forest hand detector over whole image. Then, LDA is used to check whether each area contains the hand. In validation phase, k-means was applied on samples' distribution for reducing computation by calculating centroid of clusters. 91% accuracy was achieved using this approach.

In [3], histogram of oriented gradients (HOG) and local binary pattern (LBP) features were combined to accurately recognize hand gestures. Gentle Ad-Boost was applied for training. Using this approach, 91.4% accuracy was achieved.

In [1], a novel real-time idea was proposed for hand gesture recognition. Initially using background subtraction, the hand region from the original image is extracted. Then, palm and finger segmentations are carried out in five parts, viz. palm point determination, palm segmentation using circle of maximal radius, wrist point and palm masking, hand rotation, finger and palm segmentation. 96.69% accuracy was obtained.

In [6], 3D CNN for hand gesture recognition was proposed. There was a system that consisted of two networks, viz. high-resolution network (HRN) and low-resolution network (LRN). They performed experiments and observed that depth information alone performs better than using intensity data only. Yet the combination outperforms both. A classification rate of 77.5% was achieved using this approach.

In [2], a hierarchical architecture for the task of real-time hand gesture detection and classification using a deep 3D CNN for gesture detection and a shallow 3D CNN (C3D and ResNeXt-101) for gesture classification or recognition. This approach managed to achieve 94% accuracy and 91% accuracy in real time.

3 Experimentation

3.1 Dataset

In our models, we used the Creative Senz3D Dataset [7, 8]. The dataset contains 11 gestures that are repeated 30 times and are performed by four different people, for a total of 1320 samples. For each sample, color, depth and confidence frames are available. Intrinsic parameters are provided for the Creative Senz3D. The dataset contains several different static gestures acquired with the Creative Senz3D Camera.

3.2 Models

We have tried many models and approaches for recognizing hand gestures. The different models and approaches are mentioned below.

CNN—A. We applied tenfold cross-validation on the dataset making a 90%–10% split for testing and training. The neural network classifier consisted of three layers

of convolution 3D, each followed with max pooling 2D and dropout of 24% which is succeeded by two hidden fully connected layers.

CNN—B. A 90–10% split for testing and training was applied using tenfold cross-validation on the dataset. The neural network classifier consisted of two layers of convolution 3D, each followed with max pooling 2D which is succeeded by fully connected layers.

KNN. The images of the dataset were resized and then gray-scaled after which the HOG features from these images were extracted. A 90–10% split for testing and training was applied using tenfold cross-validation on the preprocessed images of the dataset. The KNN classifier is then trained and tested on this data. The n-neighbors parameter was set to 5.

Logistic Regression. The dataset images are resized and gray-scaled before HOG feature extraction. Once HOG features are extracted, training and testing splits are done according to tenfold cross-validation. The one-versus-rest multi-class logistic regression classifier is then trained and tested on these extracted HOG features.

SVM. The dataset images are preprocessed by resizing, gray-scaling and HOG feature extraction. These features are then split according to tenfold cross-validation and then fed to one-versus-rest multi-class SVM classifier. The SVM classifier then determines support vector margins and classifies images according to those.

4 Experimentation Results

We compared the classification performance between all the different models mentioned above. We observed that CNN-A performed slightly better than CNN-B, and hence we used CNN-A model architecture as our CNN model for comparison.

The table of comparison of accuracy is as shown below.

From Table 1 and Fig. 2, we can observe that logistic regression, KNN and CNN have performed very well, whereas the accuracy of the SVM model was quite low as compared to the rest. Moreover, there is not much of a difference between the accuracy performance of logistic regression, KNN and CNN.

Table 2 and Figs. 3 and 4 compare the different models' precision and recall scores. We noticed that CNN, KNN and logistic regression significantly outperformed SVM.

Table 1 Accuracy comparison between models

Classifier type	Accuracy (%)
Logistic regression	94.1
K-nearest neighbors	93.2
Support vector machine	49.2
Convolution neural network	96.2

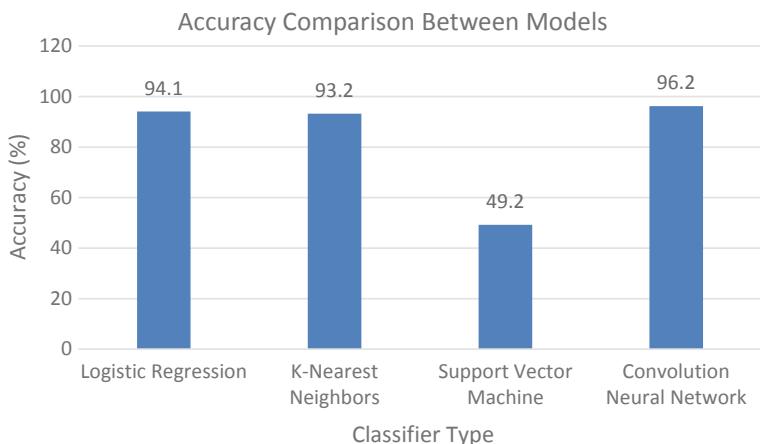


Fig. 2 Accuracy comparison between models

Table 2 Classification report

Model	CNN		KNN		SVM		Logistic	
	Precision (%)	Recall (%)						
G1	100	88.9	84.4	100	100	100	88.2	100
G2	92.3	100	93.7	100	50	100	100	100
G3	100	100	100	84.6	23	100	100	83.3
G4	100	100	93.3	93.3	18.1	100	92.3	100
G5	100	100	100	83.3	70	87.5	100	77.7
G6	100	100	100	90	50	100	100	92.3
G7	93.3	93.3	80	100	22	100	86.6	100
G8	92.8	100	100	93.75	90	12.1	100	91.6
G9	84.6	100	92.3	85.7	40	100	88.9	80
G10	100	100	100	100	90	100	100	100
G11	100	91.6	85.7	100	33.3	71.4	91.3	100

After carefully evaluating the two tables and the graphs, we can see that although CNN, KNN and logistic regression have performed quite similarly, the performance of CNN was slightly better than all the other models.

SVM is a binary classifier. But one can use SVM as a multi-class classifier in two methods: ‘one-versus-rest’ and ‘one-versus-one’. ‘One-versus-one’ uses a voting mechanism to select the resulting class from all classifiers, whereas in ‘one-versus-rest’, margins are determined for any one label from all the labels and then in determination of the second margin, the overlap with first margin is not considered. This ‘one-versus-rest’ strategy was implemented for comparison purposes.

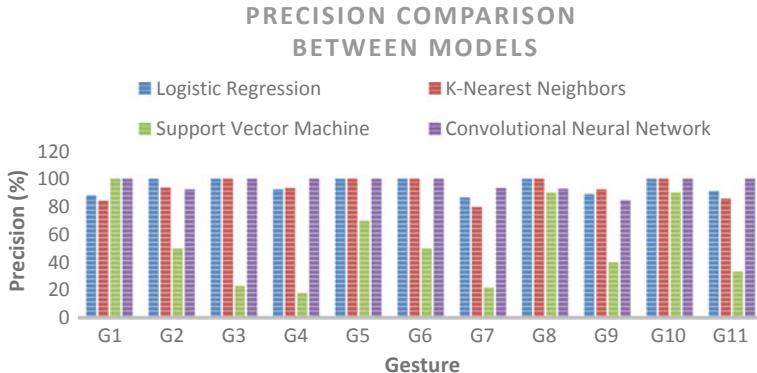


Fig. 3 Precision comparison between models

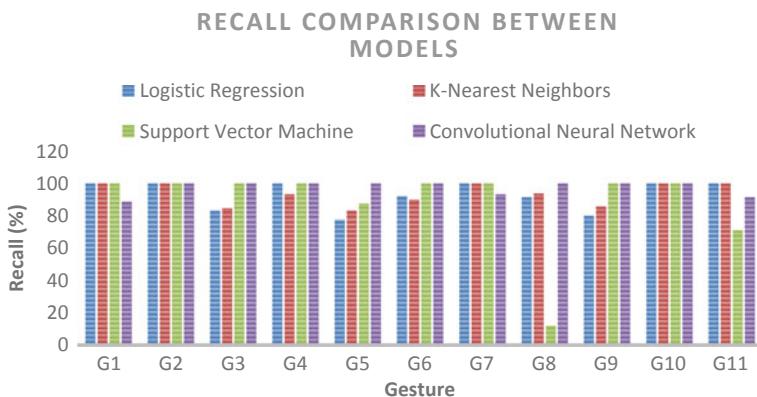


Fig. 4 Recall comparison between models

There are majorly two probable reasons why SVM performs poorly.

1. It has been seen that SVM does not work well with skewed/imbalanced data (i.e., data in which number of classes that fall in one category are far more than the other category). In the current dataset, we have equal distribution of instances in each class. Hence, while using the one-versus-rest strategy, as the number of instances in rest is always almost ten times the number of instances in the class being determined, the data becomes highly skewed. This could have resulted in the observed loss in accuracy as no SMOTe (Synthetic Minority Over-sampling Technique) or data augmentation techniques were implemented. On the other hand, logistic regression handles skewed/imbalanced datasets quite well even after using a similar one-versus-rest strategy.
2. By looking at the confusion matrix carefully, it can be hypothesized that almost all the outliers which are generally missed in the one-versus-rest methods for a

class label end up getting classified as the last label to be determined due to the overlaps not being considered. Hence, for SVM, the recall scores of all classes except one are on quite a higher side.

We believe that if enough time is spent on tuning the parameters of SVM classifier, then the accuracy can be improved.

5 Conclusion and Future Scope

We have thus compared all the different algorithms like SVM, KNN, logistic regression and CNN for our task and have concluded that the 12-layered convolutional neural network gives the best test accuracy of 96.212%, which is slightly better than the performance of KNN and logistic regression.

However, for future use in the SVM classifier, we would recommend the usage of gray-level co-occurrence matrix (GLCM) along with our color and histogram of gradients (HOG). This would increase the quality of feature extraction. This along with other techniques like bagging and boosting may still not be sufficient enough to beat the accuracy levels of the CNNs as they are far better in terms of accuracy, but it should definitely give a better accuracy than the current level.

One of the biggest problems of using CNNs for weight training is that of overfitting. Along with the drop out layers, we can also use regularization techniques to penalize the weight matrices of the nodes.

We would also like to handle dynamic image processing and event handling accordingly. This would make air gesture technologies in mobile phones more accurate and further broaden the scope of human–computer interaction (HCI).

References

1. Chen ZH, Kim JT, Liang J, Zhang J, Yuan Yb (2014) Real-time hand gesture recognition using finger segmentation. *Sci World J* 267872
2. Köpüklü O, Gunduz A, Kose N, Rigoll G (2019) Real-time hand gesture detection and classification using convolutional neural networks. In: IEEE international conference on automatic face and gesture recognition
3. Lahiani H, Neji M (2019) Hand gesture recognition system based on LBP and SVM for mobile devices. In: Nguyen N, Chbeir R, Exposito E, Aniorté P, Trawiński B (eds) Computational collective intelligence. ICCC 2019. Lecture notes in computer science, vol 11683. Springer, Cham
4. Memo A, Minto L, Zanuttigh P (2015) Exploiting silhouette descriptors and synthetic data for hand gesture recognition. In: Biasotti S, Tarini M, Giachetti A (eds) Italian chapter conference 2015—smart tools and apps in computer graphics
5. Memo A, Zanuttigh P (2018) *Multimed Tools Appl* 77: 27
6. Molchanov P, Gupta K, Kim K, Kautz J (2015) Hand gesture recognition with 3D convolutional neural networks. In: IEEE conference on computer vision and pattern recognition workshops (CVPRW), Boston, MA, pp 1–7

7. Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3D tracking of hand articulations using kinect. In: BMVC 2011, pp 101.1–101.11
8. Qian C, Sun X, Wei Y, Tang X, Sun J (2014) Realtime and robust hand tracking from depth. In: 2014 IEEE conference on computer vision and pattern recognition, pp 1–8
9. Sangjun O, Mallipeddi R, Lee M (2015) Real time hand gesture recognition using random forest and linear discriminant analysis. In: Proceedings of the 3rd international conference on human-agent interaction, pp 279–282

Chapter 45

Human Activity Recognition



**Chetashri Bhadane, M. Umair Siddiqui, Siddhant Soni
and Vijay Pratap Singh**

1 Introduction

Fitness trackers are devices connected to smartphones which are used to monitor or record various health-related activities of a user such as number of steps walked, calories burnt, etc. Due to ease of use and compatibility with different operating systems, the smartwatch segment is growing continuously.

There is an increasing demand in the market for products that help individual users to monitor their daily activity and get information such as how many steps they have walked today, how many calories have they burnt during the day and so on. The users want to see this data in a convenient manner.

In this paper, a method to recognize and track a user's daily activities using only their smartphones without any need for an additional fitness tracker device is developed. Most smartphones have sensors such as accelerometers and gyroscope sensors. The data from only these sensors is used to predict the activity that a user is currently engaging in, such as walking, running and climbing.

Algorithms such as support vector machines, deep neural networks, 1D convolution neural networks and LSTMs are implemented. The results of these algorithms are compared to find out which algorithm works the best for our model. Our dataset consists of six output classes like walking, jogging, climbing upstairs and downstairs, standing and sitting. In order to use our product, the users must have a smartphone with the required sensor whose values are used to predict the various physical activities.

C. Bhadane · M. U. Siddiqui (✉) · S. Soni · V. P. Singh
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

2 Literature Survey

Previous work on human activity recognition uses data from triaxial accelerometers which are body worn [9, 11]. Some attempts also include wearable systems [3] or biaxial accelerometers attached to different parts of the body [2]. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz performed HAR using smartphone triaxial accelerometer data [1]. They have implemented support vector machine (SVM) [4] binary classifiers and used the one-versus-all approach to generalize. SVM is one of the most robust and accurate classification algorithms. They have selected the hyperparameters through a tenfold validation procedure.

The dataset was gathered from a group of 30 volunteers who were instructed to perform activities in a specified order while carrying a waist-mounted smartphone. The six activities were standing, sitting, laying down, walking and walking down-stairs and upstairs. Every volunteer performed the activities twice: on the first run the smartphone was fixed on the left side of the belt, whereas the second time it was placed as the user preferred. Between each activity the subjects were told to rest.

A disadvantage of this approach is that the smartphone was waist mounted. In a natural environment, a user can either have his smartphone in his hand or in his pocket. Hence, even though the SVM gave an accuracy of 90.8% on this dataset, it would be different on a real-world dataset. Identical work involving special sensors have demonstrated comparable results (up to 96%), such as the work done by D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell and B. G. Celler which provided an accuracy of (90.8%) using waist-mounted triaxial accelerometer for 12 activities [6] and Y. Hanai, J. Nishimura, and T. Kuroda, which used a chest-mounted accelerometer for five activities and achieved 93.9% accuracy [5].

Convolutional neural networks are regarded as the most successful and commonly used deep learning model for feature extraction. Automatic feature-based approaches using deep learning models have been successfully applied to time series classification [12].

Song-Mi Lee, Sang Min Yoon and Heeryon Cho implemented a 1D convolutional network on a dataset collected from five graduate students who were instructed to record three activities which are walking, running and staying still, using smartphones which were carried in different positions [8]. The data was collected from the smartphone accelerometer, one sample per second. Their approach involved calculating the vector magnitude of the signal to eliminate any possible rotational inference present in the raw axial acceleration data. They achieved a maximum accuracy of 92.71% which outperformed the random forest approach (89.10%) [8].

However, this model only has three labels for output classes, i.e., run, walk and staying still which is less than what we need. The model does not predict if the user is climbing up the stairs or down the stairs. It does not differentiate between jogging versus running and sitting versus standing.

As we have seen the algorithm used by David Anguita et al. is support vector machines [1], whereas the algorithm used by Song-Mi Lee et al. that we've reviewed

1D CNN [8]. The accuracy of SVM came out to be 90.8% [1], whereas that of the CNN was 91.32% [8]. However, while collecting the data for SVM, the smartphone was waist mounted [10] which is not how the users will normally keep their smartphones. The drawback in the CNN model was that it predicts only three output classes, viz. run, walk and still [8].

3 Implementation Details

3.1 Dataset

The dataset used in this problem was collected by the wireless sensor data mining (WISDM) laboratory [7]. This dataset consists of data collected by controlled laboratory conditions.

Raw time series data was used. The dataset contains 1,098,207 examples with each example of one of the six classes of activity.

The dataset consists of activity data from 36 different users. Each example consists of user ID, activity type, timestamp and value of x -axis, y -axis and z -axis. This data is later converted into time segments of an appropriate length using the timestamps in the tuple. This will allow us to use time series data of the accelerometer for our learning models for predicting the activity performed by the user (Fig. 1).

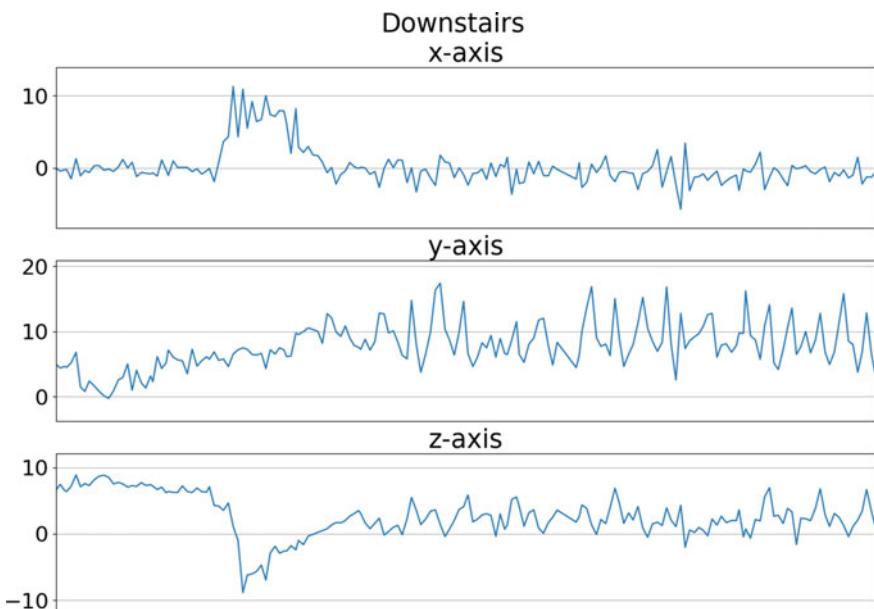


Fig. 1 Accelerometer time series for downstairs activity

In this paper, several algorithms like support vector machines, deep neural network, convolutional neural network and long short-term memory network are implemented. The accelerometer data is converted and reformatted into a time-sliced representation. Within one segment 80-time steps are used. The step variable is defined as the number of steps to take from one segment to another. If it is equal to time steps, then there is no overlap between the segments. The multidimensional tabular data is reshaped so that it is accepted by our model. The data set is split up into training, validation and test set. Examples of users with user id > 28 are used as our test set and the rest as our train set. So, 20,868 samples are in our training set and 6584 in our test set.

3.2 SVM

Here, a C-support vector classification model based on LIBSVM is used. The multi-class support is handled according to a one-versus-one scheme. A probabilistic SVM is used which instead of providing a single predicted output, provides a probability vector that represents how likely is an input sample belongs to a particular class. However, this method results in slower training.

Different kernels like linear, polynomial, radial basis function and sigmoid were experimented by us. The radial basis function or RBF kernel yielded relatively better results than the rest with an overall accuracy of 66%.

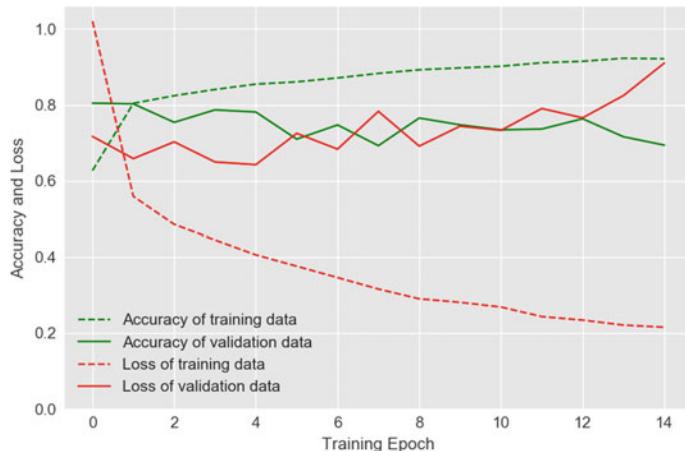
3.3 DNN

A neural network with three hidden layers of 100 fully connected nodes is used. The last two layers flatten the data and then run softmax activation function to calculate probability of each class (Fig. 2; Table 1).

3.4 CNN

A one-dimensional CNN is implemented. The time steps of the time series contain an order relationship and since CNNs work well with data consisting of spatial relationship [2], we have used a 1D-CNN.

First, two convolution layers, with 100 filters each is used. Then comes max pooling layer of size 3 followed by two more convolution layers with 160 filters each. An average pooling layer is used to avoid overfitting after which we apply a dropout layer with a rate of 0.5. The final output layer has six classes with softmax activation function. This forces all six outputs to add up to 1 (Fig. 3; Table 2).

**Fig. 2** DNN training accuracy and loss**Table 1** Model summary for DNN

Layer	Output	Params
Dense	(None, 80, 100)	400
Dense	(None, 80, 100)	10,100
Dense	(None, 80, 100)	10,100
Flatten	(None, 80, 100)	0
Dense	(None, 6)	48,006

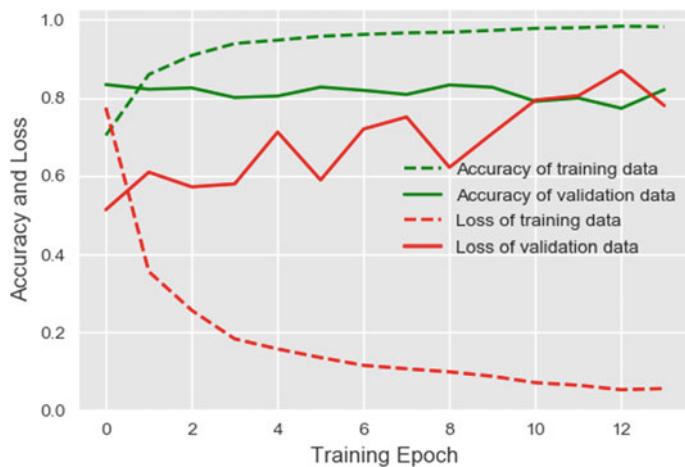
**Fig. 3** CNN training accuracy and loss

Table 2 CNN model summary

Layer	Output	Params
Conv1D	(None, 71, 100)	3100
Conv1D	(None, 62, 100)	100,100
MaxPooling1D	(None, 20, 160)	0
Conv1D	(None, 11, 160)	160,160
Conv1D	(None, 2, 160)	256,160
AvgPooling	(None, 160)	966

3.5 RNN-LSTM

Recurrent neural networks (RNNs) are mostly used while dealing with sequence problems [3]. The long short-term memory (LSTM) architecture is better as it takes care of the vanishing gradient problem of the RNNs. RNN tends to outperform other networks when the time series is longer.

Here, a simple LSTM network is implemented. Two LSTM layers are stacked on top of an output layer with a dropout of rate 0.2 in between. The LSTM layers have 50 units each. The final output layer has six outputs corresponding to the six labels and uses the sigmoid activation function (Fig. 4; Table 3).

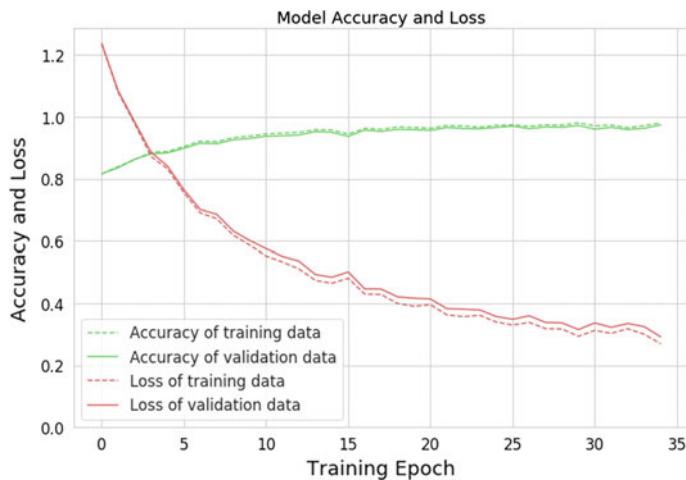
**Fig. 4** RNN training accuracy and loss

Table 3 RNN model summary

Layer	Output	Params
LSTM	(None, 80, 50)	10,800
LSTM	(None, 50)	20,200
Dropout	(None, 50)	0
Dense	(None, 6)	306

Table 4 Classification report for LSTM-RNN

Label	Precision	Recall	f1-score
Downstairs	0.9	0.91	0.91
Jogging	0.99	0.99	0.99
Sitting	0.99	0.99	0.99
Standing	0.99	0.99	0.99
Upstairs	0.9	0.92	0.91
Walking	0.99	0.98	0.99
Avg/total	0.97	0.97	0.97

Table 5 Classification report for CNN

Label	Precision	Recall	f1-score
Downstairs	0.74	0.71	0.72
Jogging	0.98	0.93	0.96
Sitting	0.8	0.94	0.86
Standing	0.99	0.69	0.82
Upstairs	0.75	0.77	0.76
Walking	0.92	0.97	0.94
Avg/total	0.9	0.89	0.89

4 Result

With the tables and charts below, we can compare the performance by each algorithm that we tested. It is observed that LSTM provides us with a maximum f1-score of 0.97. While CNN has a respectable 0.89 f1-score, SVM and DNN proved to be the weaker models for this particular problem. The highest accuracy from the LSTM is achieved when we increase the length of the time segment (Tables 4, 5; Fig. 5).

5 Conclusion

As seen from the results, the LSTM network outperforms every other algorithm. However, the CNN model is close behind. If we increase the time steps the LSTM would be far ahead. LSTMs, although more complicated, proved to better when

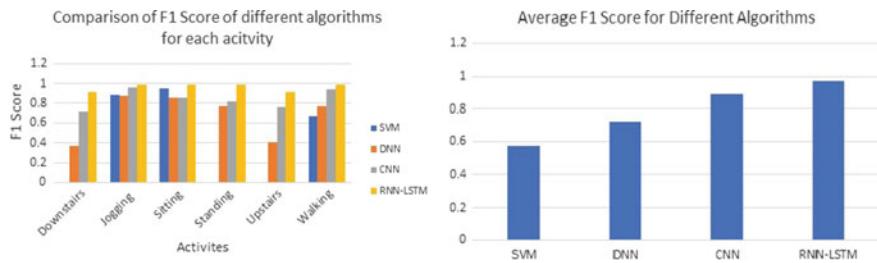


Fig. 5 Performance comparison of algorithms

dealing with time series, since they can store information about previous values and exploit the time dependencies between the samples. However, if we combine the power of CNNs to find local patterns and LSTMs, we might be able to create more powerful models. With the rise in popularity and ease of availability of activity trackers, we can make use of sensor data to provide a better prediction of the activity of the user. Today, most activity trackers have an arsenal of sensors like accelerometers, heart rate monitors, altimeter/barometers, GPS/GLONASS. The data from these sensors will provide a more sophisticated prediction. With the ease of deploying deep learning models for smartphones, these models can be used accurately for fitness tracking apps.

References

1. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2013) A public domain dataset for human activity recognition using smartphones. In: ESANN 2013 Proceedings, European symposium on artificial neural networks, computational intelligence and machine learning, Bruges, Belgium
2. Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. In: International conference on pervasive computing, pp 1–17. Springer, Berlin, Heidelberg
3. Casale P, Pujol O, Radeva P (2011) Human activity recognition from accelerometer data using a wearable device. In: Iberian conference on pattern recognition and image analysis, pp 289–296. Springer, Berlin, Heidelberg
4. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
5. Hanai Y, Nishimura J, Kuroda T (2009) Haar-like filtering for human activity recognition using 3D accelerometer. In: 2009 IEEE 13th digital signal processing workshop and 5th IEEE signal processing education workshop. IEEE, pp 675–678
6. Karantonis DM, Narayanan MR, Mathie M, Lovell NH, Celler BG (2006) Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Trans Inf Technol Biomed* 10(1):156–167
7. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newslett* 12(2):74–82
8. Lee SM, Yoon SM, Cho H (2017) Human activity recognition from accelerometer data using convolutional neural network. In: 2017 IEEE international conference on big data and smart computing (BigComp), IEEE, pp 131–134

9. Lukowicz P, Ward JA, Junker H, Stäger M, Tröster G, Atrash A, Starner T (2004) Recognizing workshop activity using body worn microphones and accelerometers. In: International conference on pervasive computing, pp 18–32. Springer, Berlin, Heidelberg
10. Ordóñez F, Roggen D (2016) Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115
11. Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity recognition from accelerometer data. Aaai 5(2005):1541–1546
12. Yang J, Nguyen MN, San PP, Li XL, Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. In: Twenty-fourth international joint conference on artificial intelligence

Chapter 46

Reference Model Storage Covert Channel for Secure Communications



Dhananjay M. Dakhane and Vaibhav E. Narawade

1 Introduction

The communication channel used for the transmission of information through legitimate network traffic is an overt channel [1], whereas the covert channels are the hidden channels used for the secret communication. The term, covert channels, has been used in various ways like network steganography, information hiding, etc. It describes the process of hiding information in network protocols. The first time Lampson defined covert channels as channels, which is not intended for information transfer at all [1]. The new definition defined by US DoD TCSEC, “any communication channel that can be exploited by a process to transfer information in a manner that violates the system security policy”, commonly known as Orange Book [1]. The term information hiding simply had not been invented for computer network, when the first covert channels in network protocols were proposed [2].

2 Related Work

Rowland initially discovered covert channels in the TCP initial sequence number (ISN) field [2]. The ISN is only transferred when a new connection is established and has a size of 4 bytes. Rowland also developed an enhanced version of the ISN-based covert channel with the goal to hide the sender’s address. Therefore, a bounce server is introduced. The bounce server is used by the sender to send messages to the

D. M. Dakhane (✉) · V. E. Narawade
Ramrao Adik Institute of Technology, Nerul, India
e-mail: dhananjay.dakhane@rait.ac.in

V. E. Narawade
e-mail: vaibhav.narawade@rait.ac.in

receiver. Therefore, the senders send a spoofed TCP packet to the bounce server. The packet contains the receiver's source address and thus lets the bounce server respond to the receiver that receives a packet which does not contain the sender's address [3]. The bounce server will increment the ISN that is transferred as acknowledgment number to the receiver that has to decrement the acknowledgment number to get the original ISN value. This model developed an enhanced ISN-based passive covert channel by indirectly initiating a TCP connection to transfer hidden information [4]. Instead, a covert channel sender waits for a regular TCP connection and modifies an ISN generated track by an ISN modification layer in the Linux kernel [5]. Another approach is to use TCP and UDP port numbers to send hidden messages [6]. J. Giffin used TCP timestamps to embed covert payload [7] (timestamps are an optional header component of the protocol). This author applied a minimal delay to create a covert timing channel in this way. In storage-based covert channels, certain fields of header in the packet are used throughout the stream. All the packets of the corresponding stream contain covert data; this is the key feature of the storage-based covert channels. The previous work done by Joanna Rutkowska and Steven J. Murdoch gave the two schemes for embedding the covert channels in TCP/IP. Joanna Rutkowska designed the scheme called "NUSHU" [4], and Steven Murdoch designed the scheme for covert channels called "Lathra" [5]. "NUSHU" encrypts the data before actually embedding it into TCP ISN field [4]. These result in normal distribution unlike that is generated by Linux and so will be detected by another TCP tests. "NUSHU" also exhibits characteristics of its own which may be exploited. The encryption operated by DES algorithm encrypts the combination of four different entities (Source port + Destination port + Source IP address + Destination IP address) with a shared key, then XORing the first 32 bits of the resulting key stream with the hidden data. When collisions occur, the ISNs can be XORed to remove the key stream; the result is XOR of two plain texts. If these plain texts are the same, as in the case, when the data are not being sent, the result would be zero. In other cases, redundancy in encoding would be apparent [4]. So, in certain cases, this protocol fails. While "Lathra" [5] designed by Steven j Murdoch and Stephen Lewis works perfectly in those cases. In both of the above covert embedding schemes, headers of the TCP/IP layer are modified, thus leaving some scope for warden in the network to detect these channels. To be totally undetectable while embedding the covert data in the packet, we should not modify anything in the headers whether it is TCP header or IP header. In such a way, it becomes difficult for warden to distinguish between overt data packet and covert data packet.

3 Covert Threat Model

The threat model for covert communication can be understood by simple scenario shown in Fig. 1. Consider an analogy, Alice is a covert sender who is connected to the Bob and exchanging some covert information by using propose covert model (TCP or ID reference model). It is an active covert channel, so both the covert users,

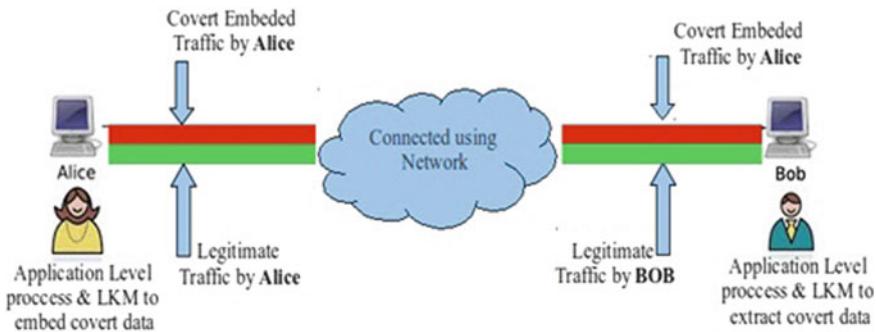


Fig. 1 Proposed covert threat model

i.e., Alice and Bob, have corresponding application-level processes running at the application layer. This generates a legitimate TCP traffic. In our proposed covert model, Alice uses covert kernel module to exploit the TCP traffic generated by the application-level process. Similarly, on the receiving side, Bob who is the covert receiver extracts the covert information using similar mechanism. The features of our reference model are,

1. Normal Distribution of ISN

In our proposed reference model, packets generated by the Linux kernel 3.2 follow the semantics of TCP ISN generation specified in the RFC1948 so that the TCP-SQN (sequence number) used by the packets which contains covert message covers the same space as the overt data packets. That is why the normal TCP-SQN distribution for covert packets is observed.

2. Persistent Connection

In the previous approaches, the researchers use new connection for each unit of covert message. While in our proposed model, it uses a single persistent connection for communicating the complete message. The advantage of this approach is to enhance the bandwidth of covert communication.

3. Reliable Covert Transmission

In the case of packet loss, kernel itself performs packet retransmission, as in our proposed model, all of the tasks of packet retransmission are done by the kernel itself. This enhances the reliability of our proposed model for delivery of the covert message.

4 Proposed Model

When two parties need to transfer the data using TCP, the sender machine will create a new TCP connection. As sender machine initiates the connection, it generates the

first sequence numbers, i.e., TCP ISN. The sequence number has dual roles. If SYN flag is set, then it is the initial sequence number (ISN). When SYN flag is clear, then the sequence number is the accumulated sequence number of the first data segment of the current session. The TCP ISN must be chosen such that the sequence numbers of new incarnations of a TCP connection do not overlap with the sequence numbers of earlier incarnations of a TCP connection [5]. Storage covert channel using TCP-SQN (Sequence Number) field as a reference, to be undetectable it is necessary that ISN's generated by these channels should cover the same space as the ones generated by the system without modification. The previous research shows that whenever the semantics of the packet are disturbed, it becomes detectable by the warden present in the network. As we are using TCP-SQN field as the storage channel for our covert data, it is most important that ISN numbers generated by our protocol should look like normal ISN distribution. To achieve this, we use ISN generated by the operating system's kernel itself and do not generate any random numbers for ISNs. This will ensure us that whatever ISN number we are using will look like any other ISN generated by the same system [8]. In this way, we will automatically follow all the specified structures and semantics used by the kernel. Now to convey the covert data, we use the TCP payload; this will contain the key; using this key, we can extract the covert data from kernel generated TCP-SQN. Basically, this key is the sequence of our covert data bits in the TCP-SQN field. This key can be distinguished throughout the payload, and each byte of the key is placed at different positions in the TCP payload [9]. We call these positions as a data pointer. These data pointers contain the symbols from the symbol table, which is usually used in the covert communication. This symbol table contains unique symbols for positions in the sequence. So actually, sender will append the TCP payload and not header to send covert data. Hence, without disturbing any header or packet semantics, covert data can be embedded into the packet.

4.1 Sequence Number Reference Model

TCP sequence number (SQN) field is used to maintain persistent TCP session between two ends. During the initial handshake between two systems, initial sequence number (ISN) is generated by the sender system, and it is carried forward by both communicating systems. The length of this field is 32-bit long. So, it is an obvious choice for covert communication by any attackers. This is because the highest possible bandwidth can be achieved to exchange the covert message. The NUSHU is developed for ISN field to embed covert message in encoded ISN [4]. While in our proposed covert reference model, covert message is a reference pointer to the sequence number field. In our proposed TCP-SQN reference model, the loadable kernel module (LKM) will embed the covert data into the TCP payload. However, the covert data in the TCP payload would not be actual covert bytes that we want to send. The reference positions which indirectly pointing to individual bits of kernel generated TCP sequence number will be in the TCP payload [10]. Figure 2

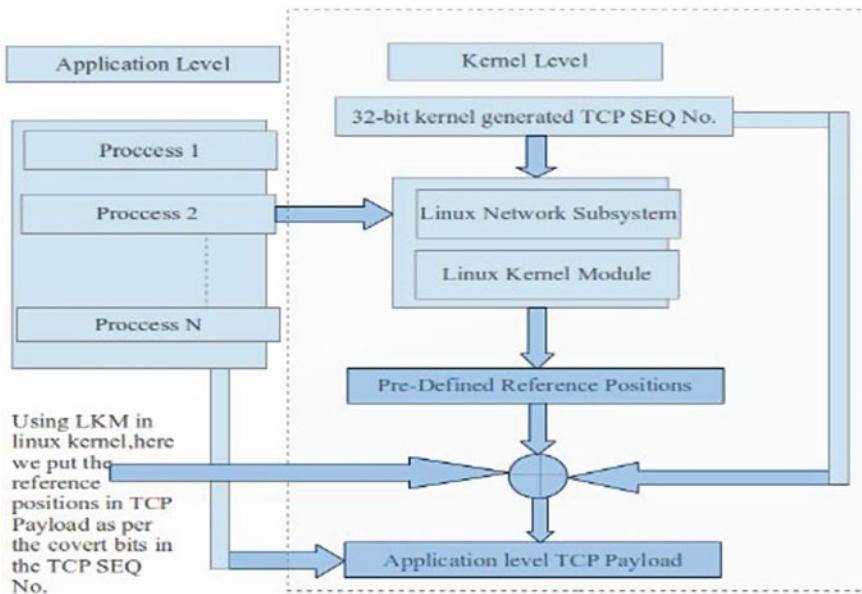


Fig. 2 Sequence number reference model

shows the TCP-SQN reference model architecture. To allow us to check whether the covert communication using TCP-SQN reference model is successful; the simple proof of concept test for client and server over TCP is conducted. These applications are written in Java and tested on Ubuntu environment having kernel version of 3.8. The architecture of sequence number reference model is shown in Fig. 2. The TCP sequence number field is considered to select the reference bit positions and embedded these positions in the corresponding payload. The covert bandwidth of this model is 8-bit, 16-bit, and 32-bit per packet depending on the model used for communication. The maximum covert bandwidth per packet is 32 bit and minimum is 1 bit per packet. This proposed model is developed 32-bit per packet.

4.2 IP Identification Model

The 16-bit identification (ID) field in IP is used to uniquely identify the fragments of a particular datagram. Fragments of a particular datagram are assembled if they have the same source, destination, protocol, and identifier. The identifier is being chosen to be unique for same source, destination pair, and protocol for the time the datagram (or any fragment of it) could be alive on the Internet. The IP identifier (ID) fields have 65,536 different values. It is important for an operating system to have some sort of mechanism to control the identification (ID) numbers correctly. In this model, the only important stage is to insert the covert reference pointers into the TCP payload

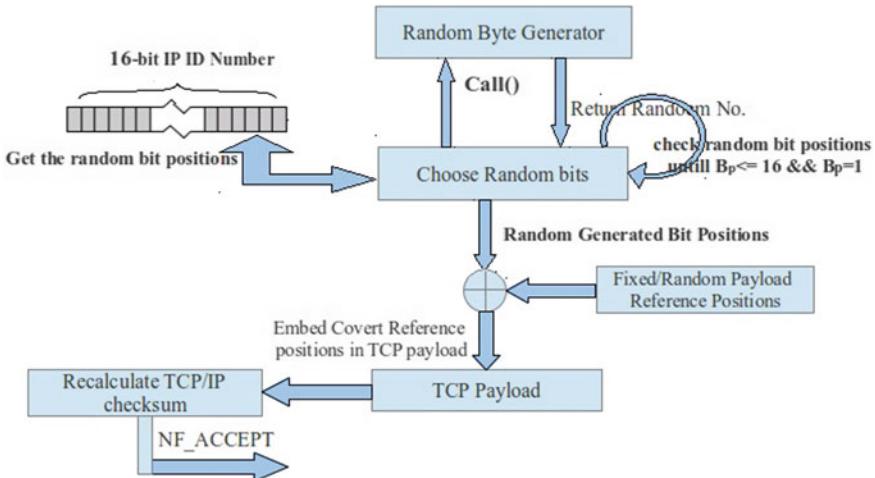


Fig. 3 Reference model for IP identification

which match the covert message pattern with identification number. If this is done, then the covert communication remains undetectable. The architecture of this model is shown in Fig. 3. In this module, the position of covert reference pointer in the TCP payload is fixed for TCP sequence number as well as for IP identification. However, it is possible to choose random covert reference positions, but it will take an overhead of handshaking every time, whenever the corresponding covert reference positions in the TCP payload change. So for simplicity, it is considered predefined TCP payload positions. The checksum for TCP and IP need to recalculate as payload has been altered.

5 Experimental Results

The test of this proposed model is taken in LAN (100+) machines in a network. This model is tested using various tests.

Covert Channel Cover Correctness Test

This test is conducted with the proposed covert reference model. All tests conducted for text message and image. The purpose of this test is to verify the percentage of covert message received at covert receiver end. The Java application verifies the accuracy of the covert message received at covert receiver. For all tests, covert message size is text 1024 bytes and image of 473,831 bytes. Table 1 shows the summary of the experimental results.

The same experiments were carried out for the combinations of 32-bit TCP-SQN (sequence number) and 16-bit IP identification number, in the same environment. The experimental results are shown in Table 2.

Table 1 Test for 32 bit TCP-SQN model

Parameters	Text message	Image
Message size	1024 bytes	473,831 bytes
Packets required	32 and 64	14,809 and 29,615
TCP data	233,606 bytes	108,034,049 bytes
Bandwidth	32 bits/packets	32 bits/packets
Message received	100%	100%

Table 2 Test for 48 bit model

Parameters	Text message	Image
Message size	1024 bytes	473,831 bytes
Packets required	32 and 64	14,809 and 29,615
TCP data	233,606 bytes	108,034,049 bytes
Bandwidth	32 bits/packets	32 bits/packets
Message received	100%	100%

The above results show the accuracy of the proposed reference model; the covert message received 100% at the receiving end with 0% packet loss.

6 Conclusion

In our propose model, the ISNs and sequence number generated by the TCP/IP stack are not changed for covert communication. The semantics of the ISN and sequence number are not changed while referencing the covert message. As a proof of concept, this model can create totally undetectable storage covert channel. It also enhances the bandwidth of storage covert channel up to 48 bit/packets. It is important to note that we are using the single persistent connection for the entire session, and all of the communication takes place through this single connection.

References

1. Dodd US (1985) Trusted computer system evaluation criteria
2. Rowland C (1996) Covert channels in the TCP/IP protocol suite. <http://www.firstmonday.org/issues/issue25/rowland>
3. Zander S, Armitage G, Branch P (2007) A survey of covert channels and counter measures in computer network protocols. (Accepted for publication in IEEE Communications Surveys and Tutorials)
4. Rutkowska J (2004) The implementation of passive covert channels in the Linux kernel. Speech held at the 21st Chaos\Communication Congress, Berlin and Germany
5. Murdoch SJ, Lewis S (2005) Embedding covert channels into TCP/IP. In: Proceedings of 7th information hiding workshop
6. Borland T (2013) Guide to encrypted dynamic covert channels. <http://turboborland.blogspot.com/2008/12/guide-to-encrypted-dynamic-covert.html>
7. Giffin J, Greenstadt R, Litwack P, Tibbetts R (2013) Covert messaging through TCP time stamps. In: Proceedings 2nd international conference on privacy enhancing technologies. 194(208)
8. Allix P (2007) Covert channels analysis in TCP/IP networks
9. Zi X, Yao L, Pan L, Li J (2010) Implementing a passive network covert timing channel. In: Elsevier computer and security 2010
10. Borders K, Prakash A (2009) Quantifying information leaks in outbound web traffic. In: Proceedings 30th IEEE symposium on security and privacy, pp 129–140

Chapter 47

Texture Synthesis and Style Transfer for Aesthetic Design Creation



**Aditya Shah, Dhruvin Shah, Harsh Shah, Sneha Shahane
and Khushali Deulkar**

1 Introduction

Neural style transfer [1] is basically a parametric model which is used for texture synthesis from images using convolutional neural networks [2]. NST is able to extract style and texture from abstract images. Generating creative images using this style transfer has been a very imaginative application in the field of image processing and computer vision. Neural style transfer uses a pretrained CNN to generate styled images. CNN uses feature representation to capture simple and complex features, and these features are then used to encode the image. This encoding nature of CNNs is useful for style transfer. Given one *content* image and one *style* image, the algorithm aims to create a new, *target* image which contains the desired content and style components. Thus in the target image, the objects and contents are similar to the content image while the style, colours and textures are similar to that of styled image. To impose this style on the content image, two loss functions are used, i.e. the content loss function and the style loss function. These loss functions are nothing but the distance functions which help to determine how different the style of the target image is with respect to the style image. The key idea behind NST is to minimize

A. Shah (✉) · D. Shah · H. Shah · S. Shahane · K. Deulkar
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: adishah3103@gmail.com

D. Shah
e-mail: dhruvinshah12@gmail.com

H. Shah
e-mail: harshbs1998@gmail.com

S. Shahane
e-mail: snehashahane98@gmail.com

K. Deulkar
e-mail: khushali.deulkar@djsce.ac.in

this distance function so that the style of the target image is similar to the given custom style image. This helps to impose style [3] of any image into the content of a specific image. As a result, NST has various applications such as colouring a black and white image, image reconstruction, styling videos and styling 3D images. Many attempts are being made to implement this in various spectres of life. Though NST has not touched the fashion and the interior designing industry significantly, prior research in this field has brought a lot of computer vision challenges such as designing of clothes, designing of interiors, generating new patterns, personalized designs for users and classification of clothes according to their attributes. This paper proposes a system to bring neural style transfer to the apparel industry.

2 Literature Review

Image style transfer, which uses neural networks, is a difficult image processing task. In style transfer, the basic idea involves taking two images, a style image and a content image. The resultant image is a mix of these two images. The content image provides the basic structure on which the style and texture are superimposed. This entire process is done while preserving the content of original image. Many algorithms are implemented for image style transfer. In these algorithms, pixel resampling is done for the content image to get a photorealistic image.

According to paper [2], previous algorithms used did not produce high-quality and distortion-free target image. Therefore, the authors introduced a neural algorithm of style transfer. For optimizing object recognition and gleaning high-level image information, the authors used image representations derived from convolutional neural networks. They were successful in producing the intended output and achieved a good percentage of accuracy.

Another approach partially builds upon the work of deep photo style transfer [2]. Their deep learning approach was to deal with plenty of image contents while faithfully transferring the reference style. However, that did not work for photographs. Thus, their contribution was to limit the transformation to be affine in colourspace. Their approach successfully removed distortions and achieved good accuracy in photorealistic style transfer of a variety of scenarios like transfer of the time of day, weather, season and artistic edits.

Photorealistic image stylization [4] found that images that appear realistic before the style transfer lost the photorealistic characteristic after the style transfer. So essentially, the new image created after the style transfer appeared unreal. The resulting image does not usually look like the one taken from a camera. This issue persists again with daytime and night-time photos. Classic techniques are related to only the RGB matching and are restricted to certain scenarios like bird view shots or seasons. Various advancements are done in this field. Our main focus is to bring its application to the fashion industry keeping in mind a fashion designer. Our contribution is to create an application for a fashion designer to generate unique patterns by incorporating various standards of artistic style transfer. They implemented a twofold algorithm.

Stylization step is where the actual style transfer happens, which includes the use of various transfer techniques. This is based on whitening and colour transformation step. This might generate a result that is not smooth and consistent. To tackle this problem, the next step is introduced. The smooth step is where the transferred style is spatially distributed and makes the style transfer consistent throughout. It is based on manifold ranking algorithm. Both these steps are efficiently implemented, and the output translates better and is more preferred by a real audience.

Portrait-aware artistic style transfer [5] found that the basic goal of artistic style transfer is to transfer the style of an existing image to a different image. But the accuracy and performance of existing style transfer algorithms were not up to the mark. This is because the synthesized photo is neither styled efficiently and distorts the images foreground. Hence, by applying style transfer to foreground and background separately, the intricacies can be efficiently transferred and the image can avoid distortion. They proposed a new method for portrait style transfer where they capture the style image and apply various style transfer strategies on the background and foreground. By using semantic separation, they have divided the image into background and foreground and generate background frame for the new content image. The frame avoids spilling of the style into the background. The transfer techniques are then applied.

In the paper, artistic style transfer [1] using deep learning neural networks is pretrained to extract deep features from the content image and represent the style in second-order statistics. The generated results have a slow optimization framework for real-time application, and the network is not trained to generalize for unseen images. This drawback is overcome by using a feed-forward network for fast stylization. Adversarial networks are applied for image translation to obtain a generalization for unseen images.

Two conditional networks are set up, the generator and discriminator. The stylization network (generator) uses adaptive instance normalization (AdaIN) to combine inputs. The discriminator is trained to distinguish between generated and real image. The transformation network is set up by using the encoder–decoder architecture. The output of the encoder is the concentrated features of the convolutional layers of each block.

The decoder is trained from scratch and made symmetric to the encoder for upsampling.

3 Proposed System

Fashion industry and interior designing constantly strive for new patterns and varied abstraction to introduce a new variety of clothing in the market. Generating design patterns acts as an important part of this. Often times, designers find a pattern attractive but are not able to replicate the pattern on clothing. This module is inclined to provide a way out to such fashion designers to try and test different patterns and designs on clothing lines. This system can be advantageous to fashion designers

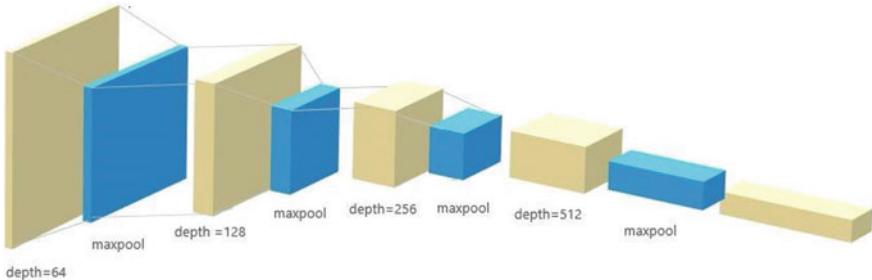


Fig. 1 VGG19 convolution neural network for style transfer

because a pre-implementation visualization is available. This essentially saves a lot of investment by designers and saves time as well. Through NST, images of fashioned clothes can be generated which are customized according to the styles selected by the user. The user can try out new abstract styles and visualize its appearance to see what best fits for the given clothing.

3.1 *Texture Synthesis*

So, the content image can be any clothing or accessory. Along with the content image, there are some style images too. A pretrained convolutional neural network like VGG16 or VGG19 (refer Fig. 1) is used which takes the content image and style image as input. First, a noisy image is generated which will be the output image. Then, the similarity of this output image is calculated with respect to content image and style image at every layer of the network. Style of image is extracted through texture synthesis by using feature mapping in convolution layers. This feature mapping is represented through Gram matrix which is a series of vector values. This Gram matrix is a dot product between the vectors of the different filters or channels in the image, i.e. intensity of RGB values. Thus, the style extracted from the image is nothing but a series of Gram matrices across all the convolutional layers. Similarly, the content of image is obtained from the higher layers of the CNN. Finally, the style loss [6] is computed using mean square error. This loss is minimized using back propagation technique to generate an artistic image from our initial random image.

3.2 *Implementation*

The user can visualize and get an insight of the fashioned clothes and designing by choosing an abstract style of his preference on the clothing. The input image of the clothing along with the styled image chosen by the user will be fed into

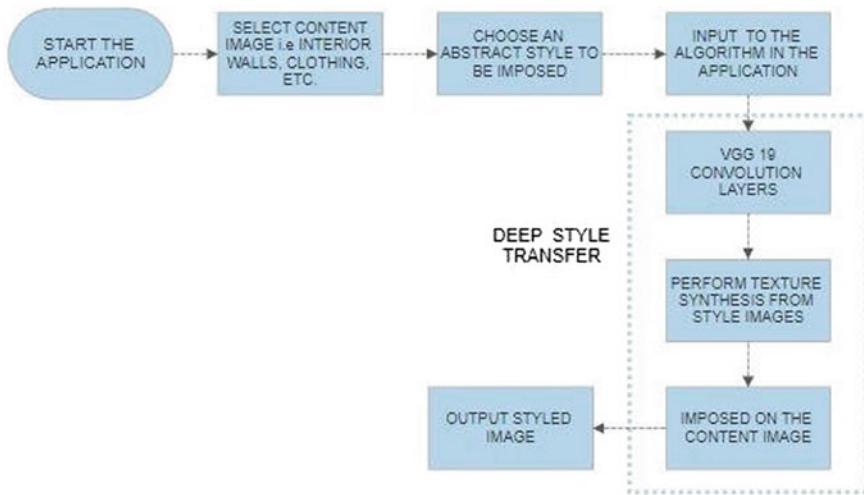


Fig. 2 System architecture for generating aesthetic images

the software system. With the help of NST, VGG19 performs texture synthesis and extracts features from the style image chosen (see Fig. 2). This style is then fused with the content image selected which can be any interior room, clothing, etc. Thus at the output, a desired artistic fused image is constructed. This way it helps to generate and visualize new aesthetic patterns which can be used in the design industry.

4 Conclusion

Neural style transfer has proved to be one of the major successes in the field of image processing and computer vision. Considering the immense potential it brings, there are still many unexplored areas for the same. It can be applied to fields such as interior designing to visualize how a room would look after applying a certain design pattern to it, generating abstract paintings on walls and floor designing. Other common applications include visualizing tattoos on skin, styling portrait selfies and picture, data augmentation, etc. So with respect to future scope, NST can be applied on some of these applications to generate more personalized images and also to improve the performance of the system.

References

1. Mounica P, Nagaratnam A, Mohammad F, Muralidhar K (2019) Artistic style transfer using deep learning. Int J Adv Res, Ideas Innov Technol 5(2). ISSN: 2454-132X

2. Luan F, Paris S, Shechtman E, Bala K (2017) Deep photo style transfer. IEEE
3. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. IEEE
4. Li Y, Liu MY, Li X, Yang MH, Kautz J (2018) A closed-form solution to photorealistic image stylization. [arXiv:1802.06474v5](https://arxiv.org/abs/1802.06474v5)
5. Xing Y, Li J, Dai T, Tang Q, Niu L, Xia S-T (2018) Portrait-aware artistic style transfer. IEEE
6. Sanakoyeu A, Kotovenko D, Lang S, Ommer B (2018) A style-aware content loss for real-time HD style transfer. [arXiv:1807.10201v2\[cs.CV\]](https://arxiv.org/abs/1807.10201v2[cs.CV])

Chapter 48

Credit Card Fraud Detection Using Meta-classifiers Consisting of Semi-supervised and Supervised Algorithms



Rutuja Taware 

1 Introduction

In this contemporary world system, people are widely adopting the “go cashless” policy. Consequently, there has been a remarkable rise in online transactions. Accompanying this rise is the increase in fraudulent transactions. Credit card frauds have resulted in tremendous loss of money and have disrupted the lives of many people.

Use of artificial intelligence in creating an efficient credit card fraud detection system faces an acute problem of extremely imbalanced datasets. In the case of credit card transaction datasets, the instances of genuine transactions greatly outnumber fraudulent transactions. To deal with a highly imbalanced dataset, sampling methods such as oversampling or undersampling can be utilized. However, in undersampling, removal of instances of the majority class results in the loss of significant characteristics of the majority class.

Another way to deal with the highly imbalanced dataset is to utilize a novelty detection approach. This approach takes instances of only the majority class as its training data and decides whether a new instance is novel or not. One-Class SVM is a novelty detection algorithm, whose training data consists of only non-fraudulent transactions, and hence, it follows a semi-supervised approach. A semi-supervised approach is useful when abundant information is available for one class, whereas a negligible amount of information is available for another data.

In the case of credit card fraud detection, a false negative (FN) may result in the loss of huge amounts of money as the transaction amount is completely lost, whereas a false positive (FP) may cause customer dissatisfaction and some administrative cost [2]. Hence, a fraud detection system must focus on detecting a maximum number of fraudulent transactions as well as customer satisfaction.

R. Taware (✉)
Pune Institute of Computer Technology, Pune, India
e-mail: rutujam77@gmail.com

2 Related Work

Research on credit card fraud detection involves various data preprocessing methods and several classification algorithms. A survey of these methods and algorithms is provided below.

In order to balance the dataset, Synthetic Minority Oversampling Technique was used in [11], followed by the application of classification algorithms like logistic regression (LR), Naive Bayes (NB), random forest (RF) and multilayer perceptron (MLP), out of which LR gave the highest recall value of 91.84% with an accuracy of 97.46%, whereas RF gave a recall value of 81.63% and an accuracy of 99.96%. An undersampling approach was taken in [8], in which RF gave the highest recall value of 92.5%. A meta-classifier was used in [7] for performing predictions in two levels, a RF classifier at level one followed by an ensemble of decision trees (DT) and gradient boosted trees (GBT) at level two. A multi-level ensemble was used in [6] which included a bagging model, and each base learner of the bagged model consisted of boosted trees. Akila [1] used a non-overlapping bagging ensemble which used NB as its base classifier; here the bags created had non-overlapping majority class instances. To reduce the risk of loss and reputation of an organization, a cost-sensitive neural network was used in [4].

3 Dataset

The dataset consists of 284,807 transactions of European cardholders out of which only 492 are fraudulent transactions. Hence, the minority class makes up only 0.172% of the entire dataset. It consists of 30 independent variables out of which 28 (V_1, V_2, \dots, V_{28}) are transformed through principal component analysis for confidentiality reasons. The remaining two features are time and amount which are not transformed. The dependent variable is the class variable which takes the value of zero for a non-fraudulent transaction and a value of one for a fraudulent transaction.

4 Metric

An efficient credit card fraud detection system should try to maximize its true positive (TP) and true negative (TN) and minimize its FP and FN. However, a FN will have a huge negative impact on the organization as it will lead to a loss of associated transaction amount [4]. As a result, it is of prime importance to identify as many frauds as possible, and hence, recall should be an appropriate metric in this case. But customer satisfaction should also be taken into consideration; hence, it is vital for a fraud detection system to have a high accuracy value.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (1)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \quad (2)$$

5 One-Class SVM Algorithm

While training a model using OCSVM as an algorithm, the training set consists of only a single class. The algorithm learns a hyper ball which encloses most of the training examples and tries to keep the volume of the hyper ball as minimum as possible. Specifically, it tries to learn a function $f(x)$ which defines the hyper ball [9]. Here, function $f(x)$ takes the following form [10]:

$$f(x) = \text{sgn} \left\{ \sum_{i=1}^n \alpha_i K(x, x_i) - \rho \right\} \quad (3)$$

where α_i are the Lagrange multipliers obtained by optimizing the following equation:

$$\min_{\alpha} \left\{ \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \right\} \quad (4)$$

Such that

$$0 \leq \alpha_i \leq \frac{1}{vn} \quad (5)$$

$$\sum_{i=1}^n \alpha_i = 1 \quad (6)$$

Here, n represents the number of instances present in the training set, while v provides an upper bound for the fraction of instances outside the hyper ball and lower bound for the fraction of support vectors. $K()$ represents the OCSVM kernel which is used for transforming the instances into a linearly separable feature space [3]. There are many kernel functions present, out of which the radial basis function (RBF) is popularly used which allows us to determine the radius of the hyper ball through the γ parameter [5].

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \quad (7)$$

An instance x is accepted if $f(x) > 0$ else it is rejected [5] (Fig. 1).

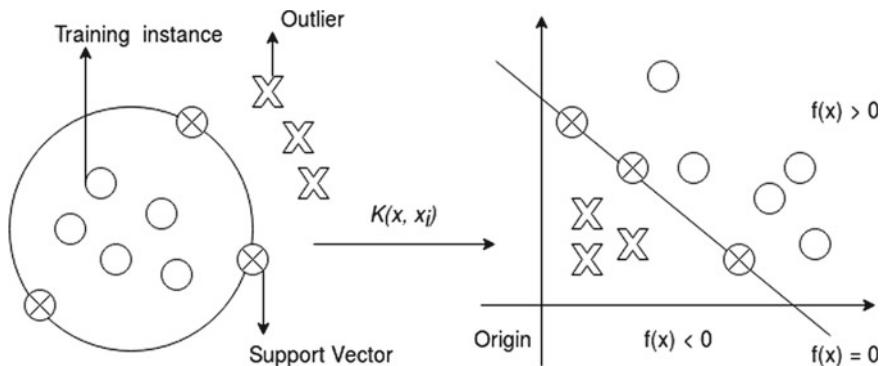


Fig. 1 Classification performed by OCSVM [5]

6 Proposed Method

This paper proposes two meta-classifiers in which prediction takes place in two phases for each classifier. For both the classifiers, prediction at level one is performed by the One-Class SVM algorithm. For level two prediction, one meta-classifier uses logistic regression (LR) while the other uses random forest (RF) as its prediction algorithm.

When a new instance arrives, OCSVM will generate a score for that instance. If the score lies in a predefined range, then the instance is sent for level two prediction else prediction of level one is considered as final. Procedure for obtaining a predefined range is explained later in this section.

Results obtained by applying the algorithms independently on the dataset are shown in Table 1. It can be observed that OCSVM performs well in finding fraudulent cases; however, it generates several false positives (FP). As a result, OCSVM gives a high recall value but its accuracy is reduced due to many FP. However, LR and RF give high accuracy but they give a low recall value. Therefore, in level two, LR and RF are used as they generate a very low number of false positives. Hence, this ensemble will help in generating a good recall and accuracy value.

Phase 1 Steps to obtain a range for data segregation (Fig. 2):

1. Split the dataset into train, test and validation set.
2. Train the model on the training set using OCSVM.
3. Obtain scores for each instance in the validation set using the decision function (3).

Table 1 Results for individual algorithms

Algorithm	OCSVM	LR	RF
Recall	93.87	91.83	84.69
Accuracy	89.98	97.68	99.95

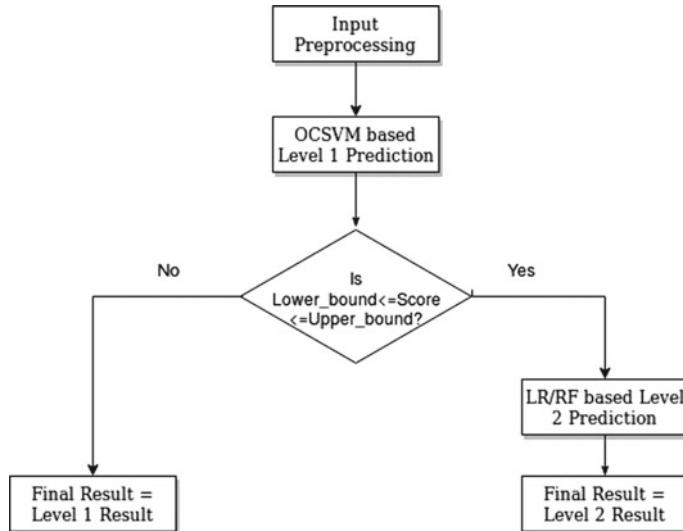


Fig. 2 Meta-classifier architecture

4. Obtain the upper bound of the range in the following manner.

Arrange the scores of predicted non-fraudulent instances in ascending order and select the first 25% samples. Find the highest score amongst the selected samples and set it as an upper bound for the range.

5. Obtain the lower bound of the range in the following manner.

Arrange the scores of predicted fraudulent instances in ascending order and select the last 50% samples. Find the lowest score amongst the selected samples and set it as the lower bound for the range.

The range is chosen such that it selects instances which are closer to the separating plane of the OCSVM. To reduce the number of false positives, it selects more samples below the threshold (Fig. 3).

Phase 2 In this phase, LR and RF are used as classification algorithms. Hence, it is essential to balance the dataset to avoid bias. As a result, Synthetic Minority

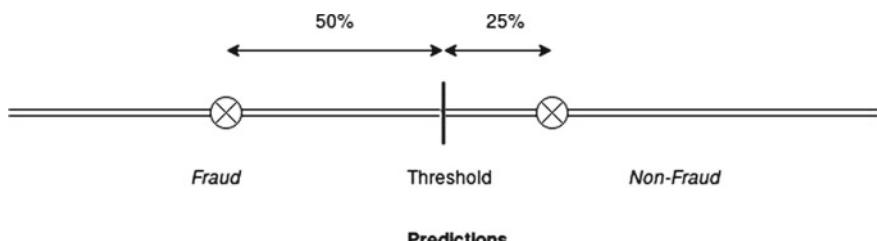


Fig. 3 Range calculation

Oversampling Technique (SMOTE) is used for balancing the dataset, and the model is trained on the training set.

7 Result

The credit card dataset was split into train, test and validation set, out of which the test set consisted of 20% of the whole dataset. These include 56,962 transactions out of which 98 were fraudulent. For balancing the dataset, SMOTE was used wherein the minority class was resampled by considering five nearest neighbours to construct synthetic samples. The results include confusion matrices for individual algorithms and meta-classifiers. Metrics like recall and accuracy have been calculated from the confusion matrices (CM) and have been mentioned in Table 2. From Table 2, it can be seen that the meta-classifiers gave a higher recall than their individual supervised algorithms and a higher accuracy value than the semi-supervised algorithm (OCSVM). As a result, the proposed method provides an intermediary solution between low recall, high accuracy, high recall and low accuracy. Hence, these meta-classifiers have enabled an increase in fraud detection without compromising customer satisfaction to a great extent.

7.1 One-Class SVM Confusion Matrix

		True Class	
		Positive	Negative
Prediction Class	Positive	92	5698
	Negative	6	51,166

Table 2 Results for meta-classifiers and individual algorithms

Algorithm	OCSVM	LR	RF	OCSVM-RF	OCSVM-LR
Recall	93.87	91.83	84.69	90.82	92.85
Accuracy	89.98	97.68	99.95	95.05	93.74

7.2 *Logistic Regression Confusion Matrix*

		True Class	
		Positive	Negative
Prediction Class	Positive	90	1340
	Negative	8	55,544

7.3 *Random Forest Confusion Matrix*

		True Class	
		Positive	Negative
Prediction Class	Positive	83	11
	Negative	15	56,853

7.4 *One-Class SVM + Random Forest Confusion Matrix*

		True Class	
		Positive	Negative
Prediction Class	Positive	89	2805
	Negative	9	54,059

7.5 One-Class SVM + Logistic Regression Confusion Matrix

		True Class	
		Positive	Negative
Prediction Class	Positive	91	3558
	Negative	7	53,306

8 Conclusion

Credit card fraud detection has been the need of the time, and hence, many organizations are investing in finding out appropriate methods to do so. This paper compares the performance of individual algorithms and two meta-classifiers. Performance is measured using metrics like recall and accuracy. High recall value is extremely important in such scenarios but completely ignoring customer satisfaction is also not appreciated. The two meta-classifiers include a semi-supervised algorithm in phase one and supervised algorithms in phase two. It was established that the meta-classifiers gave a higher recall value compared to their respective supervised algorithms and higher accuracy compared to the semi-supervised algorithm.

Further research should focus on creating ensembles which include OCSVM and other supervised algorithms along with better techniques for dataset balancing.

References

1. Akila S, Reddy US (2017) Credit card fraud detection using non-overlapped riskbased bagging ensemble (nrbe). In: 2017 IEEE international conference on computational intelligence and computing research (ICCIC), pp 1–4. <https://doi.org/10.1109/ICCIC.2017.8524418>
2. Benchaji I, Douzi S, El Ouahidi B (2018) Using genetic algorithm to improve classification of imbalanced datasets for credit card fraud detection. In: International conference on advanced information technology, services and systems. Springer, pp 220–229
3. Bergamini C, Oliveira LS, Koerich AL, Sabourin R (2009) Combining differentbiometric traits with one-class classification. Sig Process 89(11):2117–2127
4. Ghobadi F, Rohani M (2016) Cost sensitive modeling of credit card fraud using neural network strategy. In: 2016 2nd international conference of signal processing and intelligent systems (ICSPIS), pp 1–5. <https://doi.org/10.1109/ICSPIS.2016.7869880>
5. Guerbai Y, Chibani Y, Hadjadj B (2014) The effective use of the oneclass svm classifier for reduced training samples and its application to handwritten signature verification. In: 2014 international conference on multimedia computing and systems (ICMCS), pp 362–366. <https://doi.org/10.1109/ICMCS.2014.6911221>
6. Kavitha M, Suriakala M (2017) Hybrid multi-level credit card fraud detection system by bagging multiple boosted trees (bmft). In: 2017 IEEE international conference on computational

- intelligence and computing research (ICCIC), pp 1–5. <https://doi.org/10.1109/ICCIC.2017.8524161>
- 7. Kavitha M, Suriakala M (2017) Real time credit card fraud detection on huge imbalanced data using meta-classifiers. In: 2017 international conference on inventive computing and informatics (ICICI), pp 881–887. <https://doi.org/10.1109/ICICI.2017.8365263>
 - 8. Mishra A, Ghorpade C (2017) Credit card fraud detection on the skewed data using various classification and ensemble techniques. In: 2018 IEEE international students' conference on electrical, electronics and computer science (SCEECS), pp 1–5. <https://doi.org/10.1109/SCEECS.2018.8546939>
 - 9. Plamondon R, Srihari SN (2000) Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans Pattern Anal Mach Intell* 22(1):63–84. <https://doi.org/10.1109/34.824821>
 - 10. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. *Neural Comput* 13(7), 1443–1471
 - 11. Varmedja D, Karanovic M, Sladojevic S, Arsenovic M, Anderla A (2019) Creditcard fraud detection—machine learning methods. In: 2019 18th international symposium INFOTEH-JAHORINA (INFOTEH), pp 1–5. <https://doi.org/10.1109/INFOTEH.2019.8717766>

Chapter 49

Correlation Between Number of Hidden Layers and Accuracy of Artificial Neural Network



Purva Raut and Apurva Dani

1 Introduction

Artificial neural networks are relatively crude electronic networks of neurons which aims at making machines learn knowledge like a human does. The neural network takes one input at a time and then further processes and learns by comparing the desired output and the result from the neural network. The error which is calculated from the first input is fed back to the network and is used to modify the weights between the neurons. This process of reducing error correction is performed for many iterations. A neuron has two major components:

1. Input values and random weights which are associated with it.
2. Summation function $[u]$ that adds the weights together and maps it to an output $[y]$.

There are different layers in an artificial neural network which consists of neurons. The input layer is composed just of the input values and not the neurons which act as input to the next layer.

The next layer is hidden layer; there may be several of them and this paper focuses on how varying the number of hidden layers correlates with the accuracy of the model (Fig. 1).

The hidden layers take weighted inputs and the output which is given by this layer is based on an activation function. There is no fixed rule about the number of hidden layers which should be used to create the neural network.

P. Raut · A. Dani (✉)

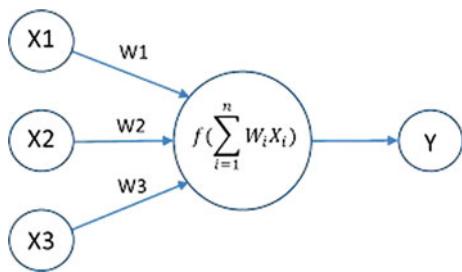
Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, India

e-mail: apurvadani@gmail.com

P. Raut

e-mail: purvapraut@gmail.com

Fig. 1 Structure of a simple neural network



2 Hidden Layer and Its Working

For training a neural network, the following steps are performed in a loop so that the weights of each input to the hidden neurons can be adjusted to get the least error possible:

1. In first step, forward propagation is implemented.
2. In second step, the loss is computed.
3. In third step, backward propagation is implemented to get the parameters to adjust the weights.
4. In step four, the parameters are updated to reduce the error.
5. In the final step, forward propagation is implemented.

In the first step, generally, the hidden layers use ReLU activation function and the output layer uses sigmoid activation function.

The forward propagation is computed using the following equations:

- Computation at first layer of activation:

$$Y^{[1]} = W^{[1]}X + b^{[1]}, A^{[1]} = \text{ReLU}(Y^{[1]})$$

- Computation at nth activation layer:

$$Y^{[n]} = W^{[n]}A^{[n-1]} + b^{[1]}, A^{[n]} = \text{ReLU}(Y^{[n]})$$

- Computation at last activation layer:

$$Y^{[L]} = W^{[L]}A^{[L-1]} + b^{[1]}, A^{[n]} = \text{sigmoid}(Y^{[L]})$$

- Computation of loss function:

$$\frac{-1}{m} \sum_{i=1}^m ((y^i) \log(a^{[L](i)}) + (1 - y^i) \log(1 - a^{[L](i)}))$$

After implementing forward propagation, backward propagation is calculated using the following steps:

- First, we perform linear backward propagation.
- After that linear to activation backward where the derivative of ReLU or sigmoid activation is computed.
- [linear to ReLU] $X(N - 1)$ to linear to sigmoid backward (entire model).

After completion of all the above-mentioned steps, we use gradient descent to update the parameters.

3 Procedure

In this paper, an artificial neural network [1] was trained for four different datasets and for each model, the number of hidden layers was varied to see the effect of the number of layers on the accuracy of the model. While changing the number of layers, all the other factors such as number of neurons for a level, activation function, and other variables were kept constant. After training each model, the same data was tested for all neural networks for a dataset. Then, for every dataset, a graph was plotted which visualizes accuracy [2] for a different number of hidden layers on a given dataset.

4 Dataset

There are four different datasets used for the following experiment. The range of number of rows varies from 1,000 to 10,000 and range of number of columns varies from 4 to 8. The datasets are further divided into 25:75 for training [3] and testing, respectively. Accuracy is used as the performance measure of the neural network in this experiment. The accuracy is defined as the number of predictions of testing set that is correct to that of the total number of cases that are used for testing.

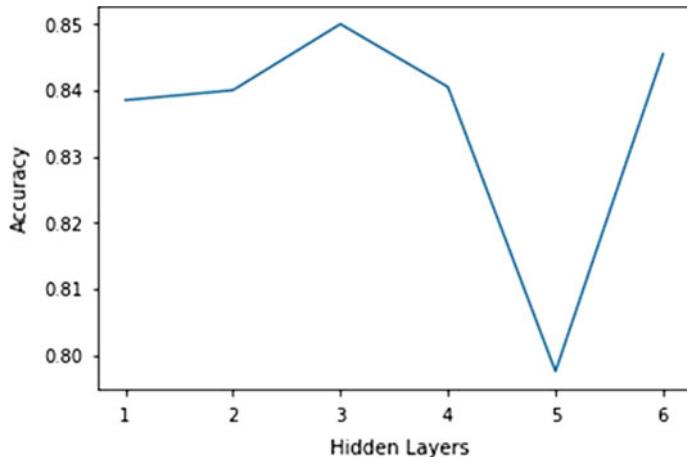


Fig. 2 Accuracy of dataset 1 with varying number of hidden layers

5 Results

The following graphs are for four different datasets where X -axis shows the number of hidden layers, whereas Y -axis shows the accuracy attained. Along with hidden layers, there is an output level which is not considered while plotting the graph.

5.1 Dataset 1

The graph below shows the accuracy of dataset 1 with number of hidden layers increased from 1 to 6 (Fig. 2).

5.2 Dataset 2

See Fig. 3.

5.3 Dataset 3

See Fig. 4.

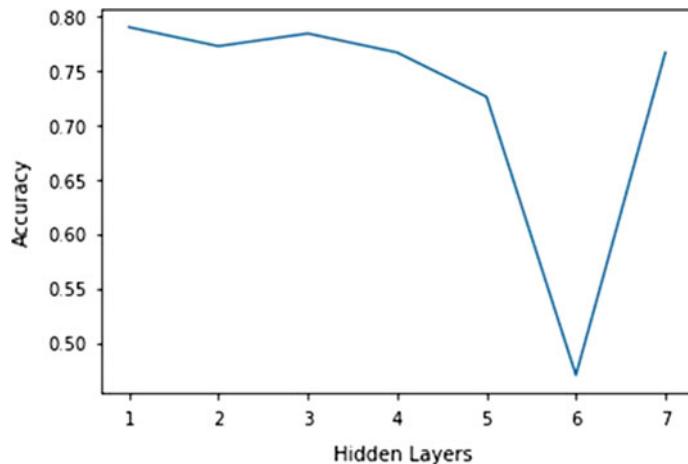


Fig. 3 Accuracy of dataset 2 with varying number of hidden layers

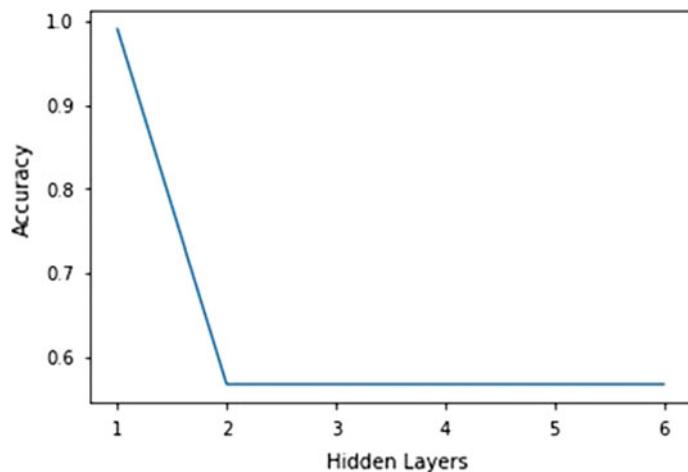


Fig. 4 Accuracy of dataset 3 with varying number of hidden layers

5.4 Dataset 4

See Fig. 5.

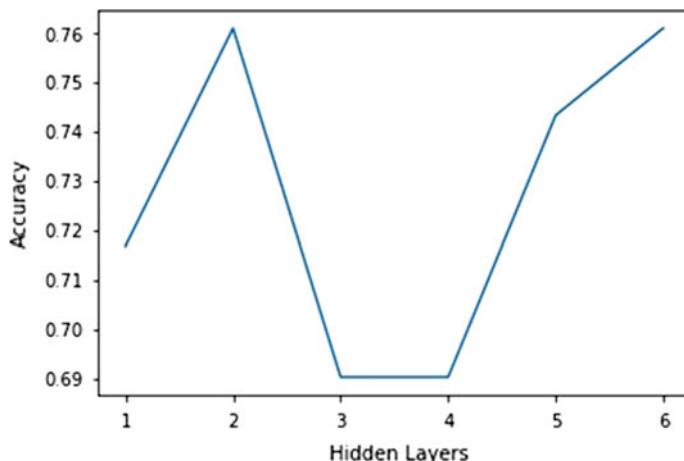


Fig. 5 Accuracy of dataset 4 with varying number of hidden layers

6 Observations

The graph above explains how accuracy varies with variation in the number of hidden layers. It can be observed that initially the model's accuracy increases gradually for certain number of layers and then drops abruptly after reaching a saturation point.

In dataset 1, accuracy starts from around 0.8385, reaches its maximum when there are three hidden layers. All the graphs can be summarized in a similar way, and Table 1 gives an insight about the maximum accuracy and hidden layers.

Theoretically, if appropriate number of neurons is selected for the first hidden layer of the neural network then it can fit most of the hypothesis and hardly there is a need to add more hidden layers for the network.

Table 1 Hidden layer at which maximum accuracy was found

Dataset	Accuracy	Hidden layers
1	0.85	3
2	0.7907	1
3	0.99	1
4	0.761	2

A function that has a continuous mapping from one finite space to another can be approximated with the help of only one hidden layer.

However, one hidden layer can approximate any function that contains a continuous mapping from one finite space to another.

An arbitrary decision boundary to arbitrary accuracy with rational activation can be represented using two hidden layers. It can also be used to approximate smooth mapping to any accuracy.

When you do not use any hidden layer, the network can only be used to represent functions which are linearly separable.

So, from the above observations, it is clear that accuracy can be improved by increasing the number of hidden layers from 2 to 3 or 1 to 2 or 0 to 1 for small datasets.

7 Correlation Between Accuracy and Number of Hidden Layers

If the number of hidden layers which are used to build the network are much more than what is required for the given dataset, the accuracy of the test set will decrease. Such networks will overfit the training data, that is, it will perfectly learn the data which is given for training, but it will fail to generalize for the test data.

Figure 6 reflects the problem of underfitting and overfitting. Here, we have a set of data points and we will try to fit the best function we can to fit the data.

In the first figure, we are trying to fit a linear function to fit all the data points. As we can see that the function is not complex enough to fit all the data points and it suffers from the problem of underfitting. In the second figure, we try to generalize the data with a more complex function. It can be seen that the model has learned the trend that the points in the data follows which is a parabola. In the last figure, we have increased the number of hidden layers more than the model required and we can see that it suffers from the problem of overfitting. That is it could not learn

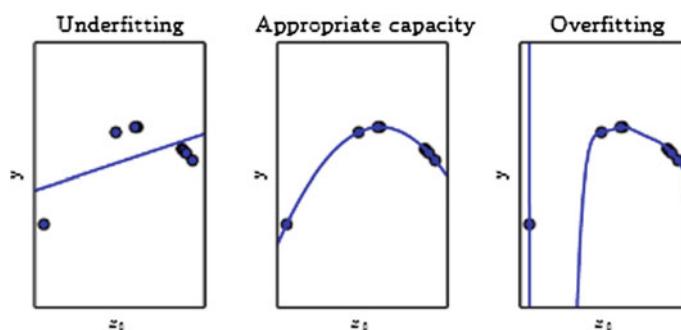


Fig. 6 Simulating the model graph with increasing number of hidden layers [4]

what the trend was and thus it fails to correctly predict the results of test data. Thus, by increasing the number of hidden layers in the neural network, the model fails to generalize the trend to the new data. Thus, it gives poor accuracy with the testing set and this can be reflected in the result section of this paper.

8 Conclusion

In most of the practical cases, where we are using small dataset, there is no need of having more than two hidden layers for getting a real good accuracy. Increasing the number of hidden layers will reduce the accuracy of the model since the backpropagation algorithm loses its effectiveness.

When you increase the number of hidden layers in the neural network, the error that you will get while using the model to predict test dataset will increase, even though the model was correctly predicting for the training set due to overfitting.

The accuracy of the model depends upon the performance of the architecture of the network and the algorithm used in its test dataset.

When a network tries to fit the data very closely, it will have a huge generalization error and a very high variance because of overfitting.

Thus, to decrease this variance, we need to smooth the network outputs, but while reducing the variance, the bias may increase to a huge value and the error in generalization will be large again. This is the case of underfitting. Thus, the balance between the bias and variance plays a huge role in applying neural network to practical applications.

Following solutions can be used to avoid the problem of underfitting:

1. There should be enough number of hidden nodes in the network so that the function properly fits the dataset. It should be capable enough to represent the mappings of the data points.
2. To reduce the cost of sum squared error, the network should be trained for enough amount of time.

To prevent overfitting:

1. The network should not be trained for extremely long time that it does not learn the trend and it only fits the training data.
2. The adjustable parameters like number of hidden layers of the network must be restricted so that the chance of overfitting reduces.

References

1. Da Silva IN et al (2017) Artificial neural networks. Springer International Publishing, Cham
2. Rauber PE et al (2016) Visualizing the hidden activity of artificial neural networks. *IEEE Trans Visualization Comput Graph* 23(1):101–110
3. Günther Frauke, Fritsch Stefan (2010) Neuralnet: training of neural networks. *R J* 2(1):30–38
4. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Chapter 50

Face Completion Using Generative Adversarial Network



Purva Raut, Moxa Doshi, Monil Diwan and Karan Doshi

1 Introduction

In crime investigation sector, recognition of the suspect is difficult due to the lack of proper pictorial evidence. At times, it becomes difficult to recognize the person when only the partial face is available as evidence. In such cases, there is absolutely no technology available to reconstruct the entire face. This drawback creates an obstruction in the investigation process. Through this project, we will try to provide a solution to this extremely critical problem. Partial part of the suspect, which is available as evidence would be fed into the system and the system will generate the entire face hence helping the investigation process. In this scenario, it is extremely difficult for an algorithm to predict various peculiarities of the face like a mole, beard, eyes shape, etc. To solve this problem, we introduce an option of providing a textual description of the suspect's image. The description would be accepted from the user by high-level description, for example, "French beard," "wavy hair," etc. and after a dual combination of both the modules, viz. face completion and text-to-feature conversion, a final output of the entire face is generated. To achieve this, we use a technique called image inpainting [1].

Image inpainting is a technique which aims to fill the missing regions of an image with plausibly synthesized contents. Inpainting is achieved by generative adversarial

P. Raut (✉) · M. Doshi · M. Diwan · K. Doshi
Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, India
e-mail: purvapraut@gmail.com

M. Doshi
e-mail: moxadoshi1610@gmail.com

M. Diwan
e-mail: monil0206@gmail.com

K. Doshi
e-mail: karandoshi98@gmail.com

networks (GANs). GAN, as defined by Goodfellow in [2], is a combination of two neural networks, generator and discriminator. He further states that the generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. We prolonged this idea by employing a generator network which takes a masked image as input and inpaints the missing region. This image is then forwarded to the discriminator to distinguish it as real or fake and backpropagates the feedback to the generator. This process eventually allows the generator to generate plausibly realistic images.

Various systems have been designed using generative adversarial networks to achieve significant results in image inpainting. Also, there are other algorithms which are used to implement image inpainting [16]. All these systems and algorithms with their features and drawbacks are discussed in the subsequent section.

2 Literature Survey

In this section, we provide a detailed analysis of different system and algorithms used for face generation. The first step which is required in order to implement face inpainting is the detection of partial face from the available image. Existing method of face detection includes using OpenCV [3] which uses Haar Cascade algorithm. Haar Cascade is extracting features from images using a kind of “filter,” similar to the concept of the convolutional kernel. This algorithm applies the model to an image at multiple locations and scales. High-scoring regions of the image are considered detection. However, this model is unable to detect unaligned faces. YOLO [4] algorithm overcomes this drawback. YOLO applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region.

Generative Image Completion

Image inpainting in a nutshell [1] is a way of generating the missing region of a given image which although synthesized looks real. Interesting work has been done in the field of image inpainting. When the system is fed with masked as well as actual images it identifies the relationship between the masked region and its unmasked counterparts. This approach has been implemented with variety of techniques. Convolutional neural network (CNN) and recurrent neural network (RNN) [5] are used. This uses CNN with Sigmoid Euclidean Loss and a simplified PixelRNN on images of size 32×32 from CIFAR-10 dataset. The objects need to be symmetric for this technology to work, and it will fail for images that are not symmetric. Also, it cannot handle unaligned images. After the development of GAN and variational auto-encoder [2, 6, 7] which produces better results with superior understanding of the image in a holistic manner, previous methods seem obsolete. Generative adversarial networks (GANs) have proved to eliminate the issues faced while using CNN and RNN.

Existing work includes image completion of symmetric objects like flowers, birds, etc. [5]. This technique utilizes the pixels in the visible part of the image to complete the image. Adobe Photoshop's Content-Aware Fill is similar to inpainting which uses the surrounding pixels to fill the missing area. GANs have been utilized in applications like generation of anime, inpainting historical monuments, for generating random human faces, etc. On similar lines, GANs can be implemented for face inpainting [8, 9]. This approach gives the best accuracy among the available systems. However, it fails to address unique peculiarities that an individual's face might have. To solve this, using a text-to-facial features synthesis is a favorable idea [10].

Text-to-Image Synthesis

Effective approach for text-based image synthesis using a character-level text encoder and class-conditional GAN [7, 10]. The purpose of the GAN is to view (text, image) pairs as joint observations and train the discriminator to judge pairs as real or fake. The input from the user comprises high-level descriptions. These are usually natural languages, but what underlies its corresponding images are essentially a group of visual attributes that are extracted from a sentence. Generating photo-realistic images from text description is not easy and needs to be developed with appropriate parameterization so that it does not produce nonsensical output. LSTM networks are commonly used in natural language processing and hence a better way to understand the high-level language input that the user provides to generate the image is to use LSTM which is capable of learning the semantic meaning of sentences and relationships between the words of a given sentence [11]. Along with LSTM, a GAN model implemented on the features extracted using LSTM generates realistic images. A deep architecture with GAN formulation was proposed in which a deep convolutional generative adversarial network (DC-GAN) conditioned on text features [10] encoded by a hybrid character-level convolutional recurrent neural network was used. Both the generator network G and the discriminator network D perform feed-forward inference conditioned on the text feature. The drawback of the model is that, on increasing the resolution for generated images, it gets unstable because of upsampling and hence cannot generate images beyond a certain resolution. Another implementation which improves on the basic GAN implementation is StackGAN [12]. This model uses two GAN networks. The first GAN sketches the primitive shape and colors of the object based on the given text description, yielding low-resolution images. The second GAN takes low-resolution results and text descriptions as inputs and generates high-resolution images with photo-realistic details. Existing work based on this concept includes generation of symmetric objects like flowers and birds, and the generation of human faces has been implemented which gives extremely blur images [13].

In our research work, we are addressing some of the issues mentioned above and propose a novel approach for face inpainting along with textual description. Let us see our approach in the following section.

3 Our Approach

We propose using two stages of GAN, viz. Stage 1 GAN and Stage 2 GAN [12]. First stage of GAN is used for image inpainting which comprises a generator and a discriminator. However, it is difficult or even impossible to model the relation between the generated face and its peculiar features. Therefore, we introduce Stage 2 GAN. This stage consists of one generator which acts as an encoder, two other generators which act as decoders, a classifier and a discriminator. Intricate details of the approach have been discussed below.

3.1 Dataset

We evaluate the proposed model on CelebA dataset, which contains two hundred thousand images. We follow a standard split in which 80%, i.e., 162,079 images, are used for training, 10%, i.e., 20,259 images, are used for validation and the remaining 10%, i.e., 20,259 images, are used for testing.

For generative image completion, masked image as well as corresponding unmasked image is needed. CelebA dataset only provides unmasked images, and hence, we were obliged to create a custom dataset of masked images. This was done by using a pillow library in Python which overlays one image on another. Hence, we used the original image as background and added an overlay, i.e., foreground which was a random-shaped image which covered different regions of the face. Covering different regions ensures versatility and better learning by the model.

For text-to-image synthesis, a list of attributes corresponding to each image is required which CelebA dataset provides. Out of 40 available attributes, only a few attributes have visual impact on the image, and hence, every attribute is meticulously chosen. Examples of few of these attributes are “Blond Hair,” “Brown Hair,” “Bushy Eyebrows,” “Eyeglasses,” “Gender,” “Mouth Open,” “Mustache,” etc.

3.2 System Architecture

The aim of the proposed architecture is to recreate the masked regions of the image. The proposed model consists of image preprocessing unit and two stages of generative adversarial networks. The goal of the model is to detect the presence of a person’s face in the input image and pass on the coordinates of the face along with image to the first stage of GAN which will then regenerate the missing regions of the face. The second stage will further improve the output and imbibe the features given through textual description.

3.2.1 Image preprocessing Unit

This unit consists of face detection implemented using YOLO algorithm [4] to detect faces present in the input images. Unlike most common applications of YOLO which is object detection in self-driving vehicles, we have trained this model to detect faces in an input image. The algorithm returns bounding boxes around the detected face along with the coordinates of the boxes. This algorithm has provided exceptional results in detecting faces when a part of the face was masked.

3.2.2 Generative Adversarial Networks

We are using two stages of GAN [2, 12] to achieve the desired results. First stage of GAN uses convolutional neural network as a generator network to generate image from its masked representation, and the second unit of GAN uses an encoder based on GAN to understand the textual description and apply it on the image generated by the first unit.

Stage 1 GAN

The generator network of Stage 1 begins with a dense layer that takes the masked image as input. This layer applies leaky ReLU activation on the input image. The desired output of the generator is an image of dimension $28 \times 28 \times 3$. To achieve this output, three convolutional layers with 128, 64 and 3 filters, respectively, of dimension 5×5 each and using same padding are applied. This generates the desired output image of dimension $28 \times 28 \times 3$. After each convolutional layer, we use a leaky ReLU activation to activate the inputs received from the previous layer. This generated image is passed on the discriminator of first stage [14].

The discriminator of first unit begins with a convolutional layer with 64 filters of size 5×5 and same padding. Leaky ReLU is again used for activation. This layer is followed by another Conv-ReLU pair with 128 filters of dimension 5×5 . This is then connected to a dense layer consisting of only one output unit to classify whether the generated image is real or fake.

Stage 2 GAN

This module comprises two basic subnetworks, i.e., encoder (G_{enc})–decoder (G_{dec}) along with classifier (C)–discriminator (D) [15]. This module will map the textual description of the facial features of the generated image. Stage 2 GAN removes the strict attribute independent constraint from the latent representation and just applies the attribute classification constraint to the generated image to assure the correct change of attribute. The encoder G_{enc} which takes an image as an input and outputs a vector is a stack of convolution layers. Out of the two decoders, one decoder G_{dec} converts the vector and original attributes back into the image. Reconstruction loss is calculated at this stage which helps us understand the generation quality of the generator network. The other decoder G_{dec} converts the vector and desired attributes to the required image. The decoder G_{dec} is a stack of transposed convolutional layers. The

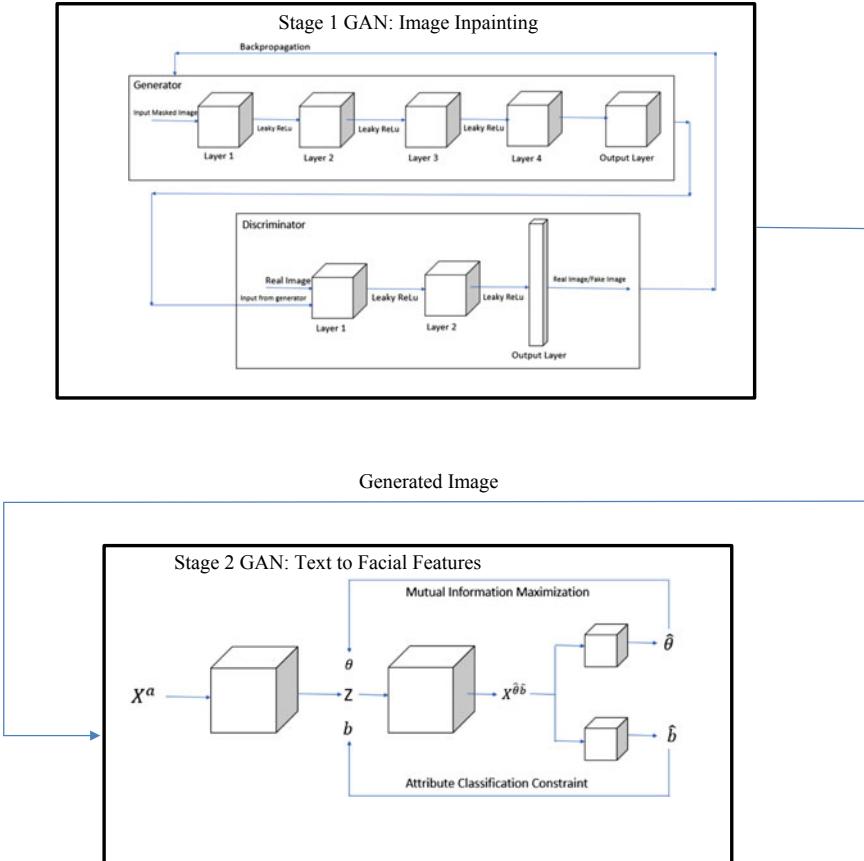


Fig. 1 System architecture

required image is then passed through a discriminator and a classifier. Discriminator is followed by a fully connected layer which calculates the adversarial loss needed to improve the model. Classifier on the other hand classifies the generated image as real or fake. Along with addressing the peculiarities of the face, this approach will also handle minor errors in semantic representation of stage 1 GAN (Fig. 1).

3.3 *Objective Function*

The objective function of our model would be to use the adversarial loss function [2]. This function is a measure of how well the generator can fool the discriminator and how well the discriminator is able to classify real and fake images. It is defined as

$$L = \min_G \max_D E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [1 - \log(1 - D(G(z)))]$$

where $p_{\text{data}}(x)$ represents the distribution of real data x , and $p_z(z)$ represents distribution of noise variable z . The generator will aim to minimize the loss by generating images which can be classified as real images, whereas the discriminator will aim to maximize the loss by correctly classifying the image as fake or real. Further, Adam optimizing algorithm is used to assist gradients to descent quickly and speed up the learning process [17].

4 Results

We have implemented the model in the most plausible manner by trying to address the maximum drawbacks of the work done in this domain. Starting with human face detection in the given image, detecting masked as well as unaligned faces was the drawback in the existing systems. We overcame it by using YOLO algorithm which is able to detect masked as well as unaligned faces by generating a bounding box around the face. Two stages of GANs are used sequentially. Output of first GAN module is fed to the second GAN module in order to achieve the final output.

Stage 1 GAN is dedicated to the task of face inpainting. Masked images are given to it as input, and it plausibly generates the completed image. Here, there is a lack of accuracy when the facial features that are unique to each individual are considered. Therefore, output of Stage 1 GAN is fed to the Stage 2 GAN. Peculiar facial features as well as errors in Stage 1 GAN can be handled here.

5 Conclusion

In this paper, we proposed a concept of using two generative adversarial networks in order to implement face inpainting. The proposed model can successfully synthesize semantically valid and visually plausible contents for the missing facial key parts from random noise. Also, with the help of the textual description, it was possible to increase the accuracy of the image that is inpainted. Our model is found to give better performance than many of the existing models related to this domain. Hence, we hope this model will prove to be of significant help to the crime investigation department (Fig. 2).

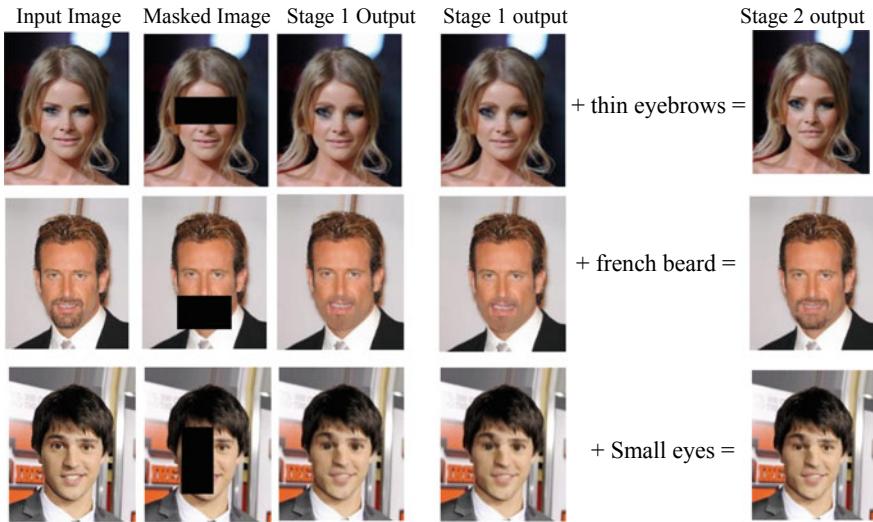


Fig. 2 Results

References

1. Wexler Y, Shechtman E, Irani M (2007) Space-time video completion. In: Journal preprint from TPAMI
2. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
3. Heisele B, Serre T, Poggio T (2006) A component-based framework for face detection and identification. Int J Comput Vision 74:167
4. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition
5. Gupta K, Kazi S, Kong T (2016) DeepPaint: a tool for image inpainting. At Standford
6. Teterwak P, Sarna A, Krishnan D, Maschinot A, Belanger D, Liu C, TW (2019) Freeman. boundless: generative adversarial networks for image extension. In arXiv preprint [arXiv:1908.07007v1](https://arxiv.org/abs/1908.07007v1)
7. Mirza M, Osindero S (2014) Conditional generative adversarial nets. In arXiv preprint [arXiv: 1411.1784](https://arxiv.org/abs/1411.1784)
8. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: IEEE international conference on computer vision (ICCV)
9. Li Y, Liu S, Yang J, Yang M-H (2017) Generative face completion. In arXiv preprint [arXiv: 1704.05838](https://arxiv.org/abs/1704.05838)
10. Yan X, Yang J, Sohn K, Lee H (2016) Attribute2Image: conditional image generation from visual attributes. In arXiv preprint [arXiv:1512.00570v2](https://arxiv.org/abs/1512.00570v2)
11. Ouyang X, Zhang X, Ma D, Agam G (2018) Generating image sequence from description with LSTM conditional GAN. In: 2018 24th international conference on pattern recognition
12. Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas D (2017) StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: IEEE international conference on computer vision
13. Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis. In arXiv preprint [arXiv:1605.05396v2](https://arxiv.org/abs/1605.05396v2)

14. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al. Tensorflow: A system for large-scale machine learning
15. He Z, Zuo W, Kan M, Shan S, Chen X (2019) AttGAN: facial attribute editing by only changing what you want. In: IEEE transactions on image processing
16. Elharroussa O, Almaadeeda N, Al-Maadeeda S, Akbaria Y (2019) Image inpainting: a review. In arXiv preprint [arXiv:1909.06399](https://arxiv.org/abs/1909.06399)
17. Kingma D, Ba J (2015) Adam: a method for stochastic optimization. In: International conference on learning representations (ICLR)

Chapter 51

Novel Approach of Computing Optimal Placement of Solar Panel Using Augmented Reality



Krupalu Mehta, Avani Sakhapara, Dipti Pawade and Vivek Surve

1 Introduction

The electricity demand is increasing day by day. Earlier we were completely dependent on the thermal and hydroelectric power stations to fulfill the domestic as well as industrial electricity needs. But due to excessive increase in the electricity demand and looking at the constraints like exhausted coal mines and insufficient water availability due to draught, now people are looking toward the renewable energy resources. In tropical country like India where ample sunlight is available throughout the year, solar energy can be a good alternative to fulfill the electricity needs. From the past few decades, many social organizations as well as government are promoting and encouraging the use of solar energy. Earlier the high cost of solar panel was the main reason behind the resistance of using it. But in past few years the costs of solar panels have been decreased to a great extent and now they are available at the affordable price. Another factor which contributes to the cost of solar energy generation is the solar panel installation cost. The average modeling time for installation of solar panels is between 1 and 2 weeks. Some of the factors that are considered while installing solar panels on rooftops are:

K. Mehta · V. Surve
Parallax Labs LLP, Mumbai, India
e-mail: krupalu@parallax.co.in

V. Surve
e-mail: vivek@parallax.co.in

A. Sakhapara (✉) · D. Pawade
Department of IT, K. J. Somaiya College of Engineering, Mumbai, India
e-mail: avanisakhapara@somaiya.edu

D. Pawade
e-mail: diptipawade@somaiya.edu

1. Earth's position toward sun (Vernal Equinox and Autumnal Equinox)
2. Size of solar panel array
3. Surface area and inclination of roof
4. Shadow analysis
5. Inclination and orientation of each solar panel.

Analysis of these factors helps in efficient prediction of amount of energy that can be produced. For this process, the technicians have to revisit the given location several times to perform required onsite analysis which indeed increases the labor cost. Also, a lack of system which can measure the efficiency of solar panels with the unavoidable change of these factors makes a crucial impact on the solar panel installation process. Thus even though the cost of solar panel is low, the high installation cost contributes to the total price.

Thus in order to provide cost-effective solution to this problem, we have developed an application which will assist in solar panel installation process which in turn will reduce the cost. The use of augmented reality (AR) technology [1] can help the technician to scan the structure of rooftops in a single visit through which real-time analysis of the onsite location can be performed and optimal placement structure view of the solar panels can be predicted. The AR view makes the site more recognizable, viewable and thus facilitates all the measurement purposes. The application uses AR technologies to detect the client location structure and orientation to perform shadow analysis on the captured location, thus finding the optimal placement of the solar panels on the client location to maximize the solar energy captured by the solar panels. It also uses the GPS location to suggest the inclination of solar panels, thus facilitating the solar panel installer, designer, developers and users' work.

2 Background Work

A lot of existing AR-based android applications are available in the market so a comparative study of all helps in determining the scope of our solution. Table 1 consists of the existing AR-based android application which has features such as location scanning, plane detection, distance calculation and model placement and all of them are getting included in our application SolAR Assistant. Just A Line application lets you draw a line in augmented reality and those augmented lines stay at the same place even if we move the camera. Similarly, the SolAR Assistant application also augments planes, walls and edges which stay at the same place, so that the augmented solar panels get placed at the right places. Measure application lets you measure real-life objects using augmented reality using your camera. Similarly, the SolAR Assistant application can be used to measure the dimensions of the location where the solar panels are to be fitted using augmented reality and back camera. ARRuler is similar in functionality with Measure application but along with it, the application also measures the area and volume of the marked coordinates. Augment application allows you to visualize 3D models in augmented reality integrated at real

Table 1 Comparison of features of existing AR-based android applications

	Just A line [2]	Measure [3]	ARRuler [4]	Augment [5]	SolAR assistant (proposed)
Scan location	✓	✓	✓	✓	✓
Plane detection	X	✓	✓	✓	✓
Distance calculation	X	✓	✓	X	✓
Model placement	X	X	X	✓	✓
Purpose	Allows to draw lines in augmented reality, and then share the creation with a short video	Detects plane surfaces and gives the measurement between any two points on the surface in terms of inches	Detects plane surfaces and allows to mark the region by connecting two points and calculates area and volume in square meter	On tapping, it displays 3D models in the augmented view	Scans the plane surfaces, permits to mark the region by marking endpoints. Performs shadow analysis and produces AR view with optimal placement of solar panels

time in their actual size. Similarly, the SolAR Assistant application helps the solar panel installers visualize solar panel installation by augmenting solar panels at the scanned location in their actual size and at optimal places.

There are many researchers working in the area of placement of solar panel. For BIPV and PIPV system [6], the power generated by solar panels installed on virtual environment was calculated using simulation of sun path and shadow analysis. For this purpose, shadow texture of PV cells and diffuse irradiance are considered as the input parameter. Author [7] put forth a system that provides user immediate feedback in the form of solar irradiance map and solar energy calculator based upon parameters like angle of inclination of panels, latitude and panel efficiency. Author [8] demonstrated the use of computer aids for providing CAEE version solar irradiance world map using mobile sensor based on latitude of location, angular orientation. Author [9] discussed a methodology that provides estimates of photovoltaic (PV) panel optimal tilt angles for all countries worldwide using country, representative city, nearest meteorological station, latitude, longitude and optimal tilt. Author [10] has developed a sun path observation system for applications in astronomy education. Here the position of sun is determined based on the orientation and elevation of

Table 2 Summary of literature review

Reference No.	Year	Scope	Methodology	Input parameter
[6]	2015	To generate power output using shadow analysis in a simulation	Render a texture for each cell, calculate shadow fraction and view factor to generate power output	Shadow texture of PV cells diffuse irradiance
[7]	2015	To provide solar irradiance map and solar power energy calculator	Give immediate feedback to user based on parameters provided and a standard equation	Angle of inclination of panels, latitude, panel efficiency
[8]	2014	Use of computer aids for enhanced and interactive delivery of education	Use of CAEE version solar irradiance world map using mobile sensor	Latitude of location, angular orientation
[9]	2017	Provides estimates of photovoltaic (PV) panel optimal tilt angles for all countries worldwide	PVWatts is used to estimate annually averaged solar output in all countries of the world assuming tilted panels	Country, representative city, nearest meteorological station, latitude, longitude, optimal tilt
[10]	2018	To develop a sun path observation system for applications in astronomy education	Calculation of position of virtual sun and sun path based on longitude, latitude and mobile device's orientation	Orientation and elevation of sun, observation date, time, longitude, latitude, mobile device's orientation and elevation

sun, observation date, time, longitude, latitude, mobile device's orientation and elevation, while sun path is determined based on longitude, latitude and mobile device's orientation. The summary of research work carried out for placement of solar panel is presented in Table 2.

3 Implementation Overview

Figure 1 depicts the application architecture that would be required for each module of SolAR Assistant. The application SolAR Assistant is built on Unity3D [11], cross-platform engine used to build and deploy application in android. Camera plugins uses the android phones rear camera to give real-time surrounding view on the mobile screen with augmented 3D models on application front end of android UI. Unity3D provides ARCore SDK [12] for AR technology which facilitates the detection of

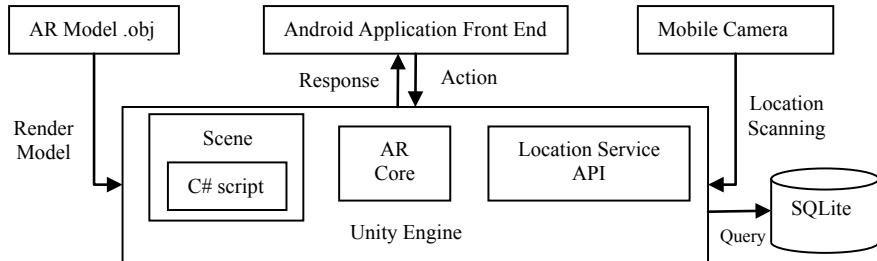


Fig. 1 Architecture of SolAR application

surfaces and objects. It uses C# as scripting languages and each scanned location is altered in terms of unity scene. Location API gets the latitude and longitude from the GPS module of android phone which is utilized for determining appropriate position of sun during simulation. 3D models of solar panels are placed on the detected planar surfaces with the help of the optimal placement algorithm mentioned in the shadow analysis module. The android app facilitates the usability by being a lightweight, gesture controlled and rapid updates along with the accurate augmented view.

Figure 2 depicts the working of the whole SolAR application where the process starts from scanning the location to the optimal placement of solar panels in augmented view of scanned location.

The entire process is divided into the following steps:

Step 1: Start Application:

Starts the scanning of the location with a button click and the rear camera starts by instantiating ARCore module.

Step 2: GPS Location Detection:

The Location API plugin gets the latitude and longitude from the Phone GPS that could be utilized for calculating the location of sun for simulation as shown in Fig. 3.

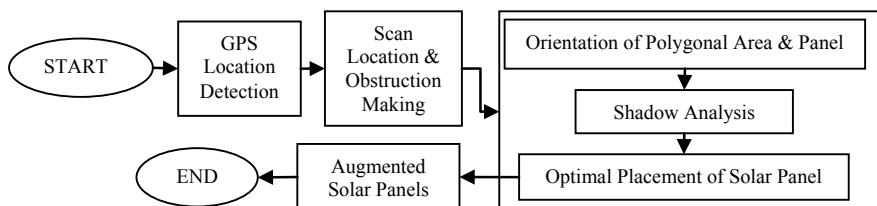
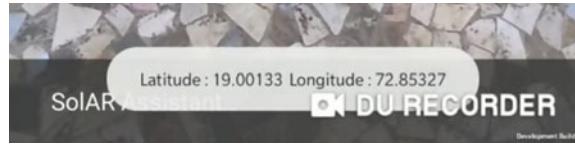


Fig. 2 Working of SolAR application

Fig. 3 GPS location detection



Step 3: Scan Location and Mark Obstructions:

1. The location is scanned through rear camera and creates a 3D planar mesh in the unity scene at which solar panel is to be installed.
2. Marks the edge vertices (vertical cylinder objects) in the same scene with the tap on the mobile screen along with boundary walls if required in a polygonal shape.
3. Scale the boundary walls and obstacles of unity meshes are scalable with an pinch on the screen to real-life sized obstructions.
4. End the scanning by clicking the button.

Step 4: Calculate Orientation of Panels:

Generally, the solar panels have a fixed orientation and facing, i.e., like South-West and inter-planar distance must remain constant for whole panel array.

Step 5: Perform Shadow Analysis:

The whole scanned unity scene is scrutinized to find optimal possible area that gets maximum intensity of sunlight. The scanned unity scene plane is first divided into a 2D matrix. Sun's position for the surface throughout the year, starting from December 21 to June 21, is simulated for the duration of 9 am to 3 pm. A normal ray from the sun to the plane is drawn for this duration to measure the intensity of sun rays on to the individual surface grid of matrix. Based on the results of this simulation, heatmap is generated for the given surface plane. Heatmap is represented in the form of 2D array consisting of hit point intensity of sunlight at each cell of scanned location by simulating the sun. The procedure of shadow analysis is demonstrated in Fig. 4.

Step 6: Calculate Optimal Placement:

The scrutinized heatmap area is evaluated for the best possible placement of solar panel array with the help of the below-specified algorithm.

PseudoCode for Optimal Placement HeatMap Algorithm:

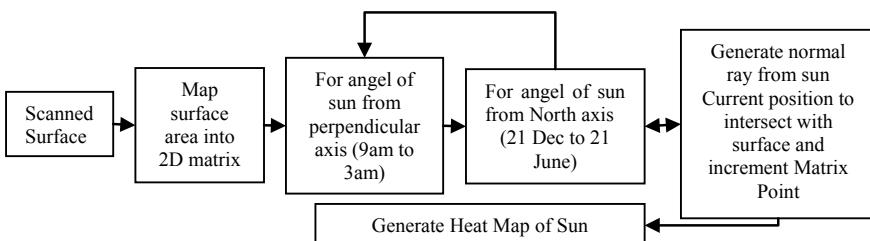


Fig. 4 Shadow analysis approach

1. Place the irregular scanned polygon inside a rectangular polygon
 - a. Calculate X_{min} , Y_{min} , X_{max} and Y_{max} as minimum and maximum x and y coordinates of the irregular scanned polygon.
 - b. The corner points of the rectangular polygon are initialized as (X_{min}, Y_{max}) , (X_{min}, Y_{min}) , (X_{max}, Y_{max}) and (X_{max}, Y_{min}) .
2. Divide the rectangular polygon into a 2D grid matrix named as A with each box of the matrix having dimension L_x and L_y as length and breadth respectively.
3. Perform shadow analysis on the 2D matrix A to obtain the intensity values at all boxes of the 2D matrix A .
4. Make new 2D matrix named as B with same dimensions as mentioned in step 1 and step 2.
5. Initialize the value at $Box(i,j)$ of matrix B as:
 - a. Let X be the no. of columns and Y be the no. of rows that a single solar panel will occupy when placed in the rectangular polygon.
 - b. Take summation of values inside all boxes in the submatrix starting from $box(i,j)$ with X and Y as the no. of columns and rows respectively from matrix A .
 - c. Initialize $box(i,j)$ of matrix B as the summation value obtained in step 5.b.
6. Initialize X_c and Y_c as the horizontal and vertical distance between two solar panels respectively.
7. For each row of matrix B , find elements which have minimum $X+X_c$ distance between them and their summation is maximum. Store the maximum summation value for row ‘ i ’ at index ‘ i ’ in a 1D matrix named as C . Also store the elements which were selected for the maximum summation for each row.
8. In matrix C , find elements which have minimum $Y+Y_c$ distance between them and their summation is maximum.
9. For each index ‘ i ’ in matrix C , if the element was selected for maximum summation in step 8, then the elements selected for maximum summation of row ‘ i ’ in step 7 will be considered for placement of solar panels.

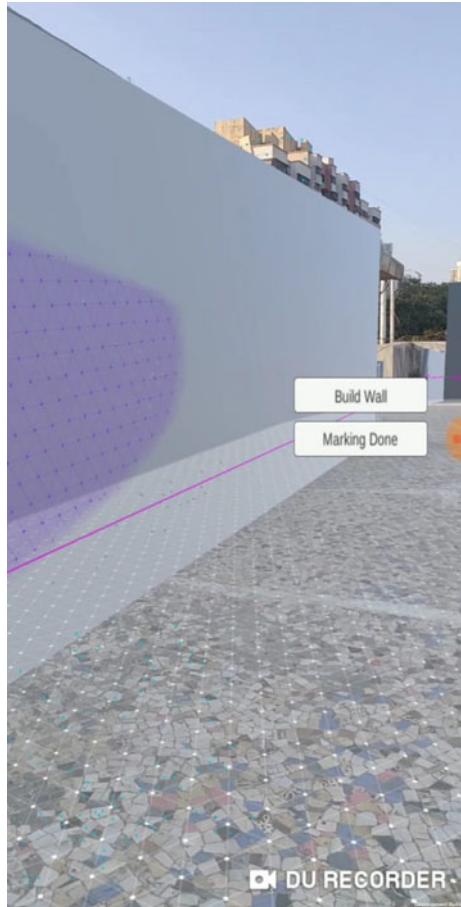
Step 7: Augment Solar Panel:

The 3D model of solar panel would be placed at the coordinates specified by the placement algorithm. The augmented view of array of solar panels can be seen on scanned location.

4 Results and Discussion

The application was used to scan rooftops of the buildings and the ground surfaces to assess the performance of the application. It was observed that the application accurately detected the rooftop surface. Also when the application was used with the ground surfaces, it was observed that it correctly detected the plane of the ground

Fig. 5 Scanning location and detection of plane surface on rooftop



surface. Also for uneven ground surfaces, the application was able to correctly detect the plane of the surface. The result of plane detection is shown in Fig. 5.

As seen in Fig. 6, the heatmap values along the boundaries are 320 and 560, which indicates the shadow region of the scanned location. Also the heatmap values –1000 indicate, that the obstruction is present over there. The results of scanning different areas are provided in Table 3.

Here Average Energy produced by a solar panel in 1 day is considered as 4.5 kWh.

5 Conclusion and Future Work

In this paper, a novel approach of conducting the initial survey of the location for the installation of solar panels is discussed. This is the first ever attempt made by anyone

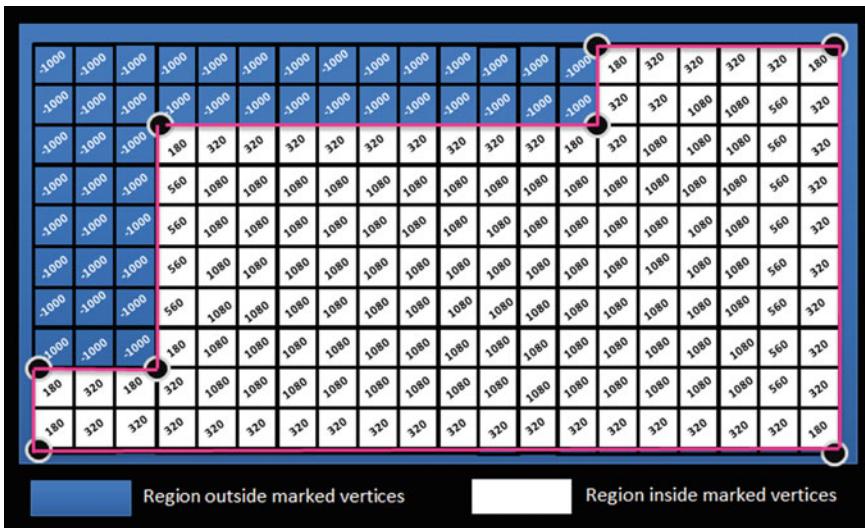


Fig. 6 Heatmap values generated for scanned rooftop

Table 3 Result of scanning different areas

Area (m ²)	No. of solar panels	Energy per meter sq. (W h/m ²) in 1 day	Time of calculation (s)
82.2	33	1810.97	69.7
7.1	3	1928.57	1.2
21.46	9	1887.23	3.1

to use augmented reality to scan the location and produce the augmented view of the optimal placement of the solar panels. The performance of the application is also good with the even as well as uneven surfaces. Further, from the results, it is inferred that our optimal placement heatmap algorithm is efficient and its processing is very quick. The application can be extended further to work with inclined surfaces. Also, the application can be enhanced to work with locations having non-linear boundaries and its performance can be evaluated.

Acknowledgements Department of Information Technology, K. J. Somiya College of Engineering, Mumbai developed this application in collaboration with Parallax Labs, LLP, Mumbai. We express our gratitude to the Institute and Parallax Labs for their constant support and guidance. We would also like to acknowledge the contribution of Bro. Soham Hichkad, Bro. Prasad Gujar, Bro. Niket Kini, Bro. Uddesh Kadu, LY B. Tech. students, Department of Information Technology, K. J. Somaiya College of Engineering, for the success of this research.

References

1. Pawade D, Sakhapara A, Mundhe M, Kamath A, Dave D (2018) Augmented reality based campus guide application using feature points object detection. *Int J Inf Technol Comput Sci (IJITCS)* 1(5):76–85
2. Just A Line: Draw anywhere, with AR, Google Creative Lab, Version 2.1.1, Google App Store. <https://play.google.com/store/apps/details?id=com.arexperiments.justaline>. Last accessed 2019/11/18
3. Measure: Quick Everyday Measurements. Google LLC, Version 2.1.1, Google App Store. <https://play.google.com/store/apps/details?id=com.google.tango.measure>. Last accessed 2019/11/18
4. AR Ruler App: Tape Measure and Camera To Plan, Grymala, Version 1.4.0, Google App Store. https://play.google.com/store/apps/details?id=com.grymala.aruler&hl=en_IN. Last accessed 2019/11/18
5. Augment: 3D Augmented Reality, Augment, Version 3.4.0, Google App Store. <https://play.google.com/store/apps/details?id=com.ar.augment>. Last accessed 2019/11/18
6. Veldhuis AJ, Reinders AHME (2015) Shadow analysis for BIPV and PIPV systems in a virtual environment. In: IEEE 42nd photovoltaic specialist conference (PVSC), proceedings in IEEE, New Orleans, LA, USA
7. Onime C, Uhomoibhi J, Pietrosemoli E (2015) An augmented virtuality based solar energy power calculator in electrical engineering. *Int J Eng Pedagog (iJEP)* 5(1):198–199
8. Onime C, Uhomoibhi J, Pietrosemoli E (2014) A demonstration of an augmented virtuality based solar energy power calculator in electrical engineering. In: 11th international conference on remote engineering and virtual instrumentation (REV), proceedings in IEEE, Porto, Portugal
9. Jacobson MZ, Jadhav V (2017) World estimate of PV optimal tilt angles and ratios of sunlight incident upon tilted and tracked PV panel. *Sol Energy* 169:55–66
10. Tarng W, Ou K Lu Y, Shih Y, Liou H (2018) A sun path observation system based on augment reality and mobile learning. *Mob Inf Syst* 2018:10. Article ID 5950732
11. Kim S, Suk H, Kang J, Jung J, Laine T, Westlin J (2014) Using unity 3D to facilitate mobile augmented reality game development. In: IEEE world forum on internet of things (WF-IoT), Seoul, South Korea
12. Voinea G, Girbacia F, Postelnicu C, Marto A (2018) Exploring cultural heritage using augmented reality through Google's Project Tango and ARCore. In: International conference on VR technologies in cultural heritage (VRTCH), proceedings in communications in computer and information science (CCIS) book series, vol 904. Springer, pp 93–106

Chapter 52

Automated Scoring System for Online Discussion Forum Using Machine Learning and Similarity Measure



Dipti Pawade, Avani Sakhapara, Rishi Ghai, Shruthi Sujith and Sneha Dama

1 Introduction

Collaborative learning is proved as an effective way of acquiring knowledge [1]. Since past few years, online discussion forum is considered as popular way where people can collaborate and carry out discussion [2, 3]. There are many evidences where contribution made in focused discussion carried out on discussion forum is considered as one of the factor to evaluate the learners' performance [4, 5]. In such cases, the instructors post a focus question, and learner has to respond to it according to their understanding. This process is really helpful to explore the various aspects of the particular question as every person think in his own way and put forth the solution. But when it comes to analysis or evaluation of the contribution made by the individual learner, it results out to be the tedious job. Some online discussion forums like in Moodle provide options by which instructor can get the number of learner who have responded to the question but they do not provide any kind of insight to the quality of the submitted response. In most of the cases, even though

D. Pawade (✉) · A. Sakhapara · R. Ghai · S. Sujith · S. Dama
Department of IT, K. J. Somaiya College of Engineering, Mumbai, India
e-mail: diptipawade@somaiya.edu

A. Sakhapara
e-mail: avanisakhapara@somaiya.edu

R. Ghai
e-mail: rishi.ghai@somaiya.edu

S. Sujith
e-mail: shruthi.sujit@somaiya.edu

S. Dama
e-mail: s.dama@somaiya.edu

the numbers of responses submitted by the learner are very large but out of these only a few actually contribute to the discussion. Some may simply make it known that they agree with a previous post or may give repetitive answers without checking if someone else had put forward the same point before them. Users have to examine large number of threads to find the answers that help. Thus, it is extremely difficult for instructors of online courses to go through thousands of posts to pick out the answers that actually contribute to the discussion. Looking at depth of this issue, we have developed modules which accept the URL of the discussion forum question and model answer for the same and process these responses efficiently, to facilitate a better learning experience by analyzing the responses based on its relevance and display the responses in descending order of their relevance using two levels of filtering.

2 Related Work

A lot of work has been done to effectively analyze the enormous amount of data that discussion forums generate. For instance, Jenders et al. [6] focus on enhancing user experience by using a machine learning model to predict the answer that is most likely to be marked as the accurate one by users. They used historical forum data of completed courses to train the model. Various types of forum data like user, thread and content features were first extracted and then distinguished. They have used features like being the first, last, most accepted, most voted answer, and the one having the most comments as baseline methods. Based on these features, the model has been trained using random forest classifier, multilayer perceptron, bagging and Naïve Bayes. The individual answers were then given scores in terms of probability of acceptance. The ones with scores above a certain threshold were assumed to be accepted. Similarly, Okfalisa et al. [7] focus on grouping similar comments and eliminating repetition so that users can get a quick overview of the conversation and understand the main points with ease. It is necessary to make discussions more focused, while maintaining accuracy and diversity. To do so, test processing, transformation and attribute selection were performed on datasets obtained from three different online forums. The centroid linkage hierarchical method (CLHM) was used for initial grouping of the comments. This was followed by the application of hill climbing methods to detect variance cluster patterns. This was found to improve up the results of CLHM greatly. To predict the similarity between the comments, cosine and Euclidean similarity measure was used. It was concluded that cosine similarity measure provided more exact results than Euclidean similarity. Oguzhan [8] proposes the use of machine learning algorithms to automate the process of classifying posts on a Finnish health discussion forum. The existing system relied on the user's judgment or manual labeling to classify the post. To automate this process, three machine learning algorithms, viz. multinomial Naïve Bayes, Bernoulli Naïve Bayes

and online passive-aggressive classifier were applied to the extracted dataset. The messages were classified into sixteen predefined categories. The multinomial Naïve Bayes classifier was found to give the most accurate result. Andrew et al. [9] use point processes to recommend threads to learners on forums. It models the probability that a learner contributes to a thread based on four factors, viz. interest, timescale, previous posts, and past involvement of the learner with the thread. The results were validated using a large MOOC dataset on Coursera. The training set includes data from a particular time interval and the test set includes the rest of the data, excluding new sets created during the test interval. The performance was checked using four baselines: popularity, recency, social influence and adaptive matrix factorization method. The model significantly outperformed all the baselines.

Another author [10] focuses on improving group task assignment on Moodle. They focused on clustering students on the basis of their collaboration competence levels, with little intervention by the instructor. The Weka tool was used to perform clustering on the data, using algorithms, namely, SK means and expectation–maximization (EM). Furthermore, Meriem et al. [11] have attempted to identify weak performers, among the students that have enrolled for online courses. The simulation was done using data extracted from the previously set up Moodle forum. A software tool named Gephi was used. The process focused on how students assimilate knowledge obtained from online cultural and social settings. Three algorithms were implemented, of which the first algorithm was used to visualize existing social networks in such forums. This algorithm mapped the nodes according to the forces of attraction or repulsion between them. The second algorithm was used to identify colonies or communities within the forum. The third algorithm, a part of the Gephi software, was used to better understand student interaction, and form clusters. Finally, the students were evaluated on the basis of a parameter called “degree centrality” which represents the numbers of links a node has. High interaction levels are linked to the acquisition of more knowledge. Hence, the students that had a high degree centrality were considered to have higher prestige in the network and are considered better learners. Students with a very low value were considered weak students that need help. Thushari et al. [12] aimed to make it easier for learners and instructors to navigate the loads of information with the help of a topic visualization dashboard. Educators need information about topics that are frequently discussed because students may be having some difficulty in understanding those. A Naïve Bayes classifier was used to classify the threads and summarization techniques were used to label them. These labels were represented using color-coded and size-adjusted bubble graphs for the instructors to easily understand. This topic-wise classification also aimed to act as navigational information for learners. Table 1 gives the summary of literature survey done in this section.

Table 1 Summary of the literature survey

Ref. No.	Dataset	Algorithm/method	Accuracy
[6]	Historical openHPI MOOC forum data	Random forest classifier, multilayer perceptron, bagging, Naive Bayes	99% (given by bagging)
[7]	Comments from Teknojournal.com, Indowebster.com and Bersosial.com	Cosine similarity and Euclidean distance	82% (given by cosine)
[8]	Posts from the “Health” category of Suomi24, Finland’s largest online discussion forum	Naive Bayes classifier	70.8%
[9]	Discussion forum datasets from 2012 offerings of MOOCs on Coursera for machine learning, algorithms, and English composition	Point process model (PPS)	15–40% improvement over the strongest baseline
[10]	Moodle discussion data	Clustering algorithms, viz. SK means and EM	NA
[11]	Discussion forums for two Moodle computer science courses	Fruchterman and Reingold, Louvain Blondel, ForceAtlas2	NA
[12]	Discussion contents obtained from three MOOCs for machine learning, statistics	Naïve Bayes classifier	NA

3 Implementation Methodology

Figure 1 gives the overview of the complete working of our system. It consists of the following steps:

1. Storing training and testing dataset:

For the experimental purpose, training and testing files available for Stanford’s NLP Sentiment140 dataset [13] are used. Here firstly, the columns consisting of text samples or comments and their polarity are extracted. The polarity value for each comment was given as 0 if negative and 4 if positive. We converted the values to [1, 0] for negative and [0, 1] for positive, in order to use them in a neural network, which only accepts a list of output variables.

2. Apply preprocessing on dataset:

Preprocessing operations were performed on the dataset in order to get it into a format that could be fed into the Long Short-Term Memory Recurrent Neural Network (LSTM RNN) model [14, 15]. The steps include:

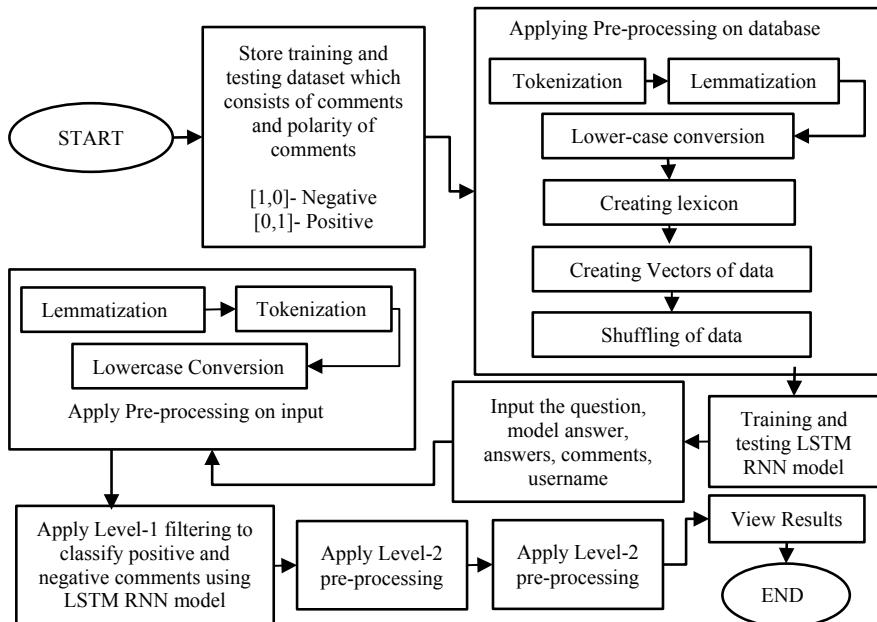


Fig. 1 Working of proposed system

- Tokenization: Each statement is tokenized each so that lexicon of unique words can be created. For example, the sentence is “Today is a sunny day, isn’t it?” then tokenized sentence: ['Today', 'is', 'a', 'sunny', 'day', ',', 'isn’t', 'it', '?].
- Lemmatization: Each token word is reduced to their lemmas, or basic form, so that multiple forms of the same word can be processed as one. For example, see, saw and seeing are forms of the same lexeme, with “see” as the lemma.
- Lower case conversion: The same words may be considered as different ones because of accidental capitalization of letters. So to avoid duplicates, all the words are converted to the lowercase form. For example, “Java”, “java” and “jAvA” will all be considered as the same word.
- Creating lexicon: Lexicon is a collection of words or a dictionary, which consists of all the unique words in the training dataset. All the unique words in the training dataset were extracted and added to a list to form lexicon.
- Creating vectors of data: A list having size equal to that of the lexicon is created. If a word is present in the lexicon, then the index of the word in vector is set as one.
- Shuffling of data: Overfitting happens when a model learns the detail in a flawed training data which negatively impacts the performance on new data. For example, consider a dataset with 10,000 samples, with the first five thousand samples belonging to class A and the other five thousand to class B. If we train the model using the first fifty thousand samples and test with the rest, the results would be quite unsatisfactory. Shuffling is necessary for this reason.

3. Training LSTM RNN model:

It involves training a single-layer LSTM RNN model for determining the polarity of text data, using the Sentiment140 training dataset previously mentioned. The model was trained for seven epochs. An epoch is a complete cycle of forward and backward propagation. The data was processed in batches of 128 rows. The Softmax function was used for calculating cost at the end of forward propagation. In forward propagation, the inputs are multiplied by weight and biases are added to it. To reduce the cost, an Adam optimizer [16] was used for backpropagation. In backward propagation, we adjust the weights to reduce the cost.

4. Input:

The instructor is expected to provide two inputs (a) link to the discussion Web page (b) the model answer. First, Web scraping is performed on the discussion page, i.e., the attributes of html tags in the page are used to extract the answers and comments. It was observed that all discussion forums share a particular structure which includes a question at the top, answers to the question, and replies or comments on the answers.

5. Applying preprocessing on input:

After extracting answers and comments from the Web page, preprocessing operations are applied on them. The operations include the following: tokenization, lemmatization and lower-case conversion.

6. Apply level 1 processing:

Level 1 processing is only done on comments, and not on answers, since only comments express agreement or disagreement with answers. When evaluating a comment, it is important to know its polarity, i.e., whether it is positive or negative in nature, since a comment supporting an accurate answer deserves more points, as compared to a comment supporting a negative answer. So, the polarity of comments using the pre-trained LSTM RNN model is detected. Using the model, the comments on answers were classified as either positive or negative.

7. Apply level 2 processing:

Here each answer is compared with the model answer to determine how similar it is to the model answer. The scores are generated in the range between 0 and 100.

Word Mover's Distance (WMD): To determine the relevance of the answers and comments with the model answer, Word Mover's Distance (WMD) is used [17, 18]. WMD determines the dissimilarity between documents. WMD scales automatically to the size of the text. This is very important for this use case, as relatively short text values are being compared with each other, and not large documents. Only the answers and comments are being compared with the model answer. WMD uses word embedding to calculate the distance. So, the absence of common words does not affect the result. The assumption is that similar words should have similar vectors. Word embedding is vector representations of document vocabulary, used to capture context of a word in a document, semantic and syntactic similarity and relation with other words. Word2Vec [19] is also used here.

8. Calculating the answer and the comment score:

Word Mover's Distance is applied to the comments (comment_wmd) and answers (answer_wmd) with respect to the model answer. The lesser the value, the more similar are the documents or sentences. Now the values of answer_wmd and comment_wmd are converted into similarity values between 0 and 1 using Eqs. (1) and (2).

$$\text{answer_similarity(as)} = \frac{1}{(1 + \text{answer_wmd})} \quad (1)$$

$$\text{comment_similarity(cs)} = \frac{1}{(1 + \text{comment_wmd})} \quad (2)$$

The values of similarity of answers (answer_sim) and comments (comment_sim) are always between 0 and 1. The higher the value, the more similar is the answer or the comment to the model answer. The score of the answer (answer_score) is calculated between 0 and 'n' marks, using Eq. (3)

$$\text{answer_score} = \text{round}(\text{answer_sim} * n) \quad (3)$$

For determining the score of the comment, WMD is applied to the comment (comment_answer_wmd) with respect to the student answer. Then the similarity of comment with the student answer (comment_answer_similarity) is computed using Eq. (4)

$$\text{comment_answer_similarity(ca)} = \frac{1}{1 + \text{comment_answer_wmd}} \quad (4)$$

Using Eqs. (1), (2) and (4), the similarity of the comment relative to the answer similarity (relative_similarity) is calculated as follows:

1. If as ≥ 0.6 and ca ≥ 0.6 , then relative_similarity = |ca - cs|

Here, the student answer is very similar to the model answer, but the comment is nearly same as the student answer. There is no new information provided in the comment with respect to the student answer.

2. If as ≥ 0.6 and ca < 0.6 , then relative_similarity = cs

Here, the student answer is very similar to the model answer, but the comment is different from the student answer. The comment may or may not contain new information with respect to the student answer.

3. If as < 0.6 and ca ≥ 0.6 , then relative_similarity = |ca - (1 - cs)|

Here, the student answer is not so similar to the model answer, but the comment is nearly same as the student answer. There is no new information provided in the comment with respect to the student answer.

4. If $as < 0.6$ and $ca < 0.6$ then $\text{relative_similarity} = cs$

Here, the student answer is not so similar to the model answer, and the comment is different from the student answer. The comment may or may not contain new information with respect to the student answer.

Finally, the score of the comment (`comment_score`) ranging from 0 to ' m ' marks is calculated using `relative_similarity` as given by Eq. (5)

$$\text{comment_score} = \text{relative_similarity} * m \quad (5)$$

9. Viewing the result:

The scores of answers and comments are displayed along with the respective answers and comments. With the comment, red color is used to indicate positive polarity and green color is used to indicate negative polarity of the comment with respect to the student answer. A report is generated having the details such as answer id, answer/comments, username and score. The results can be sorted in ascending or descending order, which would further help the instructor evaluate/gain insights from the responses of the students.

4 Results and Discussions

For the experiment, the discussion forum questions of stackoverflow [20] are used. In real-time, the URL of any of the posts from stack overflow can be inputted into our system. Till date there are 17 million questions, 27 million answers and 72 million comments on stackoverflow. This is the largest dataset of discussion forums. On stackoverflow, the correct answers are marked with green color tick mark which is treated as model answer by our system. For measuring the accuracy of the system, we have considered 120 questions, 234 answers and 368 comments. The scores for the answers are generated between 0 and 3 and the scores for the comments are generated between 0 and 2. A sample dataset with the calculated score values for the answer and comment is provided in Table 2. For the demonstration purpose, the URL of the first sample question (SN.1) is given as an input to the system. For this question, SN.2 is a model answer (SN.2) against which similarity of all the answers is calculated. For this particular question (SN.1), approximately there were 31 answers. For each answer, the WMD value and answer similarity are calculated and based on that score is given to each answer. In this case, best score was 2 for the answer given as SN.3. Now for this answer, there were approximately 23 replies which additionally contribute to the answer. So it becomes equally important to highlight the most relevant reply. Thus, level 2 filtering is applied on the comments and most contributing comment with its polarity is highlighted (SN.4).

For measuring the accuracy of the system, the system-generated scores for the answers and the comments are compared by the scores given by human evaluators. As

Table 2 Sample dataset with calculated scores

SN.1	Question	“What is an OS kernel? How does it differ from an operating system?” [20]
SN.2	Model answer	The technical definition of an operating system is “a platform that consists of specific set of libraries and infrastructure for applications to be built upon and interact with each other”. A kernel is an operating system in that sense. The end-user definition is usually something around “a software package that provides a desktop, shortcuts to applications, a web browser and a media player”. A kernel does not match that definition. So for an end-user a Linux distribution (say Ubuntu) is an operating system while for a programmer the Linux kernel itself is a perfectly valid OS depending on what you’re trying to achieve. For instance, embedded systems are mostly just kernel with very small number of specialized processes running on top of them. In that case, the kernel itself becomes the OS itself. I think you can draw the line at what the majority of the applications running on top of that OS do require. If most of them require only kernel, the kernel is the OS, if most of them require X Window System running, then your OS becomes X + kernel” [20]
SN.3	Answer	“A kernel is the part of the operating system that mediates access to system resources. It’s responsible for enabling multiple applications to effectively share the hardware by controlling access to CPU, memory, disk I/O, and networking. An operating system is the kernel plus applications that enable users to get something done (i.e. compiler, text editor, window manager, etc.)” [20] answer_wmd = 3.3742, answer_similarity = 0.22 answer_score = 2 mark (out of 3M)
SN.4	Comment	“You were doing well until you wrote the parenthetical remark in the second paragraph. An operating system handles requests made by those applications to control the hardware. So, the OS includes the kernel, devices drivers, and any other software/firmware that specifically controls the hardware at the lowest level” [20] comment_answer_similarity = 0.13, comment_similarity = 0.28 relative_similarity = 0.56, comment_score = 1 mark (out of 2 marks), polarity = positive

mentioned earlier, we have considered 120 questions, 234 answers and 368 comments for calculating the accuracy of the system. Here the accuracy is measured in terms of 3 accuracy values viz., accuracy of generating the scores of the answers, the accuracy for generating the scores of the comments and accuracy of determining the polarity of the comments. The accuracy of generating the scores of the answers is 81%, the accuracy for generating the scores of the comments is 76% and the accuracy of determining the polarity of the comments using LSTM RNN is 80%.

5 Conclusion and Future Work

In this paper, we have proposed and implemented a system to automatically generate the scores of the answers and the comments posted on online discussion forums. In online courses, this can be very helpful in assessing the participation of students in the discussion forums. But the challenge is the complex structure of the discussion forums. In our system, we have used LSTM RNN to determine the polarity of the comments and have got 80% accuracy. Using WMD for determining the scores of the answers, we achieved accuracy of 81%. We have considered the similarity between the answer and the comment, which ensures that only those comments get marks where some additional information is added as compared to the answer and the comment is original. We have achieved accuracy of 76% for the score generation of the comments. The system can be extended further to compute the contribution of students across multiple questions in the discussion forum. Also a feedback mechanism can be incorporated in the system, where the students will get a detailed feedback of where they lost marks. The system can be implemented for different types of discussion forums and its performance can be investigated.

References

1. Laal M, Ghodsi SM (2012) Benefits of collaborative learning. *Procedia Soc Behav Sci* 31:486–490
2. Camarero C, Rodríguez J, San José R (2012) An exploratory study of online forums as a collaborative learning tool. *Online Inf Rev* 36(4):568–586
3. Sun Z, Lin C, Wu M, Zhou J, Luo L (2017) A tale of two communication tools: discussion forum and mobile instant messaging apps in collaborative learning. *Br J Educ Technol* 49:248–261
4. Seethamraju R (2017) Effectiveness of using online discussion forum for case study analysis. *Educ Res Int* 2014:1–10. Article ID 589860
5. Du X, He Z, Li H, Zhu X (2016) A study on evaluation of online discussion forums in cloud learning environment. In: International conference on progress in informatics and computing (PIC), Shanghai, China, 23–25 Dec 2016
6. Jenders M, Krestel R, Felix N (2016) Which answer is best? Predicting accepted answers in MOOC forums. In: 25th international conference companion, April 2016
7. Okfalsa, Iskandar J (2017) The application of centroid linkage hierarchical method and hill climbing method in comments clustering online discussion forum. In: 5th international conference on cyber and IT Service Management (CITSIM), Aug 2017
8. Gencoglu O (2017) Automatic classification of forum posts: a Finnish online health discussion forum case. In: Joint conference of the European medical and biological engineering conference (EMBEC) and the Nordic-Baltic conference on biomedical engineering and medical physics (NBC), Tampere, Finland, June 2017
9. Lan AS, Spencer JC, Chen Z, Brinton CG, Chiang M (2018) Personalized thread recommendation for MOOC discussion forums. In: Computing research repository. [arXiv:abs/1806.08468](https://arxiv.org/abs/1806.08468)
10. Maina EM, Oboko RO, Waiganjo PW (2017) Extending moodle grouping functionality using artificial intelligent techniques. In proceeding of IEEE AFRICON, Cape Town, South Africa, Sept 2017

11. Adraoui M, Retbi A, Idrissi MK, Bennani S (2018) Network visualization algorithms to evaluate students in online discussion forums: a simulation study. In: International conference on intelligent systems and computer vision (ISCV), April 2018
12. Atapattu T, Falkner K, Tarmazdi H (2016) Topic-wise classification of MOOC discussions: a visual analytics approach. In: 9th international conference on educational data mining at North Carolina, USA, June 2016
13. Dataset Link for determining polarity. <http://help.sentiment140.com/for-students/>
14. Pawade D, Jain M, Sarode G (2016) Methods for automatic text generation. In: i-manager's J Comput Sci 4(4):32–36. December 2016–February 2017
15. Pawade D, Sakhapara A, Jain M, Jain N, Gada K (2018) Story scrambler—automatic text generation using word level RNN-LSTM. Int J Inf Technol Comput Sci 10(6):44–53
16. Kingma DP, Lei Ba J (2015) Adam: a method for stochastic optimization. In Proceedings of 3rd international conference for learning representations (ICLR), San Diego, 2015
17. Kusne M, Sun Y, Kolkin N, Weinberger K (2015) From word embeddings to document distances. In Proceeding of 32nd international conference on machine learning, Lille, France, 2015
18. Pawade D, Sakhapara A, Ratlamwala H, Mishra S, Shaikh S, Mehta D (2019) Implementation of smart legal assistance system in accordance with the Indian Penal Code using similarity measures. In 3rd international conference on advances in computing and data sciences, Ghaziabad, India, 12–13 April 2019
19. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. <https://arxiv.org/abs/1301.3781>
20. Input Data Link. <https://stackoverflow.com/>

Chapter 53

Intrusion Detection: A Machine Learning Approach



Vipul Borhade, Aparna Nayak and R. Dakshayani

1 Introduction

Intrusion generally refers to malicious activities directed at computer network system to compromise its integrity, availability, and confidentiality. Network security is important because modern information technology relies on it to drive businesses and services. Security can be enforced on the network through intrusion detection systems. These are security devices or software usually implemented by large and medium organizations to enforce security policies and monitor network perimeter against security threats and malicious activities. Other associated systems include firewall and intrusion prevention system. Essentially, intrusion detection device or application scrutinizes every incoming or outgoing network traffic and analyzes packets for known and unknown events. Detected known events and violations are logged usually in a central security information and event management system. Malicious activities or unknown events may be set up to alert system administrator or the related packets dropped depending on the configurations enabled on the intrusion detection system. Prevention of security breaches cannot be completely avoided. Hence, effective intrusion detection becomes important for organizations to proactively deal with security threats in their networks. However, many existing intrusion detection systems are rule-based [3] and are not quite effective in detecting a new intrusion event that has not been encoded in the existing rules. Besides, intrusion detection rules development is time-consuming and it is limited to knowledge of known intrusions

V. Borhade (✉) · A. Nayak · R. Dakshayani
Department of Computer Engineering, FCRIT, Mumbai University,
Vashi, Navi Mumbai 400703, India
e-mail: vborhade75@gmail.com

A. Nayak
e-mail: naparnanayak@gmail.com

R. Dakshayani
e-mail: dakshayani.r@fcrit.ac.in

Table 1 Comparative study of different algorithms

	Cross Validation (10 Folds)	
	Correctly classified	Incorrectly classified
Naive bayes	58866 (76.19%)	18245 (23.84%)
Decision tree	76141 (98.51%)	1150 (1.49%)
KNN	76474 (98.94%)	817 (1.06%)
Random forest	76829 (99.40%)	462 (0.60%)

only. Data mining techniques, on the other hand, through supervised and unsupervised learning algorithms have been shown to be effective in identifying and differentiating known and new intrusions from network event records or data [5]. It is, therefore, worthwhile to explore the application of data mining techniques as an effective alternative approach to detect known and potential network intrusions. The proposed method aims to find a suitable machine learning algorithm which can predict the type of network attack with the highest accuracy and then develop a system which uses this algorithm to detect network intrusion. Table 1 also shows a comparative study on raw data in Weka tool based on Naive Bayes, decision tree, K-nearest neighbor, and random forest algorithms. Random forest algorithm shows the highest accuracy compared to all other algorithms; hence, the method proposed in this paper is based on random forest algorithm. The dataset used for model training is NSL-KDD dataset. NSL-KDD is dataset introduced to solve a problem like experimental validation of data, possibility of dropped packets, no identification of exact definition of attack, and duplication of records in the KDDCup99 dataset. The total number of records in NSL-KDD dataset is 1,152,281 which is helpful for training and testing of model. The evaluation of the results of different projects will be consistent and easily comparable. The major reasons to chose the NSL-KDD dataset over the KDD 99 dataset were because there are no duplicate records in the NSL-KDD datasets; therefore, data will not suffer the problem of overfitting and the performance of the learners is not biased by the methods which have better detection rates on the frequent records and the classification rates of distinct machine learning methods vary in a wider range, which makes it possible to have an accurate evaluation of different learning techniques.

2 Literature Survey

Intrusion detection system is important because it helps in identifying suspicious activities in your network and prevent further damage. In recent years, many machine learning algorithms are proposed which helps in the improvement of intrusion detection systems. Chang [2] model with the help of KDD-1999 Cup dataset their model is based on feature extraction with the help of random forest algorithm and train model using support vector machine. They have claimed that their accuracy is improved from 90 to 95% when they selected 14 features instead of total 41 features available

in the dataset. But it also increases their false alarm detection rate by 2 to 3% which is not accepted in case of intrusion detection.

Primartha [4] proposed model for IoT-based systems where they used the random forest algorithm on different datasets like NSL-KDD, UNSW-NB15, and GPRS considerably different number of trees and evaluate results based on accuracy and false alarm rate. In terms of accuracy, the NSL-KDD dataset performs better with an accuracy of 99%, unlike UNSW-NB15 which has an accuracy of 95% and GPRS whose accuracy ranges from 89 to 92% depending on a number of trees.

Ahmad [1] gives a comparison of support vector machine, random forest, and extreme learning algorithms. They use NSL-KDD dataset by considering only numeric features. Machine learning algorithm is applied for full samples, half samples, and 1/4 samples by taking 80% of total available data as training samples and 20% is taken as testing samples. The accuracy of support vector machine is around 99%. For random forest, it shows accuracy from 97 to 97.5% and for extreme learning ranges from 97.5 to 99% depending upon a number of samples taken.

3 Proposed Method

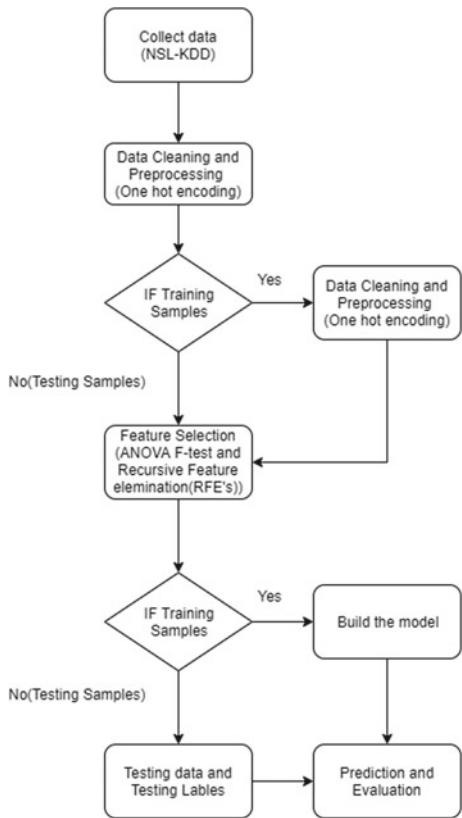
As a preprocessing, data is cleansed with one-hot encoding. All nonnumerical features will be converted into binary format. Furthermore, only training data samples are scaled by removing mean to cluster the samples. Figure 1 describes the block diagram of the proposed method. Feature selected by considering all the samples of training and testing data. The cross-validation method is used which helps to get better model. The description of the proposed method is as follows.

3.1 Data Preprocessing

All features are made numerical using one-hot encoding. This technique will transform each categorical feature with m possible inputs to n binary features, with one active at the time only. The features are standardized by scaling to unit variance to avoid the influence of large values. Each feature will have zero average with standard deviation of one after feature scaling.

3.2 Feature Selection

Eliminate redundant and irrelevant data by selecting a subset of relevant features that fully represent the given problem. ANOVA F-test is useful to determine contribution of each feature with respect to labels; univariate feature selection is done using ANOVA F-test method. Percentile of highest score is determined using percentile method (sklearn.featureselection). Recursive feature elimination (RFE) is applied on the subset to select the most contributing features in the system.

Fig. 1 Proposed method

3.3 Build the Model

A large number of individual decision trees that operate ensemble in random forest. Random forest tree model is built in Python using Colab notebook and Keras. By leaving one-third of the cases from sample, training set for the current tree is drawn. Sampling with replacement method is used. As trees are added to the forest, it is possible to run unbiased estimate of the classification because of out-of-bag (OOB) data. It is also used to get estimates of variable importance. Proximities are computed for each pair of cases as each tree is built and all of the data run down the tree. Proximity is increased by one if two cases occupy the same terminal node. By dividing the number of trees, proximities are normalized at the end of each run. To illuminate low-dimensional views of the data, the replacement of missing data and to locate outlier proximities could be used. Outstanding accuracy among all other existing algorithms can be achieved by this method. The same method could be used on large datasets. Classification of new data from input vector is possible by attaching input vector down each of the trees in the forest. Classification given by each tree is known as “votes” for that class. The classification having the most votes is chosen by forest.

3.4 Prediction and Evaluation (Validation)

Data is divided into two parts for testing and training purposes. Based on test data, prediction model is built. Multiple measures such as accuracy score, f-measure, recall, and confusion matrix are considered based on tenfold cross-validation.

4 Result Analysis

Figures 2 and 3 show the accuracy of cross-validation model using recursive feature elimination and cross-validation (RFECV) graph. A slight fluctuation in accuracy is visible if the total number of features considered to build model is changed.

Figure 4 shows the confusion matrix for the various types of attacks such as normal, probe, and DoS. Tables 2 and 3 show the confusion matrix for probe and DoS attacks, respectively. It is evident from the matrix that actual normal and probe attacks and predicted normal and probe attacks were identified more accurately both

Fig. 2 Feature selection:
DoS using RFECV

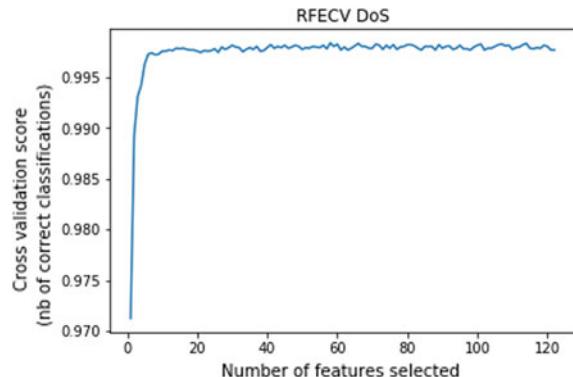


Fig. 3 Feature selection:
Probe using RFECV

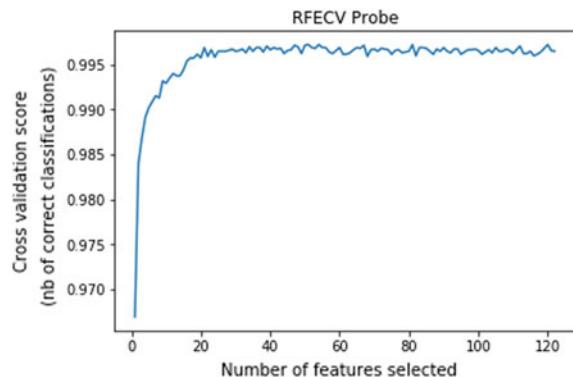


Fig. 4 Confusion matrix of three various attack

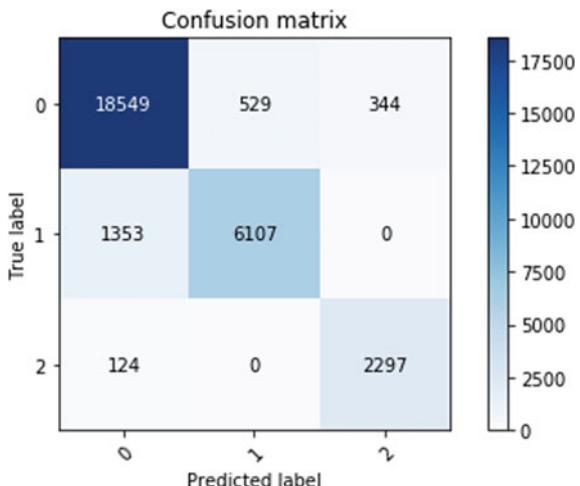


Table 2 Confusion matrix of probe attack

Predicted	Actual	
	Normal	Probe
Normal	9367	344
Probe	124	2297

Table 3 Confusion matrix of DoS attack

Predicted	Actual	
	Normal	DoS
Normal	9182	529
DoS	1353	6107

probe and DoS attacks. Accuracy of probe attack is 96.14%. For DoS attack, accuracy is 89%. Since the proposed method considered all features of the available dataset, it is found that accuracy is reduced. In [1], the author has used a very complicated procedure and not used the complete dataset to compute the accuracy.

5 Conclusion and Future Work

The proposed method reviewed the basis of intrusion detection system and application of machine learning algorithms like random forest, SVM, and Naive Bayes in intrusion detection systems. Algorithm is built for network-based intrusion detection system.

In future, it is expected that current model is further developed to detect various other types of attacks.

References

1. Ahmad I, Basher M, Iqbal MJ, Rahim A (2018) Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access* 6:33789–33795
2. Chang Y, Li W, Yang Z (2017) Network intrusion detection based on random forest and support vector machine. In: 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), vol 1. IEEE, pp 635–638
3. Kshirsagar VK, Tidke SM, Vishnu S (2019) Intrusion detection system using genetic algorithm and data mining: an overview
4. Primartha R, Tama BA (2017) Anomaly detection using random forest: a performance revisited. In: 2017 international conference on data and software engineering (ICoDSE). IEEE, pp 1–6
5. Shah SAR, Issac B (2018) Performance comparison of intrusion detection systems and application of machine learning to snort system. *Future Gener Comput Syst* 80:157–170. <https://doi.org/10.1016/j.future.2017.10.016>, <http://www.sciencedirect.com/science/article/pii/S0167739X17323178>

Chapter 54

Automated Damage Detection in Operational Vehicles Using Mask R-CNN



Naeem Patel, Shantanu Shinde and Freddy Poly

1 Introduction

Automated damage detection in vehicles has emerged as a key component in preventing accidents happening due to laxity in the inspection stage. This is a daunting task primarily due to two aspects. First of all, extracting the features for a damaged part in an operational vehicle is not very well defined and thus difficult to isolate under varying weather and lighting conditions. Secondly, the algorithm should be able to distinguish between regions of overlapping damage and thus should be able to segregate them.

With the advent of computing infrastructure and gigantic volumes of data, deep neural network [1] has achieved massive success in the domain of computer vision. The success of AlexNet [2] in the ImageNet Large Scale Visual Recognition Challenge [3] made convolutional neural network(CNN), a class of the deep neural nets the de facto standard for analyzing visual imagery. This has paved the way for major advances in the fundamental problems like image classification [4], semantic segmentation [5], object detection [6].

Our system is based on Mask R-CNN [7], which is an object instance segmentation model along with object detection. Mask R-CNN extends the faster R-CNN [8] by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. There are many advantages to this system. Firstly, Mask R-CNN surpasses all the previous state-of-the-art single-model results on the COCO

N. Patel (✉) · S. Shinde · F. Poly
Fr.C Rodrigues Institute Of Technology, Navi Mumbai 4000703, India
e-mail: naeempatel010@gmail.com

S. Shinde
e-mail: shantanushinde.shinde@gmail.com

F. Poly
e-mail: freddypoly99@gmail.com



Fig. 1 Mask R-CNN results on the COCO test set. These results are based on ResNet-101. image taken from [7]

instance segmentation task [9]. Secondly, damaged sections of vehicle are minute and far in between which makes instance segmentation [10] very vital task in solving the damage detection task.

This paper is organized as follows. Section 2 consists of the literature survey in the domain of damage detection and Mask R-CNN. Section 3 describes the approaches for data collection, preparation, and ground truth generation. Section 4 provides detailed information about the Mask R-CNN. Section 5 provides the implementation details on training of the model. Section 4 presents the outcome of our work, and finally, Section 7 outlines our conclusion (Fig. 1).

2 Related Works

2.1 Deep Learning-Based Car Damage Classification

Patil et al. [11] proposed the use of a convolutional neural network (CNN)-based methods for classifying different types of car damages like bumper dents, door dents, shattered glass, broken head, and tail lamps. After creating their own datasets and manually annotating them, they used various techniques and made a remarkable observation that transfer learning combined with ensemble learning provides better results.

2.2 *Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone*

Maeda et al. [12] talk about how different proposed methods which are available cannot be compared with each other, as they use their own datasets and their accuracy changes with different datasets. They created their own dataset with a considerably large amount of images and organized them to make it more reliable. The final product had wondrous accuracy rate and their dataset is one of the first road damage detections.

2.3 *Adapting Mask R-CNN for Automatic Nucleus Segmentation*

Johnson [13] proposed the application of Mask R-CNN for segmentation of microscopy images of cell nuclei. The model was trained on two different backbone networks namely ResNet50 and ResNet101, out of which Resnet101 provided significantly improved results. Instead of training the model end-to-end, the model was trained in three stages. First, training the network heads, then the upper layers and finally reducing the learning rate by a factor of 10 and training end-to-end.

2.4 *Automatic Knee Meniscus Tear Detection and Orientation Classification with Mask R-CNN*

Couteaux et al. [14] used Mask R-CNN to detect and isolate four different classes of Knee Meniscus Tears. They performed morphological preprocessing, namely black top-hat filter in order to bring robustness to the classification. The model was trained by transfer learning on the MS-COCO dataset using Adam optimizer and a batch size of eight images. Ensemble aggregation was applied by training five different models on random folds of the training data.

2.5 *Inshore Ship Detection Using Mask R-CNN*

Shanlan et al. [15] utilize Mask R-CNN architecture which includes a two-step procedure. The first one is region proposal network(RPN). The second step consists of a fast R-CNN and a binary mask prediction branch. The replacement of non-maximum suppression(NMS) with the soft-NMS provides robust results in the detection of battleships and merchant ships. The system achieves good performance for detecting nearby inshore ships.

3 Dataset

3.1 Data Collection

The dataset comprises of 326 images of both damaged and undamaged vehicles. The images in the dataset were generated by using smartphone cameras or were sourced by crawling the internet. Each image has a resolution of 600×600 pixel. Out of the 326 images, 80% of the images were used for training and the remaining 20% images were used for testing. While capturing the images, the following parameters like vehicle with varying color, damage types, lighting conditions, shape, intensity, and location of damages were considered.

This presents a significant additional challenge as convolutional neural networks can be expected to perform best, in general, when the input data is as uniform and standardized as possible. This includes standardization in terms of color, contrast, scale, and class balance.

3.2 Ground Truth Generation

Preparing the data involved manually annotating the masks for each fault in the vehicle using the VGG image annotator (VIA) tool [16]. No installation is required as the tool is available on the internet and can be accessed through a Web browser. This tool is used to define regions in an image and create textual descriptions of those regions. The VIA tool saves the annotations in a JSON file, and each mask is a set of polygon points. An illustration of this tool is given in Fig. 2. This resulted in 260 images and corresponding damage section.

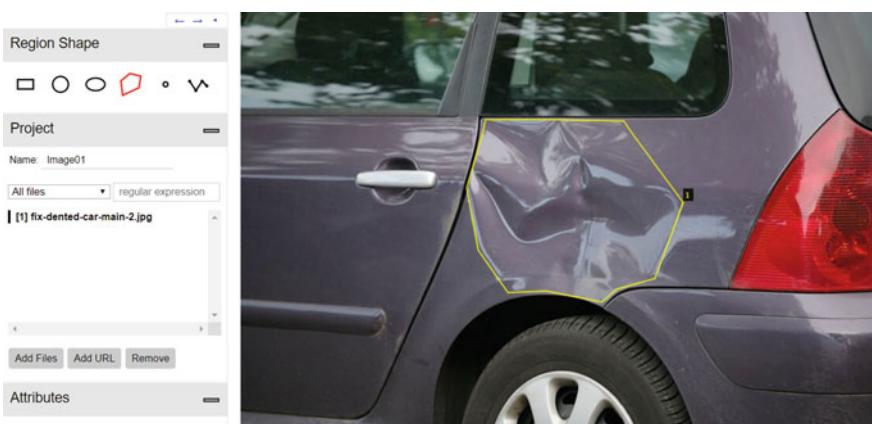


Fig. 2 Manually annotated image using VGG image annotator too

4 Mask R-CNN

The Mask R-CNN developed in 2017 is a framework for instance segmentation, localization, and object detection. It extends the faster R-CNN by adding a branch for predicting segmentation mask on each region of interest (RoI) in parallel with existing branch and bounding box regression as given in Fig. 3. Mask R-CNN replaces the less accurate ROI Pooling operation in the faster R-CNN by ROI Align which does not adjust the input from region proposal network(RPN) to fit the feature map correctly. It simply takes the object proposal and divides it into a certain number of bins. In each bin, a certain number of points are sampled and value at those points is determined using the bilinear interpolation. Also, Mask R-CNN defines a multi-task loss which combines the loss of classification, localization, and segmentation mask on each sampled ROI given by

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}} \quad (1)$$

where L_{cls} and L_{box} are the same as in Faster R-CNN. Mask R-CNN can be divided into several parts:

1. *Backbone Network*: This is a standard convolutional neural network(CNN) typically ResNet50 or ResNet101 [17] which serves as the feature extractor. The lower level of the network detects features such as edges, corners, whereas the later level detects higher level of features. ResNet101 is utilized as its additional layers help in leveraging the sparse dataset.
2. *Region Proposal Network(RPN)*: The image goes through a convolutional network which will take the feature map from the backbone network as input. A sliding window is run spatially on these feature maps. The areas scanned by the region proposal network(RPN) are called anchors. Furthermore, for each of these

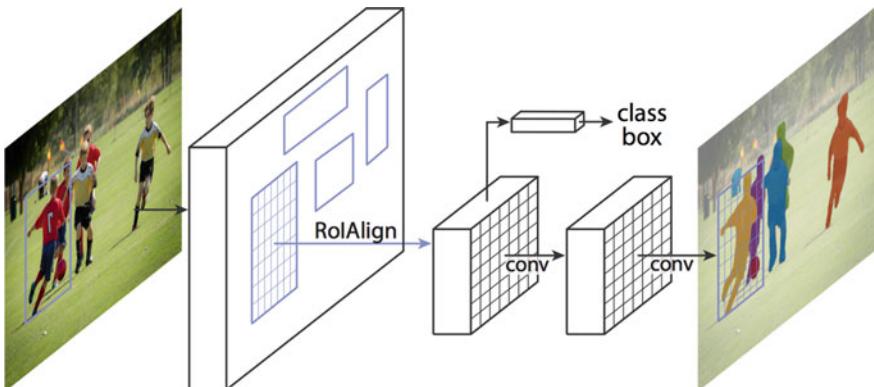


Fig. 3 The Mask R-CNN framework for instance segmentation

anchors, the probability is computed which indicates how much these anchors overlap with the ground truth bounding boxes. Finally, the extracted spatial features from the convolution feature maps are fed to a smaller network that has two tasks: classification and regression. The output of the regressor determines a predicted bounding box. The output of classification sub-network is a probability indicating whether the predicted box contains an object or it is from the background.

3. *RoI Align*: One of the most significant changes between the Mask R-CNN and Faster R-CNN would be replacement of ROI Pooling with ROI align. Both the operations generate a uniform $P \times P$ matrix for all the region of interest(ROI) from the RPN stage. ROI Pool works well in the case of object detection but fails terribly in the case of instance segmentation, as it has too many quantization steps. This affects the generation of mask, where pixel to pixel correspondence matters. RoI Align is very similar to cropping a part of an image and then resizing it. Since it does not use any quantization of features unlike RoI Pool, it manages to perform pixel-to-pixel alignment between network inputs and outputs comfortably.
4. *Segmentation Mask*: In order to generate a segmentation mask, we calculate the region of interest(ROI) so that computation time can be decreased. The Intersection over Union(IoU) is calculated by using the following formula:

$$\text{IoU} = A_{\text{Intersection}} / A_{\text{Union}} \quad (2)$$

where $A_{\text{Intersection}}$ and A_{Union} correspond to area of the Intersection and area of the Union, respectively.

If the IoU is greater than or equal to 0.5, it is considered as a region of interest. Otherwise, we neglect that particular region. After obtaining the region of interest (RoI) based on the Intersection over Union (IoU) values, a mask branch to the existing architecture is added. This returns the segmentation mask for each region that contains an object. It returns a mask of size 28×28 for each region which is then scaled up for inference and scaled down in case of training to compute the loss.

5 Implementation Details

In this proposed work, the excellent implementation of the Mask R-CNN by Abdulla et al. [18] is utilized to perform our experiments. The implementation of the Mask R-CNN is done in TensorFlow [19] and it utilizes Feature Pyramid Network(FPN) and a ResNet101 as a backbone. Due to the very less amount of training data, transfer learning was deemed as a viable option in which the pretrained weights of the MS-COCO dataset [9] are used as a starting point for training.

To generate a very large training set, horizontal flipping the images during the training time was performed. Before the training, resizing of all the images to the shape 600×600 pixel was performed. The training started with a learning rate of

0.001 along with a learning momentum of 0.9. A mini-batch size of eight images was used to train the model for 63 iterations.

All the training was performed on Google Colaboratory (a.k.a. Colab) [20], which is a cloud service based on Jupyter Notebooks for disseminating machine learning education and research. It provides a runtime fully configured for deep learning and free-of-charge access to a robust GPU. Google Colab provides NVIDIA Tesla K80 GPU with 12GB of DDR5 memory. Training and testing of the above model for damage detection were also performed on the NVIDIA GeForce 1050 Ti which has a memory size of 4GB to observe the performance in terms of detection accuracy and speed.

6 Observations

6.1 Detection Accuracy

To evaluate the accuracy in the detection of damaged parts, 100 images of damaged vehicles were collected. Figure 4 illustrates the results on the test data. Recall, preci-



Fig. 4 Results obtained from our trained Mask R-CNN model

Table 1 Performance of our system

Model/Metrics	Recall	Precision	F_1 score
Mask R-CNN (%)	73.46	80.39	76.76

Table 2 Inference speed of proposed system on various GPU

Inference speed	GPU used	
	Nvidia GTX 1080 Ti	Tesla K80
Average time per image (ms)	23	58

sion, and F1 score were used to evaluate our system as shown in Table 1. we calculate the F1 score from the recall and precision given by the formula:

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

6.2 Inference Speed

Calculation of the average inference speed on the test data of around 100 images on the Google Colab Platform and NVIDIA GeForce 1050 Ti, is performed. Table 2 shows the average time taken to run the inference for the model.

7 Conclusion

For this paper, we evaluated the gigantic capability of utilizing the avant-garde instance segmentation techniques, specifically Mask R-CNN for perceiving damaged parts in vehicles such as bumps, dents, and scratches. This system achieves wide-ranging results from a very sparse dataset for detecting damages to various parts of vehicles such as bonnets, trunks, and wheels. Comprehensive experimental results on the challenging vehicle damage detection have exhibited that our methodology of utilizing Mask R-CNN for this issue performs extraordinarily well for object classes and natural images. Further research is required to realize the application domains where Mask R-CNN can be utilized.

Acknowledgements We would like to acknowledge the contribution of all the people who have helped in reviewing this paper. We take immense pleasure to thank Prof. M. Kiruthika for her invaluable help, advice, and for providing the expertise that greatly assisted this research. We would also like to thank our families and friends who supported us in the course of writing this paper.

References

1. LeCun Y, Bengio Y, Hinton G (2015) Deep learning in nature. *Int J Sci* 521:436–444. <https://doi.org/10.1038/nature14539>
2. Alex K, Ilya S, Hinton G (2012) Imagenet classification with deep convolutional neural networks, In: *Advances in neural information processing systems*, pp 1097–1105. <https://doi.org/10.1145/3065386>
3. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis (IJCV)* 115(3):211–252. <https://doi.org/10.1007/s11263-015-0816-y>
4. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv* 1409.1556 [cs.CV]
5. Shelhamer E, Long J, Darrell T (2016) Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 39. <https://doi.org/10.1109/TPAMI.2016.2572683>
6. Lin T-Y, Dollar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection 936–944. <https://doi.org/10.1109/CVPR.2017.106>
7. He K, Gkioxari G, Dollar P, Girshick R (2018) Mask R-CNN. *IEEE Trans Pattern Anal Mach Intell.* 1–1. <https://doi.org/10.1109/TPAMI.2018.2844175>
8. Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: towards real-time object detection with region proposal networks 1–10. <https://doi.org/10.1109/TPAMI.2016.2577031>
9. Lin T-Y, Maire M, Belongie S, Bourdev LD, Girshick RB, Hays J, Perona P, Ramanan D, Dollr P, Zitnick CL (2014) Microsoft COCO: common objects in context. *ECCV*
10. Iglovikov V, Seferbekov SS, Buslaev A, Shvets A (2018) TernausNetV2: fully convolutional network for instance segmentation
11. Patil K, Kulkarni M, Sriramana A, Karande S (2017) Deep learning based car damage classification 50–54. <https://doi.org/10.1109/ICMLA.2017.0-179>
12. Maeda H, Sekimoto Y, Seto T, Kashiyama T, Omata H (2018) Road damage detection using deep neural networks with images captured through a smartphone. *arXiv:1801.09454* [cs.CV]
13. Johnson J (2018) Adapting Mask-RCNN for automatic nucleus segmentation
14. Couteaux V, Si-Mohamed S, Nempong O, Lefevre T, Popoff A, Pizaine G, Villain N, Bloch I, Cotten A, Boussel L (2019) Automatic knee meniscus tear detection and orientation classification with Mask-RCNN. *Diagn Interv Imaging* 100. <https://doi.org/10.1016/j.diii.2019.03.002>
15. Wu A, Zhang Q, Fang W, Deng H, Jiang S, Liu Q, Xia P (2018) Mask R-CNN based object detection for intelligent wireless power transfer. 1–5. <https://doi.org/10.1109/GLOCOMW.2018.8644387>
16. Dutta A, Gupta A, Zisserman A (2016) Vgg image annotator (via), <http://www.robots.ox.ac.uk/vgg/software/via>
17. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition at [arXiv:1512.03385](https://arxiv.org/abs/1512.03385) [cs.CV]
18. Abdulla W (2017) Mask r-cnn for object detection and instance segmentation on keras and tensorflow, <https://github.com/matterport/Mask-RCNN>
19. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) Tensorflow: a system for large-scale machine learning
20. Pessoa T, Medeiros R, Nepomuceno T, Bian G-B, Albuquerque VHC, Filho PP (2018) Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access* 1–1. <https://doi.org/10.1109/ACCESS.2018.2874767>

Chapter 55

Audio Tagging for Emotion Recognition: A Review



Raj Shah, Harshi Thaker, Shaurya Shettigar, Mahima Thakar
and Chetashri Bhadane

1 Introduction

Human speech is a combination of languages and sentiments. A machine is incapable of recognizing human emotions by both textual and vocal mediums. Emotion detection can play a huge role in improving the experience of the user and make machines more user-friendly through a healthy human-computer interaction. Speech-based investigations are setting and database subordinate. The great datasets can be speaker-dependent and speaker-autonomous. Varieties of speakers, their talking style, talking rate and substance of a discourse, are components that can fluctuate from individual to individual and influence acoustics.

Audio tagging is the detection and tagging of emotions within a speech sample. There are two different ways of representing emotions, they are through categorical metrics or through attributes. Emotions are classified categorically into seven basic emotions: neutral, happy, sad, angry, disgust, fear and boredom. Whereas while classifying through attributes we consider two major factors like Valence and Activation. Valence relates to the positive or negative or negative aspects of the emotion while activation is how aroused a person is going from being calm to excited. Dominance or power and expectation are a few other factors.

Currently, there is a dearth of non-invasive procedures that accurately help machines detect emotions within human speech. If done well it could greatly benefit human-computer interaction (HCI) everywhere such as in lie detection testing for criminal forensics, Automated helplines and call centres, early detection of psychological disorders, increasing the intelligence of socio-technical systems such as in the gaming industry and even making companion bots for the elderly.

There are various challenges in creating accurate SER engines:

R. Shah (✉) · H. Thaker · S. Shettigar · M. Thakar · C. Bhadane
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: rajsanchi@gmail.com

The primary one being that the datasets are mostly skewed towards the neutral emotion since there is insufficient genuine data for other emotions as most of the habitual interactions we have on a routine basis is neutral. There exist various datasets such as the Berlin dataset [10] and the Mandarin dataset, however, in order to create a comprehensive and higher quality SER than what exists today we would recommend creating a new synthetic dataset.

Another challenge is that most of the existing SER machines can easily detect the transient emotions of the speaker but sometimes these emotions are only temporary and tend to deceive the machine from the underlying long-term emotion. This along with incorporating the fact that there exists a cocktail of emotions within each speech sample is the reason why most SERs fall short of expectations.

2 Review of Literature

This paper [1] reports a trial study on six feelings: satisfaction, bitterness, outrage, dread, unbiased and weariness. The features are characterized by a neural network and a feeling name is allotted. The higher positioned highlights will be then selected for the classifier. The Berlin Database of Emotional Speech was utilized to extract the information. Feature vectors with 8, 12 and 16 features are utilized in three experiments. At the point when a contribution of eight best highlights was given, the best precision was the one which demonstrated how various arrangements of the features affected exactness. The general exactness of 77.1% was achieved in this investigation. There were 16 highlights considered. For this model, anger and neutral are the two feelings that can without much of a stretch be perceived while dread is the most troublesome one.

This [2] paper considers two dimensions, valence and activation. Anger, fear, surprise, disgust, sad and happy are the emotions considered here. Speaker variety, style, speaking rate and content of speech, are shown to have highly affected the acoustics. Emotions can be transient or long term and can be difficult to differentiate. Because of feelings like fear, anger and joy, the discourse delivered will be stronger, quicker, broad pitch run, normal high pitch and extraordinary high recurrence vitality; however, in sharpness, the speaker will have a minimal high repeat imperativeness and low-level moderate pitch. A variety of classification models have been used. After extracting the features, classification is achieved by comparing training file ‘.net’ and the feature of an untrained file.

This [3] paper proposes an RBFC approach that is utilized for division of signal and the outcomes are contrasted with other voiced division approaches. RBFC features were used for feature extraction and classification was done using support vector machine and neural networks with leave-one-out cross-validation. By physically improving the parameters and utilizing the radial basis work portion of the SVM the best precision of 71.2 and 72% with back-spread neural network and support vector machine-based grouping separately was achieved. This paper proposed new RBFC highlights for distinguishing proof of the passionate condition of human from

his/her discourse sign and giving promising outcomes. A one of a kind characteristic of this component is that it is impervious to intra and inter pitch varieties. The model classifies the neutral state of emotion the worst with about 55% accuracy and sadness with about 83% accuracy.

This [4] paper presents a robot used for human-robot interaction who is capable of classifying emotions into two categories: positive and negative. It is built for personal usage and two custom-made, distinct software subsystems are used, an emotion classification system and a behaviour system of the robot. Certain emotions are correlated with the values of some sound features of the human speech like tempo, amplitude and pitch. For example, when people are sad, they speak slower and have lower pitch. For building the classifier, 20 audio files were recorded. These files were recorded from the same person. A human was used who categorized the audio files into one of the two categories: positive and negative. After using cross-validation, an accuracy of 85% was achieved. The robotic arm is configured such that it acts according to the speaker's positive and negative emotions. This, however, leads to positive emotions being classified with greater precision than negative emotions. Negative emotions have a recall of 62.2%. The accuracy for all non-voiced intervals is 73.14%. Offline evaluation of similar systems gives accuracy between 75 and 95%. The correlation coefficient describing the similarity of the robot's classifier and a human's mental one is 0.743.

This [5] paper implements a speech emotion recognition engine using feature vector memory (or backward context). The dataset used is the German emotion database of 535 articulations by 10 speakers of seven feelings: unbiased, upbeat, dismal, irate, nauseate, dread and fatigue. For extraction of features, a successive arrangement of highlight vectors was produced per articulation and went to a measurable classifier. These features depend on a sign processing strategy like that utilized by the human sound-related framework. They have used a short-time Fourier transform to obtain a spectrogram for a discrete-time speech signal sample. Then they have further subdivided this spectrogram based on natural frequency divisions to humans. This strategy yielded a normal precision of 68% for the seven feelings in the 10-speaker test of the German Database, and a normal exactness of 91% for the 4-feeling test. This is an improvement to the 64% exactness made by people coordinated the marks in the database of eight on-screen characters and 15 feelings distributed by the Linguistic Data Consortium [6].

This [7] paper introduces an enhanced co-training calculation to use countless unlabeled discourse articulations for building a semi-supervised learning framework. It uses temporal features and statistical features of untagged sentences and their corresponding emotions to increase the sample size of a much compact set of labelled instances. This is done to reduce the skewness of the dataset. The examinations did here utilize two enthusiastic discourse corpora comprising of short words and their articulation in Chinese mandarin, one from a female and the other from a male. In every corpus, there are 300 unique sentences communicated in six feelings relating to outrage, fear, happiness, neutral, sadness and surprise, for an aggregate of 1800. The proposed framework makes 9.0% total enhancement for the female model and 7.4%

on the male model as far as normal exactness as compared to a standard supervised training algorithm.

This [8] is a pilot study that investigates a few measurable example methods to group articulations as per their emotional content. The features are broken into two feature sets; Feature Set A contains mean, standard deviation, minimum, maximum and range of the voiced pitch signal. Feature Set B contains about 17 features of the smoothing cubic splines of the pitch contour. This contains statistics related to rhythm, smoothed pitch signal, etc. The classifiers used were maximum likelihood Bayes classifier (MLB), kernel regression (KR) and K -nearest neighbours in order to compare their performances regarding pattern recognition. Then in order to tackle the curse of dimensionality problem, various majority voting of specialists' algorithms are used. Among these cooperative compositions (CC) yields an optimized master classifier. Thus, this paper successfully considers two unique aspects of pattern recognition techniques:

- Using majority voting of subspace specialists decreases the error of the system significantly
- The set of features on the spline, i.e. Feature Set B contains enough information to make the classification system as good as human performance.

In this [9] study, different approaches were utilized to remove engaging qualities of the sign. These were then prepared by a neural system classifier with four of the six feelings of the Berlin Database [10]: happiness, anger, fear and sadness. At that point, trained feelings were set in the outline by applying a correspondence between the mark, which demonstrates a trained feeling and the edge places of emotions provided by the experimentation of Russell. For instance, joy is assigned in the plane in the angular position of 34° . Then, all signs of corpus admitted to this algorithm interpreted by the classifier as happiness will be located at 34° in the polar diagram. The best feeling recognized for this classifier ended up being sadness with an exactness of 96.67%. Fear was the worst, recognized with 70%. The attributes utilized identifying with the worldly, frequential and prosodic methodologies were compelling in depicting the four feelings prepared with a triumph rate of 95%.

3 Analysis

We have analysed and summarized the different papers here. The attributes regarding the various parameters and their differences have been noted below in the tabular form (Table 1).

Table 1 Summary of literature review

Ref. No.	Features used	Classification method	Database and accuracy	No. of classes recognized
[1]	F0 and F1 signal features	Neural network classifier	Berlin database of emotional speech (77.1%)	Anger, boredom, fear, sad, happy, neutral
[2]	Intensity, fundamental frequency (F0), spectral contour, shimmer voice quality, timing, eloquent, pitch, jitter, energy	ANN, SVM (RBF), HMM	Danish emotional database, Berlin emotional database and eight others	Anger, fear, surprise, disgust, sad and happy
[3]	MFCC, relative amplitude, LPCC, GRNN and SFS	SVM (RBF) and neural network	Berlin database of emotional speech (72%)	Anger, boredom, disgust, fear, joy, sad and neutral
[4]	Tempo, amplitude's mean and maximum and pitch's maximum and deviation	Neural networks and SVM	The accuracy achieved was 70%	Anger, boredom, disgust, fear, joy, sad and neutral
[5]	Short-time Fourier transform, regression of wave	SVM, backpropagation	German emotional database (68%), 4 emotion sample (91%)	Neutral, happy, sad, angry, disgust, fear and boredom
[7]	Means, standard deviations, maximums and minimums of F0, Delta F0, log energy, first and second linear prediction cepstral coefficients (LPCC)	Enhanced co-training algorithm (HMM, multi-SVM)	1800 Chinese Mandarin utterances (75.87% for females and 80.93% for males)	Anger, fear, happiness, neutral, sadness and surprise
[8]	About 20 features including mean, standard deviation, minimum, maximum and range, rhythm, smoothed pitch signal, etc.	Maximum likelihood Bayes (MLB), kernel regression (KR), KNN and cooperative compositions	1250 training utterances (error rate of 20.5%)	Happy, sad, anger and fear

4 Conclusion and Future Scope

We have thus reviewed multiple papers and different methodologies for analysing different speech emotion recognition algorithms.

However, for broadening the scope we aim to implement a comprehensive user-friendly software that can also handle dynamic speech processing and event handling for various businesses and clients who can leverage our models for data mining purposes across sectors such as assessing customer satisfaction, distress detection among helpline callers, lie and fraud detection, etc.

Another possible improvement would be to generate our own synthetic dataset since most of the datasets are not only applicable for specific languages like German and Mandarin but are also heavily skewed since natural conversations have a disproportionately high amount of neutral words which are not ideal for training a model.

References

1. Soltani K, Ainon RN (2007) Speech emotion detection based on neural networks. In: 2007 9th international symposium on signal processing and its applications, Sharjah, 2007, pp 1–3. <https://doi.org/10.1109/ISSPA.2007.4555476>
2. Yerigeri VV, Ragha LK (2017) Marathi speech emotion detection: a retrospective analysis. In: 2017 8th international conference on computing, communication and networking technologies (ICCCNT), Delhi, pp 1–6. <https://doi.org/10.1109/icccnt.2017.8203925>
3. Kudiri KM, Verma GK, Gohel B (2010) Relative amplitude-based features for emotion detection from speech. In: 2010 international conference on signal and image processing, Chennai, pp 301–304. <https://doi.org/10.1109/icsip.2010.5697487>
4. Kirandziska V, Ackovska N (2012) Human-robot interaction based on human emotions extracted from speech. In: 2012 20th telecommunications forum (TELFOR), Belgrade, pp 1381–1384. <https://doi.org/10.1109/TELFOR.2012.6419475>
5. Guven E, Bock P (2010) Speech emotion recognition using a backward context. In: 2010 IEEE 39th applied imagery pattern recognition workshop (AIPR), Washington, DC, pp 1–5. <https://doi.org/10.1109/AIPR.2010.5759701>
6. Liberman M et al (2002) Emotional prosody speech and transcripts LDC2002S28. Web Download. Philadelphia: Linguistic Data Consortium
7. Liu J, Chen C, Bu J, You M, Tao J (2007) Speech emotion recognition using an enhanced co-training algorithm. In: 2007 IEEE international conference on multimedia and expo, Beijing, pp 999–1002. <https://doi.org/10.1109/icme.2007.4284821>
8. Dellaert F, Polzin T, Waibel A (1996) Recognizing emotion in speech. In: Proceeding of fourth international conference on spoken language processing. ICSLP'96, vol 3. Philadelphia, PA, USA, pp 1970–1973. <https://doi.org/10.1109/icslp.1996.608022>
9. Bustamante PA, Lopez Celani NM, Perez ME, Quintero Montoya OL (2015) Recognition and regionalization of emotions in the arousal-valence plane. In: 2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC), Milan, pp 6042–6045. <https://doi.org/10.1109/EMBC.2015.7319769>
10. Burkhardt F, Paeschke A, Rolfes M, Sendlmeier W, Weiss B (2005) A database of German emotional speech. In: 9th European conference on speech communication and technology, pp 1517–1520. 5

Chapter 56

Implementation of ROS in Drones for Animate and Inanimate Object Detection



**Chinmay Sankhe, Bhavesh Ahuja, Austin Coutinho, Chandan Bhangale
and Nupur Giri**

1 Introduction

The algorithm used here (YOLOv3) is faster with a better detection rate and confidence than any traditional object detection algorithm. Also, existing systems are aimed at merely providing all the video data as it is and does not draw focus to suspicious activities in the captured recordings or live stream returned by FPV camera of the drone. They try to reduce the semantic gap and try to extract low-level features. We aim at developing AI-based scene understanding models which will be able to employ artificial intelligence to its full power to get real-time information from the battleground, make accurate predictions of the scene by noise reduction and elimination of redundant features. The basic idea can be summarized—imagine a drone flying over a particular area given by the user. The drone will capture video using its bottom camera and will mark boundaries over objects and name them. This object detected video will be shown to the user in the control room. The video of a particular period of time can be summarized into text using the concept of scene understanding. Using its front camera, the drone would navigate the way by detecting obstacles and dodging them.

C. Sankhe (✉) · B. Ahuja · A. Coutinho · C. Bhangale · N. Giri
V.E.S.I.T., University of Mumbai, Mumbai, India
e-mail: 2016.chinmay.sankhe@ves.ac.in

B. Ahuja
e-mail: 2016.bhavesh.ahuja@ves.ac.in

A. Coutinho
e-mail: 2016.austin.coutinho@ves.ac.in

C. Bhangale
e-mail: 2016.chandan.bhangale@ves.ac.in

N. Giri
e-mail: nupur.giri@ves.ac.in

2 Motivation

In India, the defense field is very large. For guarding the border areas, employees seating at the control room will have to continuously track the activities happening manually. This may increase the workload of the employees. As the whole work is manual, it may lead to human error. But in the security or in defense such human errors are very dangerous as it may lead to a threat to a country. So we aim to develop a reconnaissance system that employs, recognizes models and minimizes the burden of analysis by providing insights about the field, and also to train and test the model in a simulated immersive environment of similar reconnaissance activities as a prototype to target out enemies, identify equipped weapons, vehicle units and so on. The model, once developed, could be further used as a part of a comprehensive system of other modules to inculcate the concept of human-in-the-loop in the military system for surveillance missions. Also, we develop ROS packages such as navigation, landing and object understanding that would help the drone to navigate and plan missions in automation mode.

3 Literature Review

The authors of [1, 2] focus on the significance of incorporation of rudimentary laws of physics and intuition as innate features of currently existing neural network-based recognition models. References [3, 4] provides a method of identifying and recognizing complex recurrent actions in video sequences by using the concept of trees to store each action and sub-actions to understand the real world. Reference [5] makes use of depth feature apart from the traditional RGB images for object detection and segmentation. These systems use human activity recognition techniques that are locally invariant. The robustness of these drones is explained along with its ambiguity and processing performed in them [6, 7].

References [8, 9] provide various innovative ways of identifying human–motor interaction, and the former formulates a visual–motor language used for identifying common motor activities providing various linguistics and insights on it. The former proposes a view-based approach to identify these actions when clear facial features are not visible based on the emulation of their gait information in frame to exemplar distance vector and training it on the hidden Markov model. Most of the applications of these are in surveillance, human–machine interaction, pose estimation and biometrics. Various surveillance tasks have been studied using unmanned aerial vehicle [10]. Reference [11] shed light on the methods deployed by DARPA and other international leading defense-based organizations to tackle the problem of analyzing the copiousness of surveillance data available at their disposal. Various factors like visual scenes, reasoning and general knowledge used for human action recognition in [12] for actions like running, walking or jumping. The proofs for

declarative semantics for logical programs have been explained in detail. These formulations are performed using simple negation [13]. Using 2D CNN (Convolution Neural Network), only limb movement is recognized through this approach. This helps in activity recognition [14].

References [15, 16] bring forth the plethora of ways to optimize performance of surveillance systems in autonomous vehicles like moving the computation to the cloud, make do with lack of GPS in remotely situated areas, etc. [17, 18] manifest the latest developments toward the problem of scene understanding beyond mere detection of elements in an image. The magnitude and direction are calculated using the Euclidian method and tan inverse function for every video frame to identify actions [19]. Uncertainty in output is an integral part of a project. This needs to be minimized to a huge extent to make an accurate system. Prolog is designed for understanding these uncertainties in login and quantity [20]. References [21, 22] provide a method of identifying and recognizing complex recurrent actions in video sequences by using the concept of trees to store each action and sub-actions to understand the real world.

4 Methodology Used

4.1 Data Assembling

In this, we gathered the images using source like ImageNet in the form of images. ImageNet also contains COCO dataset, which is extensively used for object detection, segmentation and captioning of data.

4.2 Data Preprocessing and Augmentation

In this step, we increase the volume of processed data by intensity variations, rotations, cropping, etc. so that models can be trained up to high accuracy.

4.3 Developing the Model Using YOLOv3

Here, we developed a YOLOv3 model for detection and classification. This model is applied to an image at different regions. Based on the score of regions, the detectors have been considered.

4.4 Training the Model

We train the devised models for a large number of epochs on the acquired data and set up parameters. The function which we have used for activating neural network is used for finishing layers. The rectified linear activation method is used for all other layers as shown below:

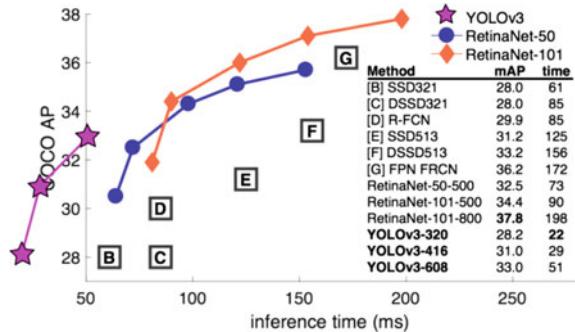
$$\Phi = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

Optimization using sum of squared error is easy for our model, but the main aim of maximizing average precision is tough. Localization error and classification error have equal weightage which is not ideal for our case. Confidence score decreases to zero in image cells with no objects, which overpowers gradient with those cells containing objects. This could cause model instability, inflicting training to diverge too soon. We increase loss from the predicted box coordinate and reduce loss from probability predictions from objectless boxes. We consider two parameters which are α_a and α_b to achieve such a task. α_a is set to 5.0 and α_b is set to 1/2. Error in big and tiny boxes is the same as the sum squared error. Little error deviations do not matter much in big bounding boxes. Such an error metric needs to be designed. We find half power of the breadth and length of the rectangular pack for our error metric. During training, a single bounding box is selected from the multiple boxes predicted by YOLO in each rectangular cell. The box with the highest Intersection over Union (IoU) is chosen as our predictor. Our bounding box predictors now specialize in estimating some classes, dimensions and aspect ratios of objects. This revises overall recall. The multi-part loss function is optimized during training as follows:

$$\begin{aligned} & \alpha_a \sum_{i=0}^{s^2} \sum_{j=0}^B \left\| \begin{smallmatrix} abj \\ ij \end{smallmatrix} \right\|^2 \left[(X_i - \hat{X}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \alpha_a \sum_{i=0}^{s^2} \sum_{j=0}^B \left\| \begin{smallmatrix} abj \\ ij \end{smallmatrix} \right\|^2 \left[(\sqrt{W_i} - \sqrt{\hat{W}_i})^2 + (\sqrt{h_i} + \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{s^2} \sum_{j=0}^B \left\| \begin{smallmatrix} abj \\ ij \end{smallmatrix} \right\|^2 (C_i - \hat{C}_i)^2 + \alpha_b \sum_{i=0}^{s^2} \sum_{j=0}^B \left\| \begin{smallmatrix} abj \\ ij \end{smallmatrix} \right\|^2 (C_i - \hat{C}_i)^2 \\ & + \sum_{j=0}^{s^2} \left\| \begin{smallmatrix} abj \\ ij \end{smallmatrix} \right\|^2 \sum_{c \in \text{classes}} (p_i(c) - \hat{p}(c))^2 \end{aligned}$$

The model being used for training is YOLOv3. It has a very less inference time, i.e., it begins at the time when an image is detected. As shown in Fig. 1, the starting time for other models is far more than our model, while the starting time for our

Fig. 1 Differentiation of YOLOv3 with commonly used models w.r.t. inference time



used model is 22 ms. A shift of origin is applied to mark the inference time for other models too. The below graphs are compared by training models on COCO dataset.

4.5 Deployment and Analysis in Real-Life Scenario

The full image is processed by implementing a single neural network. Then, the network splits given image into different regions. Each region has its own bounding boxes and probabilities. Weightage is given to these bounding boxes using the predicted probabilities. The predictions are compared with a threshold value to find high-scoring predictions. The trained and tested vehicle and weapon detection model will be deployed in a real-life scenario and will be leveraged for further improvement in the methodology. We also developed ROS for automation of drone, precision landing and simulation of drone using gazebo_ros_pkgs.

4.6 ROS Packages

See Fig. 2.

Obstacle Avoidance Package: Drone should be able to detect obstacles in its path using LIDAR sensors and control its flight speed accordingly and also plan its move in the right direction by rotating a specific degree and thus avoid crashes and enable smooth flight when made to run on automation mode.

Shortest Path Planning Package: Navigation of drone from position A to position B in automation mode should use the shortest path considering all the obstacles in its path and also all the waypoints. It would also be useful if there are specific safe flying zones and some danger/restricted zones.

ArUco Marker Value Package: This ROS package uses the ArUco Library of OpenCV to compute values of ArUco Markers returned by the raw image stream obtained by subscribing to the topic generated by bottom FPV camera of drone.

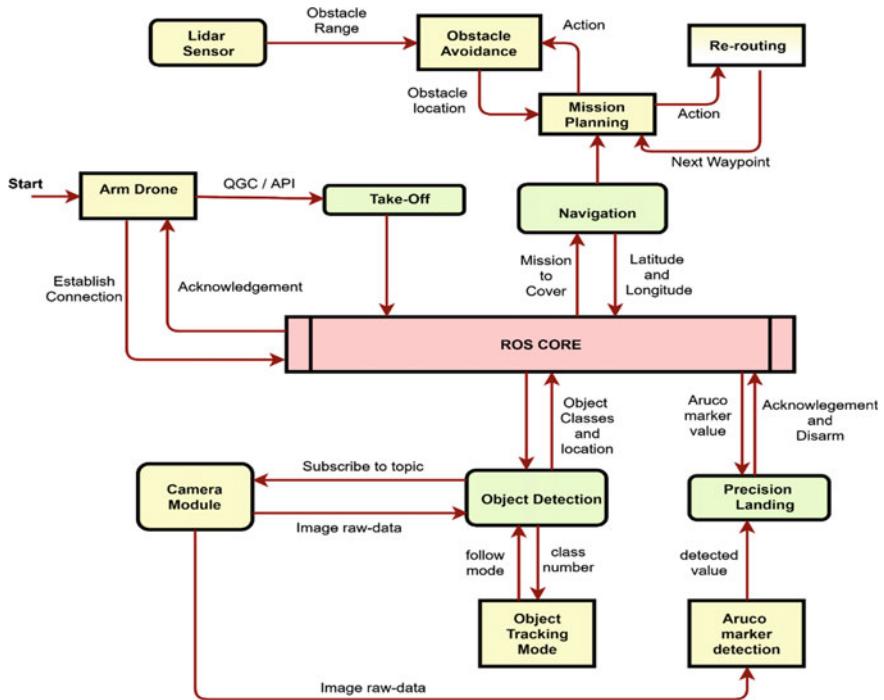


Fig. 2 State transition diagram for ROS packages

Precision Landing Package: ROS package for making the drone lands precisely on ArUco markers placed at its final destination. The drone uses its bottom FPV camera and returns the image raw data via a topic that can be subscribed by ArUco marker value package.

5 Results

5.1 Detection Output

We have deployed our object detection YOLO model and have done the markings on the data collected by the drone. We can see below that it detects the soldiers and cars. Figure 3 shows the simulation of an environment consisting of vehicles and humans. The object detection models detect a soldier and a car in the below image.

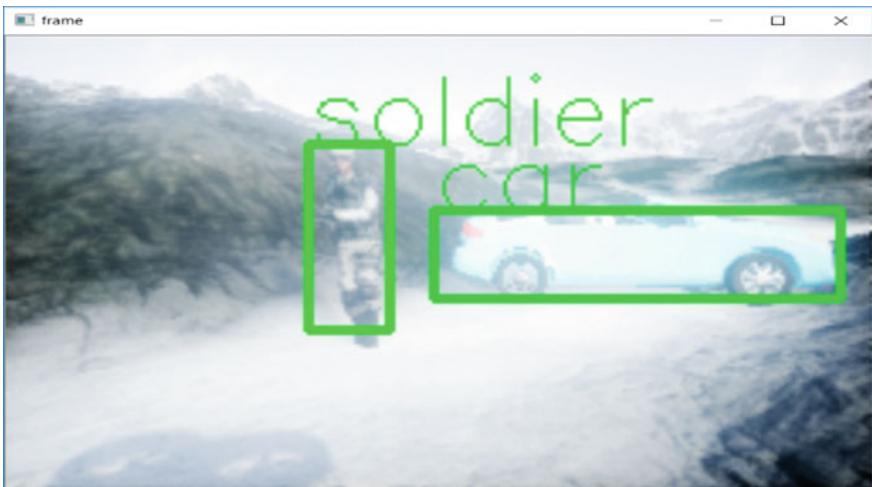


Fig. 3 Detection of objects in virtual simulator

5.2 Performance Comparison Table

YOLOv3 is 3 times faster than SSD, while both have the same AP metric. The AP with IoU has a value of 0.75 which decreases significantly; thus, it is not better than RetinaNet's AP. YOLOv3 has higher localization error, but the detection of small objects has been improved.

5.3 Success Rate Graph

YOLOv3 has a success rate of 0.54 which is higher than any other mentioned below NN models. Refer Fig. 4 for comparison of YOLOv3 with other object detection models. The X- and Y-axes have been plotted with parameters as shown:

5.4 Precision Graph

Since YOLOv3 is fast, it is less precise than CNN-SVM model with precision about 0.73 (refer Fig. 5). The graph is plotted with the location error threshold on the X-axis and precision on the Y-axis.

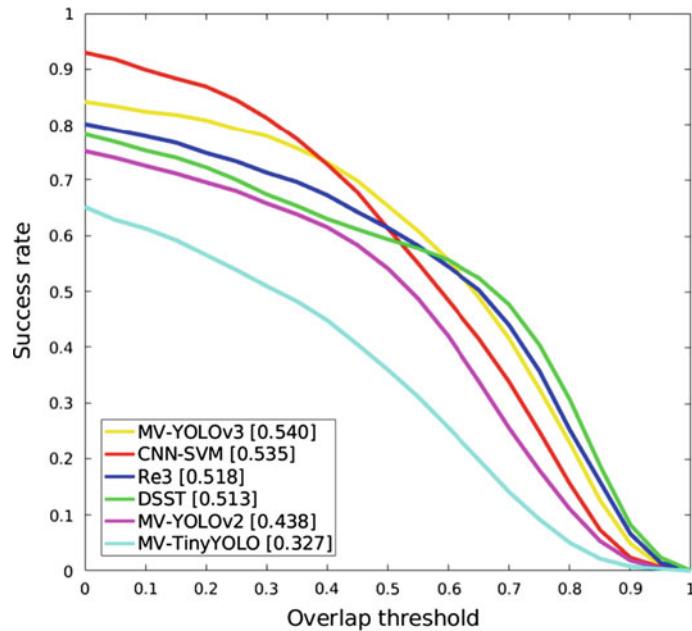


Fig. 4 Success plots of one-pass evaluation

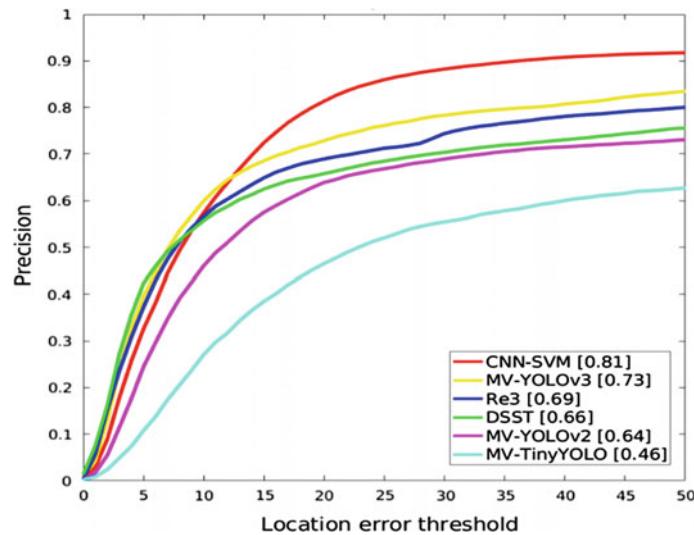


Fig. 5 Precision plots of one-pass evaluation

5.5 ArUco Marker Landing Using ROS Packages

An ArUco marker is placed at the site of landing for the drone to land itself. The landing is quite precise and accurate in most cases unless the marker is not clear or hidden by shadows. The landing is automatic and hence needs to be precise (refer Fig. 6).

6 Conclusion

The dataset was collected through various sources for training our model. The model was trained on 95% data and tested on the remaining 5% data. Again, the model was tested on video and live stream inputs giving correct results. Then, the simulation was performed on Unreal Engine 4. Characters were compiled using Autodesk Maya and imported on Unreal Engine project. Then, physics and automation were added so that characters could stand and walk randomly on their own in the map. The model was implemented in the project. A simulated drone's video stream was used to classify objects successfully. The drone was controlled using DroidJoy remote control software. Then, ROS modules were implemented including automatic takeoff and precision landing.

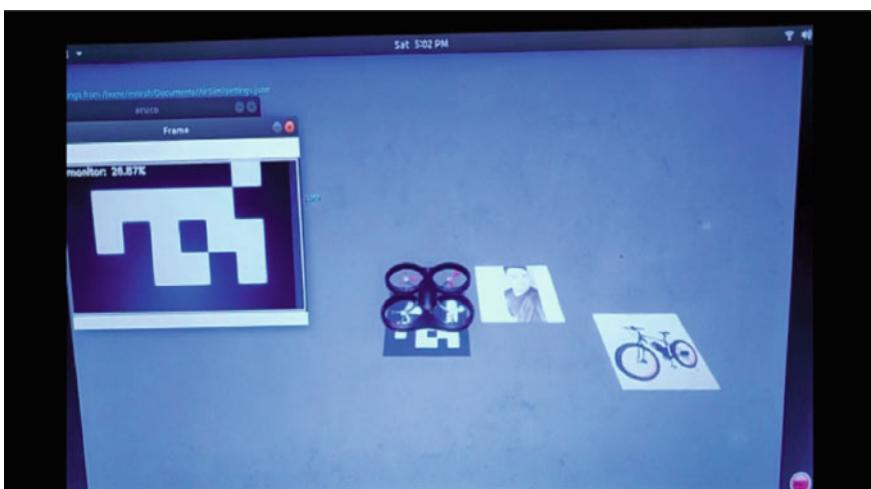


Fig. 6 Precision landing of the drone on the ArUco marker. We can see that the landing position is quite accurate

7 Future Scope

7.1 Object Track and Follow

Vision-based automated object tracking and the following can be configured on any custom drone.

7.2 Scene Understanding in an Area

The drone can move around through specific points, come back to the start point and complete the mission. It should be able to understand the scene of the complete area.

7.3 Identification of Shadows

A module can be added for shadows to detect them independently and differentiate it from actual objects.

7.4 Implementation of Black Box

In case of accidents or attacks, if the drone is damaged, the system should be able to backtrack the details of the last few minutes.

References

1. Battaglia PW, Hamrick JB, Tenenbaum JB (2013) Simulation as an engine of physical scene understanding. *Proc Natl Acad Sci*
2. Riochet R, Castro MY, Bernard M, Lerer A, Fergus R, Izard V, Dupoux E (2018) IntPhys: a framework and benchmark for visual intuitive physics reasoning. In: Computer vision and pattern recognition, June 2018
3. Summers-Stay D, Teo CL, Yang Y, Fermüller C, Aloimonos Y (2012) Using a minimal action grammar for activity understanding in the real world. In: IEEE/RSJ international conference on intelligent robots and systems, October 2012
4. Teo CL, Yang Y, Daumé H, Fermüller C, Aloimonos Y (2012) Towards a Watson that sees: Language-guided action recognition for robots. In: IEEE international conference on robotics and automation, May 2012
5. Gupta S, Girshick R, Arbeláez P, Malik J (2014) Learning rich features from RGB-D images for object detection and segmentation. In: European conference on computer vision (ECCV), July 2014

6. Ben-Arie J, Wang Z, Pandit P, Rajaram S (2002) Human activity recognition using multidimensional indexing. In: IEEE Trans Pattern Anal Mach Intell
7. Carberry S (2001) Techniques for plan recognition. In: User modeling and user-adapted interaction, 2001
8. Guerra-Filho G, Fermuller C, Aloimonos Y (2005) Discovering a language for human activity. In: Proceedings of AAAI, University of Maryland, 2005
9. Kale A, Sundaresan A, Rajagopalan AN, Cuntoor NP, Roy-Chowdhury AK, Kruger V, Chellappa R (2004) Identification of humans using gait. IEEE Trans Image Process
10. Baral C, Gelfond M, Rushton N (2008) Probabilistic reasoning with answer sets. In: Logic in computer science, December, 2008
11. Mohta K, Watterson M, Mulgaonkar Y, Liu S, Qu C, Makineni A, Saulnier K, Sun K, Zhu A, Delmerico J, Karydis K, Atanasov N, Loianno G, Scaramuzza D, Daniilidis K, Taylor CJ, Kumar V (2015) Fast, autonomous flight in GPS-denied and cluttered environments. J Field Robot
12. Santofimia MJ, Martinez-del Rincon J, Nebel JC (2012) Common-sense knowledge for a computer vision system for human action recognition. In: Ambient assisted living and home care. Springer
13. Gelfond M, Lifschitz V (1998) The stable model semantics for logic programming. MIT Press
14. Garvila DM (1999) The visual analysis of human movement: a survey. Comput Vis Image Underst
15. Tunick A, Meyers RE (2017) Developing scene understanding neural software for realistic autonomous outdoor missions. US Army Research Laboratory
16. Mohta K, Sun K, Liu S, Watterson M, Pfrommer B, Svacha J, Mulgaonkar Y, Taylor CJ, Kumar V (2018) Experiments in fast, autonomous, gps-denied quadrotor flight, June 2018
17. Bondi E, Kapoor A, Dey D, Piavis J, Shah S, Hannaford R, Iyer A, Joppa L, Tambe M (2017) Near real-time detection of poachers from drones in AirSim. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, 2017
18. Omran M, Lassner C, Pons-Moll G, Gehler PV, Schiele B (2018) Neural body fitting: unifying deep learning and model-based human pose and shape estimation. Comput Vis Pattern Recognit (August)
19. Chaudhry R, Ravichandran A, Hager G, Vidal R (2009) Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: IEEE CVPR workshops, 2009
20. Baral C (2011) Logic programming and uncertainty. In: International conference on scalable uncertainty management, 2011
21. Kautz HA (1991) A formal theory of plan recognition and its implementation. In: Reasoning about plans
22. Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3D tracking of hand articulations using Kinect. In: Proceedings of the British machine vision conference, BMVA Press, 2011

Chapter 57

Modeling CNN for Best Parameter Investigation to Predict Viable Exoplanets



Gaurav Singh, Sarang Gawane, Amandeep Prasad and Kalpita Wagaskar

1 Introduction

The Kepler spacecraft missions were launched in the year 2008 by NASA as a part of deep space imaging and research work with its fundamental objective being detecting signals of extra solar planets by focusing their high-resolution cameras on outer star systems. The observations produced immense amount of data [1, 2] analyzing which was an insurmountable task if done manually. Hence, various techniques were developed to automate the process. Many of these earlier vetting techniques were developed by NASA itself. The most notable being the Robovetter and Autovetter. The Autovetter is a machine learning system for classifying a specific type of signals called Kepler threshold crossing events (TCEs) into three different categories. Autovetter has set a benchmark for identification of TCEs [3, 4]. The Robovetter on the contrary is not a machine learning system. Here, the heuristics have been explicitly defined by humans instead of automatically learning from data [5]. The Autovetter and the signal detection using random forest algorithm (SIDRA) are the two main detection techniques that employ the random forest algorithm for transit detection. The SIDRA for the general case uses around 100 decision trees in order to achieve an accuracy of about 90% [6].

The research was commenced with the testing of the datasets in the conventional order of increasing complexity in other machine learning models. Both linear and logistic regressions fail to produce optimum results owing to the comparatively higher

G. Singh (✉) · S. Gawane · A. Prasad · K. Wagaskar
Don Bosco Institute of Technology, University of Mumbai, Kurla, India
e-mail: gauravisvirtual@yahoo.com

S. Gawane
e-mail: saranggawane@gmail.com

K. Wagaskar
e-mail: kalpitags@gmail.com

nonlinearity present in the datasets. The datasets used in this work for the purpose of training retrieved from the archives of the NASA exoplanet missions archive in the form of light curve as termed by astronomers. A light curve is the graph of the intensity of the star system plot against progressive time. This work recognizes the light curve as an image which contains the scatterplot of the flux values plotted against time. On further testing and background research with dense networks, it was realized that an analogous relation could be established between the CNN models and computer vision models where a massive amount of image processing and learning was being performed [7]. By analogy, it was easy to draw parallels as to what type of network structure would suit best for this research. It was found that CNNs performed exceptionally well in image-related learning processes. They not only ramped up the precision but due to the selective nature of internal connections significantly mitigated the processing time and at the same time enhanced the efficiency of the model.

The conjecture was cemented by an almost identical research paper and the fundamental source of inspiration for the paper by Shallue and Vanderburg [8]. Their research categorically demonstrated the distinct shortcomings of the dense networks. This paper intends to capitalize this research and forward the proposition of the irrefutably appealing efficiency of CNN-based models for the classification of TCEs in the Kepler archives. The following sections will demonstrate the construction of a rudimentary piece of conventional convolutional logic which also acts as a prototype for the conceived paper.

2 Background Information

2.1 *Introduction to Convolutional Neural Network*

Fully connected neural networks connect every neuron in layer ' n ' to every neuron in layer ' $n + 1$ '. Every pixel in a 2D input image would be treated independently without using the fact that some pixels are located near each other. Features that are composed of neighboring pixels, like edges and shapes, need to be learned independently by the model for each location in the input.

On the contrary, CNNs exploit spatial structure by learning local features that are detected across the entire input. Each feature is acquired and learnt only once. This reduces the number of parameters that are required to be learnt, hence reducing the memory usage and number of computational operations required to compute the output.

2.2 Mathematical Description

A CNN comprises of convolution layers and pooling layers. The input to a (one-dimensional) (k) convolution layer is a set of K vectors $a_{i-1}^{(k)}$ ($k = 1, 2, \dots, K$) of length n_{i-1} , and the output is a set of L vectors $a_i^{(l)}$ ($l = 1, 2, \dots, L$) of length n_i . The operation that takes the set of K input vectors to the l th output vector is called a feature map and is defined by the operation [8]:

$$a_i^{(l)} = \phi \left(\sum_{k=1}^K w_i^{(k,l)} * a_{i-1}^{(k)} + b_i^{(l)} \right)$$

where $*$ is the discrete cross-correlation operation—convolution, $w_i^{(k,l)}$ is a vector of length m_i of learnt parameters called the *filter*, $b_i^{(l)}$ is a vector of length n_i of learnt bias parameters and ϕ is an element-wise activation function. Typically, the kernel size is small (e.g., $m_i = 3$ or 5) and the feature map detects the presence of a local feature along its input [8].

A pooling layer aggregates values within small neighborhoods by taking the mean or maximum value within each neighborhood. The regions are spaced ‘ s ’ neurons apart from one neuron, where s is called the *stride length*. The network hence becomes invariant to small translations of the input. This makes the number of neurons in the pooling layer much less.

3 Solar System Dataset Representation

The training dataset of stars was derived from the Exoplanet Hunting in Deep Space. The data presented therein is cleaned and derived from observations made by the NASA Kepler space telescope. Over 99% of this dataset belongs to Campaign 3.

The data describes the change in flux (light intensity) of several thousand stars. Each star has a binary label of 2 or 1. ‘2’ indicates that the star is confirmed to have at least one exoplanet in orbit; some observations are in fact multi-planet systems. And ‘1’ states that the light curve is not confirmed to contain an exoplanet yet.

4 Data Representation in Exo-Drive

4.1 Training Set Description

The dataset is retrieved from Kaggle competition [9], which comprised 5087 observations with 3198 features.

Feature 1 is the label vector. Features 2–3198 are the flux values over time. In the dataset, there are 37 confirmed exoplanet stars and 5050 non-exoplanet stars.

4.2 Testing Dataset

The testing dataset is also taken from Kaggle competition [9] which has 570 observations and 3198 features. Feature 1 is the label vector. Features 2–3198 are the flux values over time. The testing dataset consists of 5 confirmed exoplanet stars and 565 non-exoplanet stars.

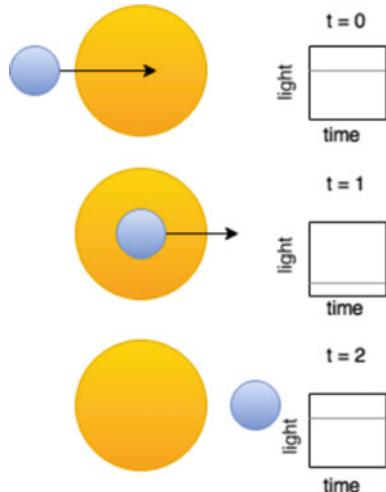
4.3 Reason for Flux Variation

The star is the main source of light within any solar system. If a given star is observed over several months or years, there may be a regular ‘dimming’ of the flux (the light intensity) [8]. This is evidence that there may be an orbiting body around the star; such a star could be considered to be a ‘candidate’ system of possible exoplanets.

Flux variation is the key grab of the research. The candidates predicted with >90% probability by the system can be confirmed as an exoplanet candidate with the help of sophisticated methods like gravitational lensing, radial velocity measurement, etc (Fig. 1).

In the above diagram, a star is orbited by a blue planet. At $t = 1$, the starlight intensity drops because it is partially obscured by the planet, given its position. The

Fig. 1 Generation of light curves [12]



starlight rises back to its original value at $t = 2$. The graph in each box shows the measured flux (light intensity) at each time interval.

4.4 Input Representation

The input to the system is given in the form of an image of light curve graph. The graph is scatterplot by nature and contains only the curve as shown in Figs. 2 and 3. Label 2-type light curve represents the star system with exoplanets, and Label 1 represents the star systems with 0 exoplanets.

The plots of both types are generated from the numerical data.

5 Methodology

The dataset imported from Kaggle serves as the basis for the model to train and test. The datasets were directly used to create a scatterplot images which were stored accordingly. The graphs were stored in different locations on the basis of their type and nature. For example, the *Light_curve* folder stored scatterplots divided into two subfolders, namely *train* and *test*. As the name suggests, *train* subfolder contained the training dataset and the *test* folder contained the testing dataset. The *train* and *test* folders contained the two subfolders each, containing images of the confirmed and non-confirmed class. In order to reduce the computation and see the limits of our model, we used only a subset of the actual training dataset to train our model. Hence, the training was done on 195 observations. Thirty-six observations were dedicated for confirmed class and rest 141 for the non-confirmed class.

The images were directly imported from these locations, and then they were converted to a multidimensional array. The array contained the pixel values, representing

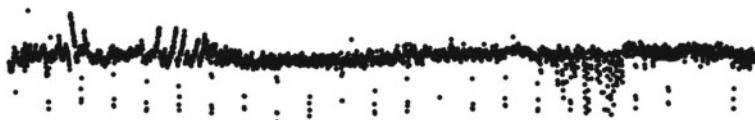


Fig. 2 Label 2 light curve for confirmed planet



Fig. 3 Label 1 light curve for non-confirmed planet

the image in its entirety. Due to the large number of images, present in the system for training, divide-and-conquer approach was used. That is, the images were imported in a batch_size of 10, so that the main memory could easily bear the load of the imported images.

5.1 Exo-Drive Architecture

The prototype implemented for the testing phase of the research used the Keras Python library. Various models were generated by regulating the hyperparameters comprising of number of input nodes, the size of the filter in the convolution layer, number of convolutional layers, the size of input datasets and activation functions. The aforementioned hyperparameters were the primary variables for the prototype. The intent of this exercise was to examine and select the most appropriate model for obtaining optimum results while training. The results of every model, for the test dataset, were collected just after the first epoch of training.

The main aim of the training module was to work with most simple architecture and yet provide the most efficient results. The solution came in the form of LeNet-5 architecture, which formed the basis for the CNN design. LeNet-5 is a relatively very simple network [10]. Inspired by that architecture, this work aims at proposing the Exo-Drive design.

Design Configuration

The proposed model comprises of five layers, among which there are two convolutional layers (C1 and C3), two subsampling (pooling) layers (S2 and S4) and one fully connected layer (F6), that are followed by the output layer. C1 convolutional layer uses 5 by 5 convolutions with stride 1, and C3 uses 3 by 3 convolution with stride 1. Subsampling layers are 2 by 2 max pooling layers. ReLU activations are used throughout the network, with the exception of the last layer where sigmoid function is used. ReLU activations were chosen because they have been proven to perform better as compared to tanh (which is even better than sigmoid for multilayer neural networks) [11].

The convolutional kernel of the layer C3 does not use all of the features produced by the layer S2. One reason for doing that was to make the network computationally less demanding. The other reason was to make convolutional kernels learn different patterns. If different kernels receive different inputs, then only will they be able to learn different patterns.

Designs Work Efficiency

LeNet-5 was able to achieve a loss value <1 and accuracy ~98% on the MNIST dataset, which was very close to the state of the art at the time (produced by a boosted ensemble of three LeNet-4 networks). In this paper, the proposed modified version of LeNet-5 was able to achieve a loss value ~0.25. The accuracy on the other hand hits ~93% on the provided test dataset. Note that the environment, dataset and

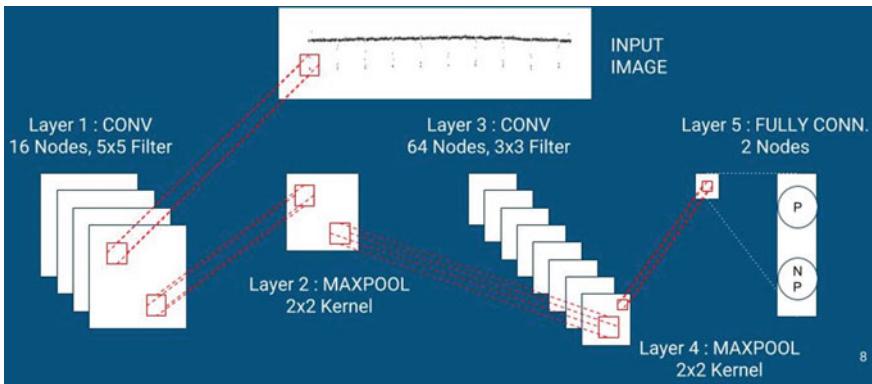


Fig. 4 Exo-drive architecture for training the Kaggle dataset

the use case were completely different from the original LeNet-5. However, a simple comparison based on the accuracy and loss allowed us to understand the capabilities of the proposed model in contrast to the LeNet-5 architecture (Fig. 4).

Since the model proposed in this paper is inspired from the LeNet-5, a lot of similarity can be found in between them, for example, the presence of max pooling layer immediately after every convolution layer and the presence of a single fully connected layer in the end. However, the place where the model becomes different is in terms of the activation function used in the early layers, i.e., ‘ReLU’ instead of ‘tanh’.

6 Experimentation

LeNet-5 was tuned for the MNIST dataset; therefore, there was a need to experiment with the hyperparameters that would give the best result for the light curves. Hence, experimentation began with the no. of layers, nodes per layer and the filter size for each layer.

In order to focus more on the hyperparameters, this paper reduced the number of images that would be used to train. Thus, the training set was dropped to mere 195 images as compared to the original ~5000 images. The focus here was on the parameters which would give the best results despite the lower number of images to train with.

6.1 Training Environment

The entire exhaustive training regimen was loaded upon the Intel Core i5 CPU systems. However, the exhaustive list of hyperparameters were trained and tested in a GPU-less environment. Thus, every single iteration of every single possible parameter set was worked out completely by CPU-only machines. A given parameter set on such a machine could easily take from 20 to 400 s for a single epoch.

The entire algorithm for the model was implemented in Python 3 with Ubuntu OS support in the background. Python 3 was chosen as the main medium for implementation due to the availability of a wide array of machine learning libraries. The Ubuntu system in backend was chosen in conjunction with Python 3 because of its developer-friendly environment.

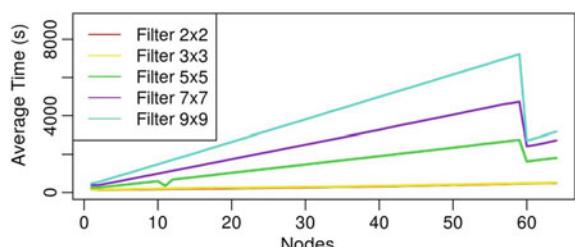
6.2 Experiment Outcome

The very first hyperparameter to consider was the number of layers that would be required. The original LeNet-5 talked about seven layers. But here, we considered five layers for testing. The reason for five layers was simple: reduction in the amount of calculations. If the model could fare well in five layers, then usage of seven layers would be redundant. Simultaneously, the nodes and filter size per layer were varied. The nodes were varied from 16 to 256 in multiples of 2 (i.e., 16, 32, 64, etc). The filter sizes that were chosen to experiment upon were {2, 3, 5}. Though the {7, 9} could have also incorporated, still it was not done because of few reasons. The first one being the amount of time taken for processing at each layer with such filter size was relatively large, as compared to that of {2, 3, 5} as shown in Fig. 5.

Hence, they were dropped right after the end of the five-layered model's test case. And the second reason is the exponential increase in the amount of the parameters to test upon, just by the mere presence of these two extra filters.

Now, remained 3 filters {2, 3, 5} and 5 nodes {16, 32, 64, 128, 256}. Given these 2 sets of parameters, one can easily say that for a single layer there will be 15 possible combinations. Hence, every single increment in the amount of layer will lead to exponential increase in the no. of parameters to train with, therefore for first

Fig. 5 Comparison of the filter sizes



layer: 15 parameters, second layer: 225 parameters, third layer: 3375 parameters and so on. So iteratively, each set of parameters would be picked up and trained upon. Also, since this procedure's aim was to obtain the single best set of hyperparameters, for each hyperparameter set, the corresponding model would be trained just for one epoch. That single epoch's accuracy, loss and time consumption would be later used to select the most appropriate parameter set.

A special note to be considered for further reading: The term 'layer' will now be used as a term to denote a set of convolutional, max pooling layers put one after the other. So for instance, Layer 2 will denote that there are actually four layers as follows: 1 convolutional, followed by a max pooling which is again followed by a consecutive convolutional and max pooling layers, respectively.

7 Results

Since the training was exhaustive in nature, the amount of results was exhaustive too. Hence, to get the best of all, the top five results were picked up based on the accuracy and loss for each layer and were presented over here. Note that Layer 2 and Layer 3 mentioned over here are basically the second and third set of convolutional layers with the max pooling layer, rather than the literal second or third layer. Thus, here every layer mentioned is basically a combination of convolutional layer and max pooling layer, where the max pooling filter size has been set to 2×2 in order to avoid loss of critical data.

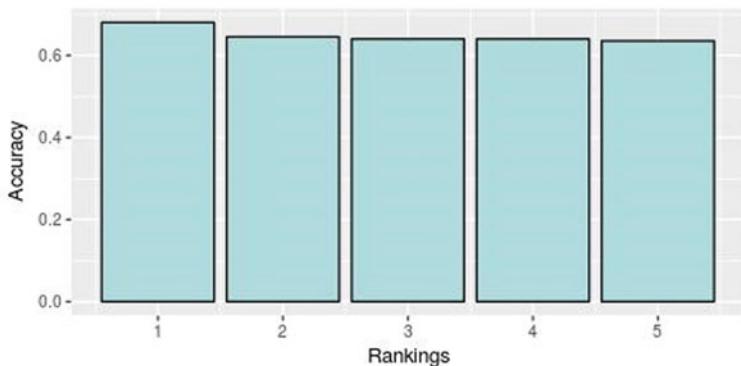
7.1 Metrics for Evaluation

The following metrics were used to judge the performance of the system.

1. Accuracy: The fraction of correct classifications that the model can predict.
2. Loss: The value obtained by applying a function over the difference of the predicted and actual value. And in this case, specifically the function is cross-entropy.

7.2 Layer 2 Results

Based on the maximum accuracy hit by each parameter set, in Fig. 6 graph is plotted. The plot is arranged in descending manner. Hence, the first bar shows the maximum accuracy. And the table beneath shows the parameter values taken up by the respective CNN layers. It is very much clear from the graph that 64 nodes and 3×3 filter for the



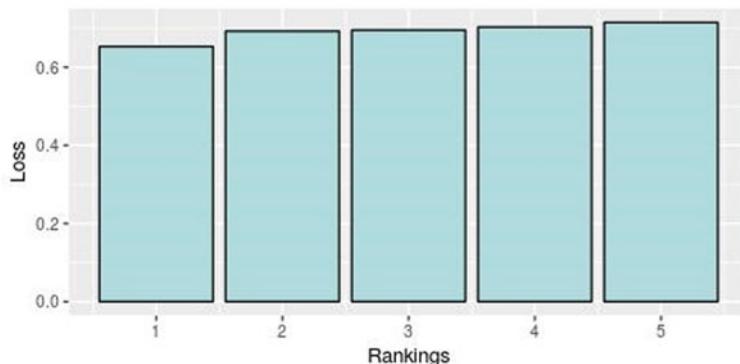
	Layer_2_Nodes	Layer_1_Nodes	Layer_2_Filter	Layer_1_Filter
1	64	16	3	5
2	128	64	5	2
3	64	32	2	2
4	128	64	5	3
5	32	128	5	5

Fig. 6 Results generated for two-layered configuration focused on accuracy

second layer in combination with 16 nodes and 5×5 filter for the first layer provide the highest accuracy possible for the two-layered CNN.

In Fig. 7, graph shows the loss values accumulated by the parameters while training. However, here the plot is arranged in ascending manner in order to show the minimum loss configuration first. The same will be reflected in the rankings area. Note that the configuration with the minimum loss is same as the one in the one with the maximum accuracy.

In order to be sure about the configuration, the accuracy-to-loss ratio was calculated corresponding to each configuration. Also, this would be a rather better metric, since it allows the accuracy to hit the max and at the same time expect the loss to hit its minimum too. Again, the represented results were found to be similar to the accuracy and loss as shown in Fig. 8. Hence, it is clear to assume that the configuration of 16, 5×5 for first and 64, 3×3 for the second is the best for two-layered CNN configurations.



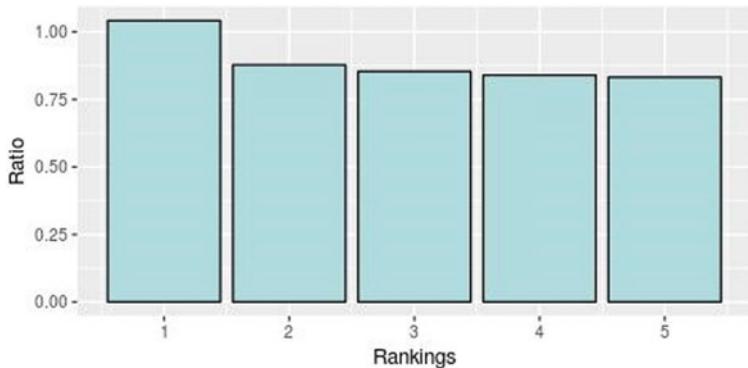
	Layer_2_Nodes	Layer_1_Nodes	Layer_2_Filter	Layer_1_Filter
1	64	16	3	5
2	128	16	2	3
3	128	16	2	5
4	64	16	5	5
5	32	16	3	3

Fig. 7 Results generated for two-layered configuration focused on loss

7.3 Layer 3 Results

In Figs. 9 and 10, graph shows the loss plot for the top five configurations which have minimum loss within the third layer. The graph has been arranged in the ascending fashion. Thus, the first bar shows the configuration with minimum loss. Note that the configuration achieved here is much different as compared to the one obtained in the accuracy plot above. To resolve this discrepancy and obtain the optimal configuration, we relied on the accuracy-to-loss plot. Hence, the final configuration will be obtained from the ratio plot.

In Fig. 11, ratio plot has been arranged in the descending fashion, so as to obtain the configuration with maximum accuracy combined with minimum loss possible. To be precise, the graph shows the top five configurations based on the accuracy/loss ratio. Hence, best configuration was selected from this plot itself, i.e., 32, 3 × 3 for first layer, 16, 3 × 3 for second layer and 64, 3 × 3 for the third layer.



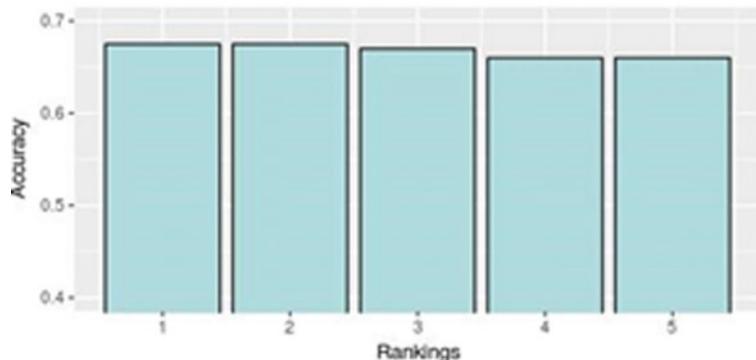
	Layer_2_Nodes	Layer_1_Nodes	Layer_2_Filter	Layer_1_Filter
1	64	16	3	5
2	128	16	2	5
3	64	16	5	5
4	32	16	3	3
5	16	16	5	2

Fig. 8 Results generated for two-layered configuration focused on accuracy-to-loss ratio

7.4 Comparison of Results

After the exhaustive test rounds, the results for both the second and third layers are in hand, as depicted in Fig. 12. Both of them have their own best possible configurations, which have been finalized based on the accuracy/loss ratio plot. Thus, this section will now pit the results of the top five configurations of both second and third layers in order to decide which will be the best one among them. The final metric remains the same, i.e., the accuracy/loss ratio.

The result of the above graph makes it pretty evident that on comparison in between the first ranks of best configuration for Layer 3 and Layer 2 on the basis of accuracy-to-loss ratio, Layer 2 is the better one. Though in the rest of the ranks (2–5) Layer 3 takes the lead, since the concern was related to maximum accuracy with lowest loss, Layer 2 was picked up as the final winner among all.



	Layer_3_Nodes	Layer_2_Nodes	Layer_1_Nodes	Layer_3_Filter	Layer_2_Filter	Layer_1_Filter
1	16	16	256	2	2	3
2	32	32	256	2	2	2
3	32	128	128	2	5	5
4	64	16	32	3	3	3
5	64	64	64	2	2	5

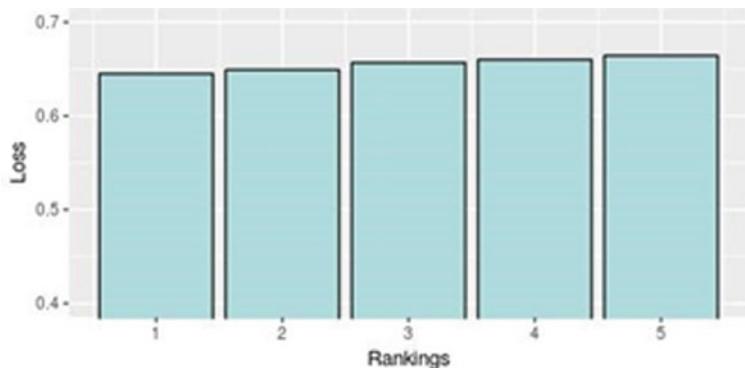
Fig. 9 Results generated for three-layered configuration focused on accuracy

8 Concurrent Research

Our research work derives inspiration from the work done by Andrew Vanderburg and Christopher Shallue in finding exoplanets from the Kepler mission data. Unlike the research done by Vanderburg and Shallue, we cut short on many preprocessing techniques and feed a relatively crude dataset for a system. Nonetheless, our model has considerable success in detecting exoplanet signals in smaller planetary systems. The model developed by Vanderburg and Shallue performs much better in discerning transits for multiplanetary systems. Our research work is currently still under development and in the process of incorporating a more variegated form of data. The BLS algorithm for transit detection is also a very successful method for detecting transits; nevertheless, we also planned to incorporate the BLS algorithm in our preprocessing phase.

9 Future Scope

This paper has focused on the design aspect of the convolutional neural network. The future work will involve actual prediction based on this model. The prediction will be



	Layer_3_Nodes	Layer_2_Nodes	Layer_1_Nodes	Layer_3_Filter	Layer_2_Filter	Layer_1_Filter
1	32	256	16	3	3	2
2	64	32	16	5	3	2
3	16	128	64	3	3	5
4	16	32	32	3	3	5
5	128	16	16	2	3	3

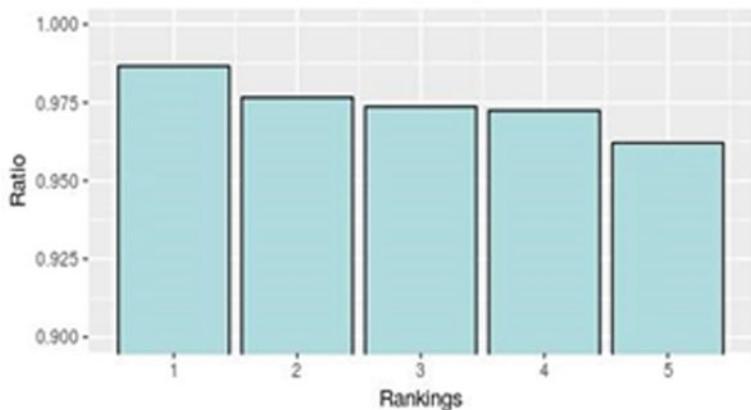
Fig. 10 Results generated for three-layered configuration focused on loss

performed for multiple star systems at once for searching possible exoplanets within the star system. The obtained results will be then compared with the existing models. The Exo-Drive model will be fine-tuned, respectively, to increase the accuracy of the designed CNN model.

10 Conclusion

Through this work, an effort was made to find a simple and effective neural network model which could be used to train the system in order to recognize graphs, and especially the light curves from the Kepler mission.

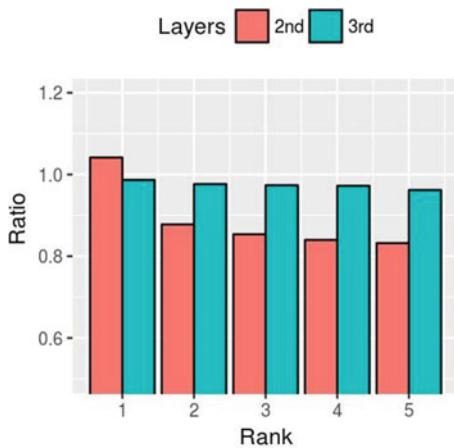
The entire research was successfully able to conclude that convolutional neural networks are the best option among all the neural network designs that would help us in working with images. Its ability to focus on the principal component pixels, rather than every minute detail, makes it a reliable choice. Now, even among the CNN world itself, there are a plethora of architectures. In order to keep the entire model as simple as possible, the LeNet-5 architecture was used as an inspiration and thus it was modified to the research needs as per the parameter configuration and the number of layers that would bring out the best results. With the help of the exhaustive



	Layer_3_Nodes	Layer_2_Nodes	Layer_1_Nodes	Layer_3_Filter	Layer_2_Filter	Layer_1_Filter
1	64	16	32	3	3	3
2	32	256	16	3	3	2
3	64	64	64	2	2	5
4	32	256	32	5	3	2
5	16	32	32	3	3	5

Fig. 11 Results generated for three-layered configuration focused on accuracy-to-loss ratio

Fig. 12 Comparison of the two- and three-layered configurations, based on their accuracy-to-loss ratio



training on all possible parameters (for nodes and filters), it was easy to claim that a two-layered CNN with $16,5 \times 5$ for first and $64,3 \times 3$ for second layer is the model which can provide us with the optimum results.

The paper and the corresponding research were aimed at analysis of neural networks, in order to train the system for graphs. In this case, the signals came in the form of images, which were later used to train the system by means of CNN. The CNN would learn by itself the pattern among all these images and give us a model. This resulting model would then help us to decide whether or not the light curve belongs to a star system with at least a single planet in their solar system. The convolutional neural network has the ability to easily extract features out of the images that appear as a pattern among the large training dataset and provide a model, capable enough to predict the result.

The main concern was to obtain a simple and effective model to train the system to recognize mainly the curves that appeared in the light curve scatterplots. The curves are result of the dip in the brightness mostly caused by a passing heavenly body in front of the star. The heavenly body by its very definition comprises planets, asteroids, etc., objects which are present in the outer space. Thus, the aim was to obtain a CNN model that would be the most suitable for the task of differentiating planets from other heavenly bodies from the light curves. Through a process of simple elimination, LeNet-5 architecture was used to obtain a rough outline for the CNN's architecture. The fine-tuning was then done with the help of exhaustive search for the parameters which would provide us with the best accuracy-to-loss ratio. The main varying parameters were the number of nodes and the filter size for each convolutional layer. Based on the accuracy-to-loss ratio, it was easy to come up with the top five configurations for Layer 2 and Layer 3 separately. Again, these separate results were combined to obtain the most optimum parameter configuration for the CNN. And Layer 2 was the one that was found out to be the most suitable among all to give the result that was hoped for.

Acknowledgements The work carried out in this paper is supported by Don Bosco Institute of Technology, Department of Computer Engineering. We also express our gratitude for granting us the resources required for the exhaustive search methods to obtain the required results.

References

1. Borucki WJ, Koch DG, Basri G et al (2011) Characteristics of planetary candidates observed by Kepler. II. Analysis of the first four months of data. *The Astrophys J* 736(1). ApJ, 728, 117—. 2011b
2. Koch DG, Borucki WJ, Basri G et al (2010) Kepler mission design, realized photometric performance, and early science
3. Catanzarite JH (2015) Autovetter planet candidate catalog for Q1-Q17 Data Release 24 (KSCI-19090-001), Tech. rep
4. Jenkins JM, Chandrasekaran H, McCauliff SD et al (2010) Overview of the Kepler science processing pipeline. *The Astrophys J Lett*

5. Coughlin JL, Mullally F, Thompson SE et al (2016) Planetary candidates observed by Kepler. VI. Planet Sample from Q1–Q16
6. Mislis D et al (2016) SIDRA: a blind algorithm for signal detection in photometric surveys. Astron Soc
7. Krizhevsky A, Sutskever I, Hinton GE (2012) In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems 25. Curran Associates, Inc.
8. Shallue CJ, Vanderburg A (2018) Identifying exoplanets with deep learning: a five-planet resonant chain around Kepler-80 and an eighth planet around Kepler-90. The Astron J
9. Dataset: Exoplanet hunting in deep space, Kepler labelled Time Series Data. <https://www.kaggle.com/keplersmachines/.kepler-labelled-time-series-data/version/3>. Last accessed 2019/06/17
10. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. <https://doi.org/10.1109/5.726791>. CiteSeerX 10.1.1.32.9552
11. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural network. In: 14th international conference on artificial intelligence and statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR:W&CP 15
12. Blog Post: Gabriel Garza, Exoplanet Hunting with Machine Learning and Kepler Data -> Recall 100%. Last accessed 2019/07/31

Chapter 58

Blockchain-Powered Real Estate System



Aman Jain, Bhumi Chitroda, Aditya Dixit and Harshal Dalvi

1 Introduction

Real estate is a limited liquid asset and is one of the significant choices of the investors as it is capital intensive. The real estate sector in India is growing at a rate of about 20% per annum, and this sector has been contributing to about 6–7% of India's GDP. The real estate industry has always been the primary choice of investors because it is an investment asset that increases in value over time and is less volatile than other investment assets, particularly equities. However, over time, the investors have faced the following problems, such as ownership rights cannot be easily transferred and the frauds related to property document forgery. These drawbacks may slow down the industry and its lush growth. The distortion created by large-scale corruption and inefficiency in India's land markets have shaved 1.3% off the country's GDP every year [1].

In the traditional approach, if the buyer and the seller have an agreement over the exchange of the property, then a sale agreement must be signed between the developer/seller and buyer. This sale agreement then makes the buyer as the legal owner of the property. This agreement along with the various documents such as Khata certificate, Receipt of property tax, Encumbrance certificate, Occupancy certificate, A power of attorney need to be registered with the Sub-Registrar office, and this

A. Jain (✉) · B. Chitroda · A. Dixit · H. Dalvi
Dwarkadas J. Sanghvi College of Engineering, Mumbai 400 056, India
e-mail: amanjain1483@gmail.com

B. Chitroda
e-mail: bhumi.chitroda75@gmail.com

A. Dixit
e-mail: adidixit98@gmail.com

H. Dalvi
e-mail: hddalvi.hd@gmail.com

process is carried out by a lawyer. All the property documents that require registration needs to be submitted to Sub-Registrar of Assurances-authority with which the sale deed needs to be registered within four months of the execution of the document. Documents required to be presented to the Sub-Registrar include property card with original documents and proof of payment of stamp duty. For the registration of the documents, Sub-Registrar will verify the stamp duty paid for the property. Stamp duty should be paid according to stamp duty ready reckoner. In case if there is any deficit in the stamp duty, the registrar will refuse to register the documents.

The whole process, which is described above, has made the transfer of ownership in the real estate system a tedious process that had many intermediaries that incurred the two-party cost. Also, the security in the system transactions has majorly suffered because of the bureaucracy and the irresponsibility of document handling and their forgery. The systems currently available are unable to remove the flaws that the industry faces. Thus, the drawbacks should be eminently removed using the technology that gives revolutionized outcomes. One such technology that could help eradicate the following problems is blockchain technology.

Blockchain is a growing list of records, linked using cryptography, also known as blocks. Each block in blockchain consists of a timestamp, transaction data, and previous block's cryptographic hash [2]. Blockchain can annihilate the problem of forgery as no other party can claim the property if the transactions they are performing are not logged into the system.

A smart contract is a rule-based computerized protocol which digitally facilitates, negotiates, verify or enforce performance of the contract. A smart contract does not require third-party intervention. Also, the transactions are irreversible and trackable. When compared with traditional contract law, smart contract aims to provide robust security along with reduced transaction cost [3]. The the technology effectively eliminates problems such as insecure transaction, forgery, lesser transactional fees and non-repudiation.

2 Existing Work

There are various proposed solutions/papers which offer a blockchain-based solution for the real estate applications. These solutions enable buyers and sellers commercial-property trades with asset security and smooth transactions. The blockchain technology solutions enable permanent, immutable storage of property documents, and transactions.

Nathan Shedroff proposed blockchain is open source, distributed, and shared peer-to-peer ledger of transactions. The transaction is added to the ledger when more than one node verifies/sees that transaction is occurring only then the transaction is considered as authentic. Once a verified record is added in the blockchain, it cannot be changed and available to the system user whenever required creating a new kind of transparency. Smart contacts are self-governed software codes which practically manages blockchain technologies. Blockchain eases the process of managing

the documents centrally rather than spending time and money behind real estate agent or current owner. By only submitting a query, user can get complete details of house including the past owners and tax assessment. Any blockchain-powered home will collect and record all the information of the house through blockchain-enabled services [4].

Alex Norta et al. proposed a way in which commercial-property-based trades can be done using distributed applications(Dapps). A blockchain serves as a storage space which provides an immutable, event-recording ledger. The solution proposes faster transaction times, cost reductions, and greater transparency in dealing which trades. The commercial-property trading introduced is based on the P2P (peer-to-peer) value-chain platform that avoids dis-intermediated intermediaries for enabling P2P trades. The system token is intended to be further used in Internet of Things (IoT) based on the hotel management system. The system also provides an auction pool that helps to trade in a decentralized manner providing a reliable and transparent environment for transaction [5].

QiuHong Zheng et al. proposed a way in which storage of documents and other data is possible with the limited space provided in a traditional blockchain. The issues with ample data storage and synchronization can be solved using an IPFS-based blockchain data storage model to solve this problem. The main idea is to store the data in the IPFS network; the network returns a hash for the provided data; this hash is then stored in the blockchain block. The validity of the blockchain can be directly verified from the local blockchain data. The IPFS network is supported by almost all types of blockchain that can store the hash result provided by the IPFS network. The disadvantage of storing in this network limits the retention of the historical traceability of blockchain. This storage model depends on the number of nodes in the network for providing storage and usually has low incentives [6].

2.1 *Existing Systems*

The following are some of the existing systems that have been developed or are developing blockchain-based solutions for the real estate industry.

2.1.1 **Rentberry**

Rentberry integrates blockchain technology to provide long-term rental platform in major cities of America. The platform provided by Rentberry not only cuts the operational costs but also provide renting auctions and crowdfunding for preventing the security deposits from being held by landlords providing full transparency that prevents bidding wars [7]. Rentberry is already successful in the process where tenants participate in the auctions for the properties they are interested in renting. In the auction, the buyer will have to pay 1000 BERRY tokens for participating in a property auction, and if the bid is accepted, 950 BERRY token covers the part of the

rent and remaining 50 BERRY token as a fee. In a case where the bid is not selected, the entire 1000 BERRY tokens are refunded. BERRY token is payment method on Rentberry where landlords collect rent via BERRY tokens [8].

2.1.2 Atlant

The Atlant system provides two major services, P2P rentals and tokenized ownership. The P2P rentals are the most direct and safest way in which blockchain helps tenants and landlords to connect with each other directly, pay directly, and solve disputes directly in an efficient manner. The tokenized ownership makes Atlant an interesting project. The system once legally verifies real estate property, the value of the property can be tokenized even in area measurements. In a case where the property is owned by multiple stakeholders, tokenized ownership can make it easy to transact and sell an illiquid asset causing a lot of investment potential. This technology helps the user to invest in small amounts on a particular land, thereby increasing the potential investment in the market and help users to perform various transactions, earning them profits through trade [9].

2.1.3 Bee Token

A Bee Token's system is a system that adapts the economic model of platforms like Uber and Airbnb, and implements the model with a distributed network technology and uses it for home rentals in a P2P manner. The system thus eliminates the problems that the users of Uber face, such as a hefty cut on the services provided using smart contracts, thus eliminating the intermediaries cost. The system strives to create a sharing economy. The platform where this sharing economy will be executed, the system BeeNest makes this possible with the help of a decentralized economic rental platform that helps its users to earn without worrying about the hefty cut of the intermediaries.

Bee protocols provide following features which include maintaining reputation, dispute resolution and quick payment [10].

2.1.4 Ubitquity

Ubitquity is the blockchain platform for real estate record keeping. The system provides software as a Service (SaaS) platform that provides individuals, municipalities, and businesses with secure storage and easily trackable properties. All these properties are provided through the blockchain ledger, which is encrypted, transparent, and incorruptible. Currently, the system is designed for the B2B market, aiming to help them speed up their real estate transactions. The system also has a blockchain-based notary service called BitNotarize that allows individuals to store and sign notary documents [11].

2.2 *Observation Related to Existing Systems*

Various use cases and approaches from the above projects can be taken in order to solve the existing problems and develop a blockchain-based solution for the real estate industry in India. The renting solution from rentberry can be applied by limiting the deposit used for buying the property, which directly helps in reducing the cost structure for buying and renting the properties. The tokenization based on the land area (1 token per land area concept) can be used for cryptocurrency dealings in the future. Applications can be referred for smart contracts and peer-to-peer transaction technology.

The existing systems do not calculate costs based on the area of the real estate, in India, the prices and the tax may vary from area to area. The storing and new document formats need to be made based on the area of dealing. The costs of the tokens used in the above systems should be lesser than the traditional system. The legal aspects of the systems need to be suited to the Indian legislature.

3 Proposed Architecture

The following architecture allows the buyer and seller to connect and query the blockchain using a Web interface. The blockchain allows the storage of documents in a distributed ledger, the distributed ledger serves as a consensus of replicated, shared, and synchronized digital data geographically spread across multiple sites, countries, or institutions. The architecture allows the users to login, make an offer, query the blockchain, receive an offer, agree on smart contracts, logout.

3.1 *Our Approach*

Initially, the details of the property, such as ownership and certificates regarding the property's previous details would be recorded on the blockchain. Storing the information on the blockchain, which serves as immutable storage, removes the drawback of document forgery or document mishandling which is prominently seen in the traditional approach. A website, enabling the user to search the property by using the property id assigned to it. The user can now query the blockchain to find details about the property (previous owners, debt, legal information). The information obtained need not be verified by the lawyer or a third party since the blockchain structure implies the inability to make adjustments to the data after they are recorded in the database. Since there is no need to verify the documents, the overall time and cost are lesser than the traditional approach.

On the website, the user can directly interact with the property owner for buying, renting, or leasing the property. All the interactions between the buyer and the seller

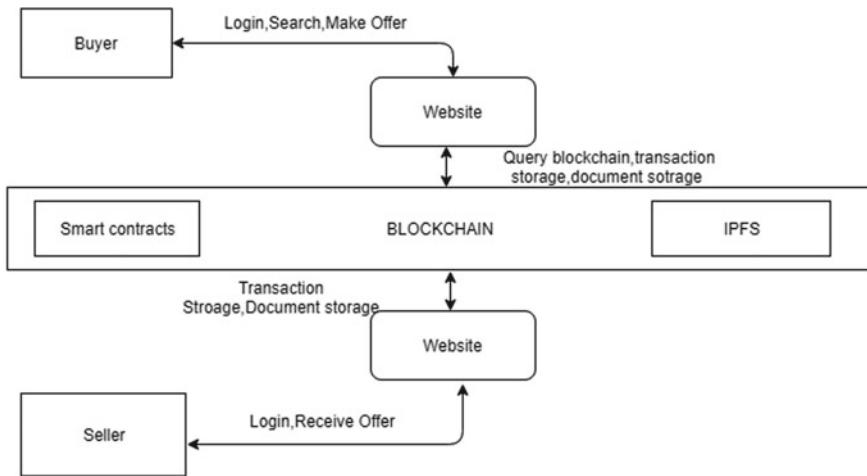


Fig. 1 System architecture

would be stored on the blockchain to prevent repudiation. Property transactions like purchase, sale, renting, leasing, and management transactions would be done through smart contracts. The use of smart contracts has various advantages over traditional agreements (sale agreement). Smart contracts would provide accuracy in recording the terms and conditions; inaccurate data entry would result in transaction errors. The agreement is fully transparent since the terms and conditions of the contract are fully visible and accessible to the relevant parties. The cost involved in using smart contracts and transactions would be significantly lower than the physical agreements used in the traditional approach. The cost associated with a smart contract transaction depends on the size of the transaction and could be up to 1.2 dollars for the proposed system.

Separate document formats would be provided to buyers and sellers based on the location of the property and the type of transaction to be performed. The storage of documents would be done using the IPFS network, and the hash mapping of the document would be stored on the blockchain. All transactions would be stored on the blockchain and can be queried easily to provide complete transparency (Fig. 1).

3.2 Roles of Various Components

3.2.1 Buyer/Seller

The buyer or seller can access the website to search for various properties available for sale, rent, or lease. The user can also interact with the buyer or seller through the various options available on the website.

3.2.2 Website

A website (based on PHP and SQL) would be available through which the user can log in, search, view, and have detailed information regarding the property. The website queries the Ethereum blockchain to get the relevant data based on user demand. It also enables the interaction between users through the use of smart contracts. The website fetches and displays the documents or details stored in the IPFS network.

3.2.3 Ethereum

Ethereum is a global, open-source platform for decentralized applications. Ethereum blockchain taken as a whole can be viewed as a public distributed ledger of transactions. Ethereum provides a platform that gives users to run distributed applications in a decentralized manner. Ethereum provides two types of accounts, externally owned account and contract account. The externally owned account has been automatically created on installation. A contract account can be used to set up policies regarding the account. The proposed system uses the Ethereum blockchain to store the transaction hashes and its ability to create smart contracts for secure peer-to-peer transactions [12]. Transactions cost on the Ethereum network varies with the size of the transaction, Gas consumed, and the Gas Price. Gas is a unit of measurement that calculates the amount of computational effort that it will take to execute certain operations. The Gas consumed varies with the size of the transaction. The user can select the Gas price based on the speed at which they want to send their transaction. The transactions such as storing the data of a new property would cost around 400,000 Gas. Based on the Gas Price selected, the transaction could cost from 0.08 dollars up to 1.2 dollars. The speed of the transactions varies from 2 mins (costliest) to 30 mins (cheapest).

3.2.4 Smart Contract

A smart contract is a computerized form of transaction protocol that carries out terms of contracts written for the blockchain. A smart contract is a collection of code (containing the functions) and data (containing the state). The smart contracts are stored and executed in blockchain nodes.

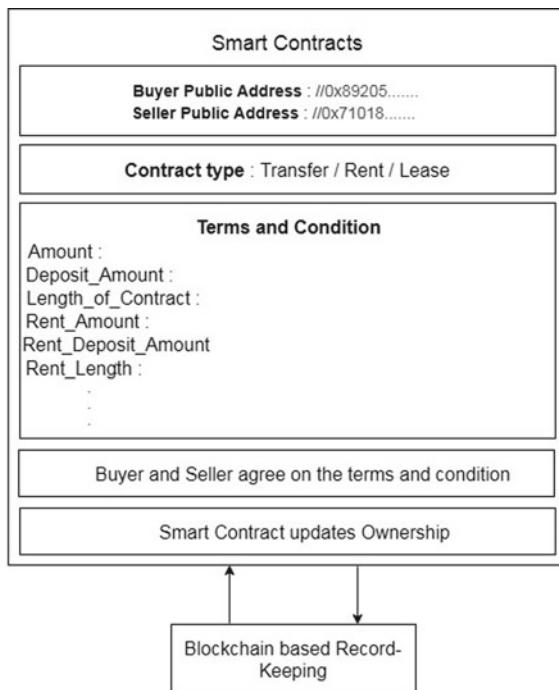
Smart contracts in the proposed systems allow the buyer and seller to agree on terms and conditions based on the transfer of property. The terms and conditions could be land price, deposit, rent amount, rent deposit, rent duration, lease amount, lease duration, etc. Once the price is paid to the seller, the smart contract automatically transfers the ownership to the buyer.

Smart contracts are self-executing contracts, which allows them to execute functions or transactions based on the terms and conditions. For example, if the length of the lease agreement is 12 months, then after 12 months, if the contract is not renewed, the smart contract automatically transfers the ownership back to the original owner [13].

3.2.5 IPFS

IPFS network allows an efficient way of storing document data. IPFS network allows storing property-related documents such as certificates and receipts. Storing the documents in the IPFS network reduces the size of the blockchain transaction and thus reduces the cost of the transaction. Using the IPFS network, bloating the blockchain with property document data is avoided. The miners in the IPFS network can directly store the data; the network then returns a hash that can be stored in the blockchain. The hash provided by the network is unique and can be verified to see if the data stored is different (Fig. 2).

Fig. 2 Smart contract



3.3 Features of the Proposed System

3.3.1 Excellent Accessibility

Real estate industry has become a power of capital appreciation and unexpected income through its operation over a period of time. The system provides excellent accessibility to their customers, investors to browse land, property, and capitalize in stocks of real estate using cryptocurrency.

3.3.2 Defeat Fraud and Hacking

The blockchain technology helps the real estate industry to prevent businesses against cybercrime. The system records the occurrences of hackers when they try to access data and logs every transaction (authentic or illegal) within the system. This system eliminates real estate fraud by allowing reliable blocks to store digital control records for resources.

3.3.3 Removing Middlemen Away

Most businesses require the use of intermediaries in some or the other way. The focus of intermediaries usually costs more to both the dealer and the customer. The technology of this system supports a shared database where anyone is capable of recording data, without any need of controlling it, and without any requirement of the support from middlemen.

3.3.4 Efficient Search Option

Real estate brokers, property owners, buyers, and sellers often search different properties based on the area of location, rental cost, the capital value of the property, and facilities within the property. With the help of blockchain application, development system provides accurate search option that can provide details on property location, directions, rental rates, lease period, property value, purchase history and previous owners, current owner details, and powerful evidence to prove that property is authentic.

4 Result Analysis

The following section compares the traditional system with the proposed system on the basis of some important parameters listed below. These parameters play a critical role in determining the efficiency of any real estate system and its impact on the users of the system (Table 1).

Table 1 Traditional system versus proposed system

Sr. No.	Parameter	Traditional system	Proposed system
1	Storage model	Physical document storage and digital records	Blockchain-based storage
2	Intermediaries	Significant role of lawyers and brokers	No intermediaries needed
3	Duration	8–12 weeks	Upto two days
4	Cost	30,000 rupees upto 1% of property value plus additional intermediaries fees	35–200 rupees
5	Security aspect	Tampering, mishandling of documents and single point of failure	Tamper-proof and decentralized

4.1 Storage Model

The traditional approach uses physical document storage and digital records for storage of land records. The physical document storage models usually suffer from document mishandling, but this issue is solved using digital records. Digital records are maintained on a Web-enabled central database. But, the centralized storage has a single point of control and a single point of failure. If data is lost or hacked in a centralized system, retrieving it would be impossible.

Blockchain storage models offer a decentralized and distributed approach for storing document data. The blockchain data is usually stored on thousands of devices on a distributed network of nodes, the system and the data are highly resistant to technical failures and malicious attacks. Each network node stores a copy of the database, and this avoids the issue of a single point of failure.

4.2 Intermediaries

In the traditional approach, the role of intermediaries is significant. Brokers are usually trusted to negotiate and arrange real estate transactions, writing contracts, and overseeing transactions for sales and purchasing activities. Lawyers are used to offer legal advice, verify documents, and assist with the transfer of properties.

In proposed system, the role of the broker is eliminated as the system allows the users to interact directly with each other for negotiation and arrangement of real estate transactions. Smart contracts are used to ensure that the transactions are error-free. The blockchain structure implies the inability to make adjustments to the data after they are recorded in the database, this means that the documents on the blockchain do not need verification by a lawyer.

4.3 Duration

The transfer of ownership in the traditional approach involves a sale agreement that must be signed between the developer/seller and buyer, along with the various documents that need to be verified (usually by a lawyer) and submitted to the Sub-Registrar. The whole process involving paying the registration fees and stamp duty could take up to 3 months.

In the proposed system, the transfer of ownership is carried out by the use of smart contracts. Smart contract transactions usually vary from 2 to 30 mins. Transfer of ownership is carried out by the smart contract based on the terms and conditions agreed on the contract. The documents stored in the system do not need to be verified, and the new documents are created based on the predefined formats available on the system. The use of smart contracts and the fact that the documents do not need to be verified reduces the duration of transfer significantly.

4.4 Cost

In the traditional system, the transfer of ownership includes the registration fees, stamp duty fees, document fees, lawyer fees, and broker fees. Registration fees could be from 30,000 rupees up to 1 percent of the property price, and stamp duty serves as a tax levied on any kind of transaction (related to property) that takes place and is documented. Stamp duty charges vary from 0.5 to 8%. Lawyer fees and broker fees are based on the individual lawyer or broker.

In the proposed system, the registration fees, document fees, lawyer fees, and broker fees are eliminated as the system does not require the use of intermediaries. Stamp duty fees may be still applicable based on legal proceedings. Transaction fees for the ethereum blockchains vary from 30 to 200 rupees.

4.5 Security Aspect

The use of physical document storage and centralized digital records can lead to various security issues. Physical documents can be misplaced or tampered. There is no way to detect the tampering of such data. In centralized databases, the security of the database is a concern. The centralized databases if insecure could lead to alteration of data. Central databases also suffer from a single point of failure; data retrieving would be difficult in case of data loss or hack.

The use of blockchain provides immutable storage that prevents data from being tampered. Blockchain technologies are highly resilient to hacking and other forms of external attacks. The same data is stored on all the nodes on a blockchain, and there is minimal to no risk of data loss.

5 Conclusion

The proposed system describes a distributed approach for the real estate industry in India. The system would provide a more secure and easy to use approach for the real estate applications such as buying, selling, renting, and ownership. The system uses blockchain as a storage medium which provides a decentralized, immutable, and secure way for property trading and ownership. The middlemen in the traditional approach are removed in the proposed system, which enables peer-to-peer transaction. The whole process of ownership transfer and exchange would be done in a time-efficient manner. Although the blockchain technology is relatively new, the technology could potentially provide a better and efficient approach for the real estate industry.

References

1. Blockchain is helping build a new Indian city, but it's no cure for corruption. <https://qz.com/india/1325423/indiast-andhra-state-is-using-blockchain-to-build-capital-amaravati/>. Last accessed 5 Nov 2019
2. What on earth is blockchain technology? <https://towardsblockchain.com/blogs/what-on-earth-is-blockchain-technology>. Last accessed 1 Dec 2019
3. The Beginner's guide to smart contract testing thong Nguyen-Thuc Nguyen. <https://www.logigear.com/magazine/test-automation/the-beginners-guide-to-blockchain-and-ethereum-smart-contract-testing>. Last accessed 1 Dec 2019
4. Shedroff N (2018) Self-managing real estate. In: Proceeding of IEEE computer 2018 conference, pp 104–104
5. Norta A, Fernandez C, Hickmott S (2018) Commercial property tokenizing with smart contracts. In: Proceeding of international joint conference on neural networks (IJCNN), pp 1–8
6. Zheng Q, Li Y, Chen P, Dong X (2018) An innovative IPFS-based storage model for blockchain. In: Proceeding of IEEE international conference on web intelligence (WI). Santiago, pp 704–708
7. Long-term Apartment Rentals on Rentberry. <https://rentberry.com/>. Last accessed 1 Dec 2019
8. What is rentberry? Coin Central. <https://coincentral.com/rentberry-ico-analysis>. Last accessed 1 Dec 2019
9. ATLANT world's real estate blockchain platform. <https://atlant.io>. Last accessed 1 Dec 2019
10. Bee Token (BEE) price, charts, market cap, and other metrics. <https://coinmarketcap.com/currencies/bee-token>. Last accessed 1 Dec 2019
11. The enterprise ready blockchain-secured platform for real estate recordkeeping: one block at a time. <https://www.ubitquity.io>. Last accessed 2 Dec 2019
12. Anoica A, Levard H (2018) Quantitative description of internal activity on the ethereum public blockchain. In: Proceeding of 9th IFIP international conference on new technologies, mobility and security (NTMS). Paris, pp 1–5
13. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang F (2019) Blockchain-enabled smart contracts: architecture, applications, and future trends. In: Proceeding of IEEE transactions on systems, man, and cybernetics, pp 2266–2277

Chapter 59

Optimizing Reverse Image Search by Generating and Assigning Suitable Captions to Images



Dhvani Kansara, Aditya Shinde, Yashi Suba and Abhijit Joshi

1 Introduction

Reverse image search is a query technique that formulates a search query based on the sample image provided to it. It works on the principle of content-based image retrieval (CBIR) which allows users to obtain results related to the content of the sample image provided by the user. It also helps to find out other derivative or manipulated versions of the sample image. Thus, reverse image search helps to find images that most closely resemble the input image.

The human mind remembers more of what it sees than it reads. Images help us visualize the information we are looking for. Thus, it is widely crowd-pleasing, and a large amount of population depends on image search for various activities which was concluded by the survey conducted by the authors. It showed that about 80% of people use reverse image search frequently for different purposes. Out of which about 70% of people did not receive expected results. The reason is that the current system generates its output based on a single keyword search, color distribution or meta tags relating to the image. Such superficial information is not able to provide accurate results because it does not take into consideration the content of the image. This motivated us to improve and optimize the search engine. Visual knowledge fed to the computer through images can be interpreted with the help of descriptions or captions. Captions are the detailed description of images that help to distinctly locate an image from its pool. These descriptions are composed of objects represented in

D. Kansara · A. Shinde · Y. Suba · A. Joshi (✉)

Dwarkadas J Sanghvi College of Engineering, Mumbai University, Mumbai, India

e-mail: abhijit.joshi@djsce.ac.in

D. Kansara

e-mail: dhvani.djk@gmail.com

A. Shinde

e-mail: adityashinde989@gmail.com

the image and the relationship among those objects. Using captioning, one can find results that are more accurate and apt to the user's search query. Captioning also allows us to filter out any irrelevant and disturbing results that are widely published on the Internet. Thus, developing the said model will help users to search images that are more in sync with their query and provide results efficiently.

Thus, the system presented in this paper will generate captions or descriptions for the images. This will help in optimizing the reverse image search technique by searching images based on the captions rather than the current criterion for search. This will provide optimized results because the interpretations are universal, and scope for confusion is minimized. This will increase user engagement, popularity and improved search engine optimization of the system.

2 Literature Review

In the literature review, various approaches for image captioning were discussed, and the drawbacks of the existing systems were identified. The details are given in the subsequent sections.

2.1 Existing System

Most of the systems for image captioning use the basic pipeline of image feature extraction followed by sentence generation. The captions generated by some of them are not grammatically correct. A few systems provide a complex but an accurate way for getting captions with correct captions and grammar.

The existing system for reverse image search labels the entire image by a single keyword. For example, for an image of a beach, the existing system will refer to it as "beach" and fetch results accordingly. This leads to inaccurate query results, because a broad range of images can refer to the word "beach."

Elamri and Planque [1] generated image captions using deep learning techniques. This system was developed for visually impaired people, so that they can better understand their surroundings. One of the objectives of the system is to automate caption generation of online images which will make the Web a more inviting place for visually impaired surfers. In this system, image features are extracted using convolutional neural network. Then, the dimensions of this image feature vector are reduced using principal component analysis (PCA). This resulting feature vector is then fed into the recurrent neural network model for sentence generation. The generated description of the image is in valid English. This system was capable to generate precise captions.

He et al. [2] used the method of Parts of Speech (POS) tags. This is used to guide the training and testing process. POS tags are used to generate relative context for the sentences that are developed. An important feature presented in this is the use of

a recursive method which manages a memory cell to keep a track of the complete sentence and thus uses the context from the entire sentence to append the next word while generating the caption. Also, POS tags ensure that the right part of speech appears at the right place ensuring grammatical soundness of captions generated.

Yang et al. [3] presented a novel method for caption generation using pretrained vectors. The new scope covered by them is in terms of diversity of images. It uses a model that is already trained by Oxford to learn input features in addition to the new features learnt from the user-specific dataset, thus making the model more exhaustive with respect to object identification. The concept of “attention mechanism” is used in this model. It presents a distinct approach, wherein attention is concentrated on features using probability distribution of objects based on their brightness and intensity in the overall image. For sentence generation, long short-term memory (LSTM) model is used (an improvement to RNN) which uses memory to remember context of words.

Zhou et al. [4] presented a model which showed better performances than other state-of-the-art models developed until 2016. The authors talk about the use of text conditional-based attention; it aims at training the model for “captioning” rather than mere feature extraction. The scope of this model is expanded to cover detailed and intelligent feature extraction where the models try to find objects related to the object detected previously. This theory covers methods such that the model becomes intelligent enough to recognize relations between objects of an image.

Asakawa et al. [5] trained the model in unsupervised environment by remembering long-term details in memory. In order to incorporate a complete reference, a more robust system is developed using autoencoders. The method of using autoencoders and paragraph vectors was found to have better context in this methodology. This approach has an encoder and a decoder, both work recursively by using words as input and giving sentences. In this case, the intermediate sentences are generally ungrammatical and do not transit smoothly from one to other. In later iterations, it takes text from neighboring sentences rather than only the current sentence. Later, formation of sentences was recommended using a generative model which is based on a regularized version of standard autoencoder. Once the features are translated into probability distributions, the process of extracting words for sentences becomes faster. Autoencoders yielded similar results as the standard recursive models but improved autoencoders can be increasingly used for more narrative descriptions and for generation of longer passages.

Chen et al. [6] replaced the CNN part with three state-of-the-art architectures (VGGNet, AlexNet, GoogleNet). It is found that VGGNet performs best according to the BLEU score. A simplified version of Gated Recurrent Unit (GRU) (a variant of LSTM) is used as a new recurrent network which is implemented using Caffe. Caffe provides a modifiable framework for the state-of-the-art deep learning algorithms. The simplified GRU achieves comparable result when it is compared with the LSTM method. But, it has few parameters which save memory and are faster in training.

In all works stated above, the performance is measured using BLEU metric. It measures the difference and deviation between captions generated by the computer and those given by humans.

2.2 Literature Related to Works

The information gained from reading all the existing work is that the captions are generated in two steps: first, feature extraction using object detection followed by sentence generation using language interpretation.

Feature extraction: This is the process of extracting features and objects from images uploaded by the user. The main approach followed by all the existing systems is, initially, the image is divided into various parts for identifying components in each section. This makes it easy for the model to focus on each part and retrieve maximum information for each section. This also ensures that any important information is not left out. Using the similar approach, Karpathy et al. [7] present a very accurate approach by detecting the minute details and even those which can be missed by any interpreter. Zhou et al. [4] used attention mechanism along with object detection through which the model tries to predict the object it has to focus. Khurana and Awasthi [8] used template matching for object detection. In this technique, templates of objects are stored and each input image object is compared with the stored templates to find out a match.

Sentence generation: In this process, captions are generated using language interpretation based on the features extracted in the previous step. In the existing systems, a recursive approach is used where sentences are generated based on Parts of Speech, i.e., using grammatical context and relevance to the previous words of sentences. A recursive method is used where each word of the sentence which is accepted by the model is followed by the next word, which is most suitable in context and grammar. Other approaches using autoencoders and probability distributions for generating sentences are also proposed by Asakawa et al. [5].

2.3 Observations Made on the Existing Systems

The major drawbacks in the above systems and the approaches are:

- They fail to generate accurate and contextual captions for moderate-to-high complex images.
- They sometimes overestimate the relationships between objects by forcefully trying to focus on things which are not present in the image.
- The models using attention mechanism generate captions which ignore the background details due to reduced intensity of such objects but provides a concise description including all the important objects.
- Autoencoders are not feasible for generating concise captions; they are more suited for producing large paragraphs rather than captions.

In the proposed work, we tried to overcome these issues.

3 The Proposed Approach

After a thorough literature survey, it was concluded that convolutional neural network (CNN) is the most suitable method for the purpose of feature extraction; thus, objects from the image will be extracted using CNN. In order to upgrade the object detection, a pretrained model by Oxford called as “VGGNet” is used. This will lead to improved accuracy as initial weights in the neural networks will not be random, rather will be based on a model which is already trained on many images. The system uses additional datasets on top of it, thus leading to improved accuracy.

In order to generate the sentences, long short-term memory (LSTM) is used. LSTM is a form of recurrent neural network that uses an internal memory to remember its inputs. Thus, it is appropriate for problems involving sequential data (same as in our problem). The internal working of LSTM uses three gates, the input, output and forget gate. The forget gate determines which information is important and which is not. Based upon the decision taken by the forget gate, the algorithm deletes or remembers certain information. The importance of information is assigned by the weights calculated during the training process. Thus, after receiving the input, the image is passed through the system in order to receive the most appropriate caption. Figure 1 shows the flow of information while generating the caption of an uploaded image.

The overall caption generation process is divided into two phases, namely the training phase and the execution phase.

In the training phase, the required data is acquired from various sources, and then, it is gathered at one place which is later used for the training purpose.

1. Acquisition and Gathering: Two datasets are used for training in order to obtain a good accuracy:

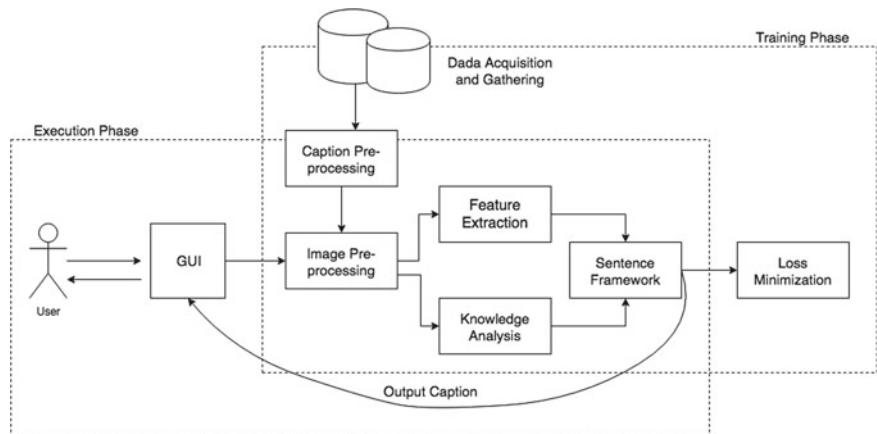


Fig. 1 System architecture

Flickr8k: Consisting of 8,000 images and 5 different descriptions/captions for each image provided by different people based on their perspectives.

Flickr30k: Similar to above with 30,000 images and respective captions. These images also contain celebrity images to detect renowned faces in an image.

Thus, the proposed model uses a total of 38,000 images for training. All captions are labeled by humans based on their perspective about the image. The majority of captions consist of succinct descriptions with an average length of 10 words. Because of the volunteer nature of the project, language errors are relatively common (in a random sampling of 100 captions, five had either a spelling mistake or a significant grammar mistake). These datasets contain a text file with the image names and corresponding 5 captions. Another database contains all the images. The captions are extracted by reading the text file linewise and creating a list of each caption along with corresponding image name. Images are read from the database and stored in a matrix form. It is necessary to preprocess the captions and images from the dataset so that the model training is easier and more efficient.

2. Preprocessing: After the creation of the required dataset of images and captions, captions are preprocessed by splitting each of the five captions for an image and inserting <START> and <END> tags at the beginning and end of each caption. The words in each of the captions are tokenized based on their context. In the process of tokenization, vector values are assigned to each word in a multidimensional free space. For example, the apple vector and fruit vector will be nearby, but apple vector and tiger vector will be mapped distantly in free space. Images are also preprocessed by converting them from RGB to BGR format, for faster processing during training. After preprocessing, the objects in the image are identified by feature extraction, which is the next stage.
3. Feature Extraction and Knowledge Analysis: Feature extraction algorithm, here convolution neural network (CNN), is applied on the image to extract the objects contained in it. First, it splits the image into segments and then extracts features/objects from each part. The convolution layer is applied on these segments and convolutes a kernel (feature vector) over the pixels of the image, which gives a feature map. In this way, the network first learns edges and curves (based on the feature vector) and then slowly understands complex and intricate shapes. Since we used a pretrained model—VGGNet, this process is faster. The output of the convolution layer is fed into the pooling layer which reduces the spatial dimensions by averaging or taking the maximum value of the subregions in the feature maps. This improves computation performance and also reduces chances of overfitting. The next layer is flatten layer, which converts 2D matrix into 1D vector. At the end, the output objects are extracted using an activation function. The objects that are extracted in this step are related using a sentence framework.
4. Sentence Framework: In order to determine the relationship between objects, the model is then trained to understand and interpret the captions in the training data; i.e., the model is taught to learn “English” with the use of Parts of Speech (POS) tags so that it can generate sentences. The algorithm chosen is such that internal memory is used to process the sequence of inputs, which is called long short-term

memory (LSTM). In this algorithm, the information cycles through a loop. The input to the neural network is the current input as well as the output from the previous inputs, based on their importance and significance. For example, if the object extractor has fetched “man,” “TV,” etc, then this step will associate them by “man” is watching “TV.” These sentences will be generated by the knowledge analysis performed in the previous step.

5. Thus, the system generates caption for the given input image. In the execution phase, this generated caption will be directly provided back to the user because the model is already trained. But during the training phase, in order to improve the accuracy, it is necessary to minimize the loss of information that might be produced in this step.
6. Loss Minimization: This step is performed only during the training phase to minimize the inaccuracy, while the model is still in the learning phase. Here, the model learns from the wrongly generated captions so as to update the weights in the neural network using backpropagation. For this, the error is calculated by comparing the caption generated by our model and the caption already present in the training set. The difference between these vectors is calculated to check the error, and weights of the neural network are updated to minimize this error. After the above steps are performed, the model is tested to check the accuracy of the generated captions with the test dataset.
7. Thus, the system is ready to take user input and generate appropriate captions by generating output based on the trained and tested model; this is called the execution phase. During the execution, the user image passes through the feature extraction and sentence generation framework to provide the output to the user.

4 Results and Analysis

Figure 2 shows the home page of our system. The user can upload/drag an image from their directory or can also upload the URL of the image. An API is also developed which can be integrated with a search engine to improve the search results.

Figure 3 represents a general output caption, “a group of young men playing a game of football” for the input image uploaded by the user. After performing black-box testing for multiple classes of images, it was found that the caption generator could label images in domains of sports, movies, environment, nature, animals, birds, etc., accurately.

Figure 4 shows that the model is able to recognize renowned monuments as well. For an image of the Taj Mahal uploaded by the user, the system caption generated was “a group of people standing in front of the Taj Mahal,” rather than calling it just another building. In addition to this, it is also able to identify renowned celebrities and other revered personalities as well.

Figure 4 shows that the model is able to recognize renowned monuments as well. For an image of the Taj Mahal uploaded by the user, the system caption generated

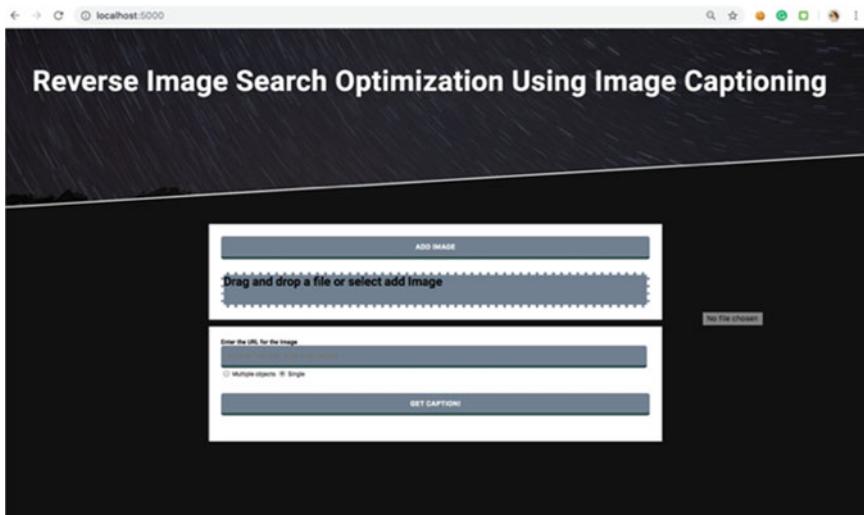


Fig. 2 GUI for client

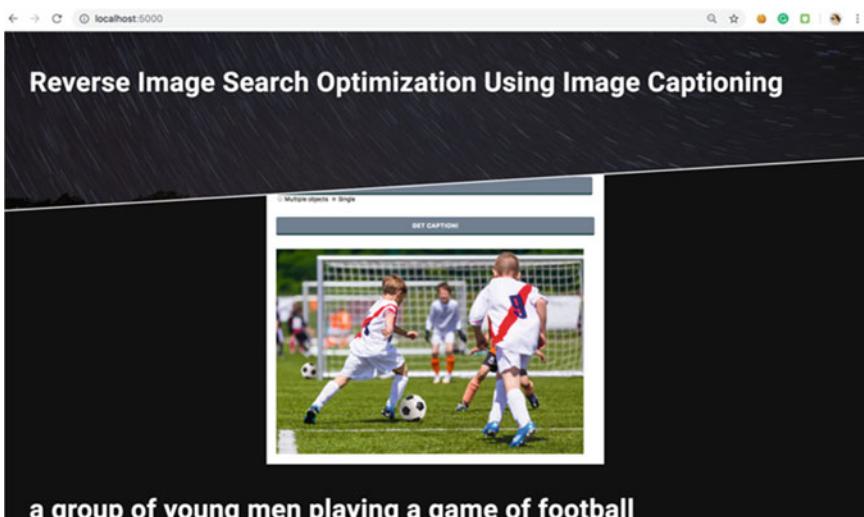


Fig. 3 Example output 1

was “a group of people standing in front of the Taj Mahal,” rather than calling it just another building. In addition to this, it is also able to identify renowned celebrities and other revered personalities as well.

When recognizing multiple objects in the image, the model gave output in a succinct form which could not list all the objects in the image rather provide a



Fig. 4 Example output 2

generalized caption, “a herd of cattle standing on top of a grass covered field” for the whole image as seen in Fig. 5.

In order to address this problem, the model contains a feature where the user can select a radio button of whether the user wants to display a succinct caption or a dense caption. A dense caption for Fig. 6 would look like one in Fig. 6. Such a caption will



Fig. 5 Example output 3

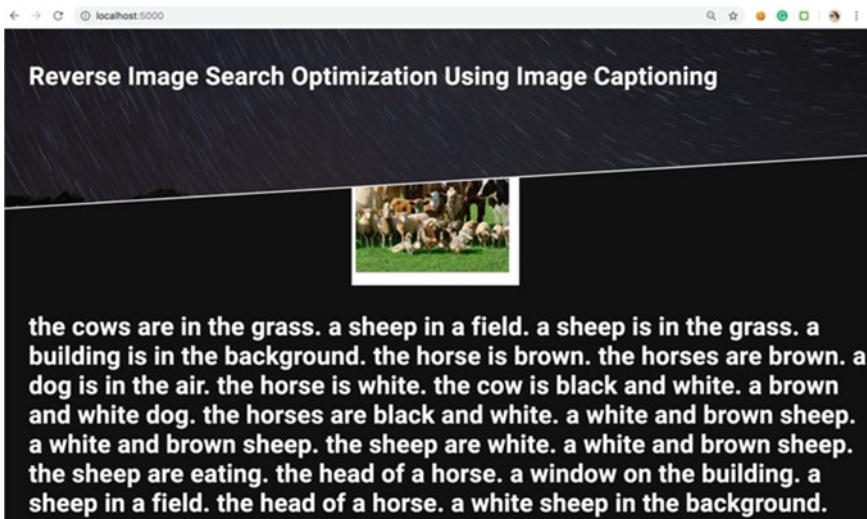


Fig. 6 Example output 4

enlist all the details included in the image. The mechanism used in Fig. 5 is called “attention mechanism” where focus is given on the image as a whole, whereas in Fig. 6, the focus is on each individual object present in the image.

5 Conclusion

In the proposed model of image captioning and thus reverse image optimization, images are taken as the input from the user and efforts are made to formulate most specific and relevant description to the image. This description will be in the form of captions and given as the output to the users. These descriptions are the universal to prevent any misinterpretation. The search engine is directly queried using these descriptions so that the user gets the resultant images which most closely resemble the input image. The proposed system will help the users to find the relationships between objects in the image and information about the image.

References

1. Elamri C, de Planque T (2016) Automated neural image caption generator for visually impaired people. Department of Computer Science, Stanford University. <https://cs224d.stanford.edu/reports/mcelamri.pdf>. Accessed 18 Oct 2018
2. He X, Shi B, Bai X, Xia G-S, Zhang Z, Dong W (2016) Image caption generation with part of speech guidance. In: IEEE conference on computer vision and pattern recognition 2016, school

- of electronic information and communications, Huazhong University
- 3. Yang Z, Zhang Y-J, ur Rehman S, Huang Y (2016) Image caption with object detection and localization. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. <https://arxiv.org/pdf/1706.02430.pdf>
 - 4. Zhou L, Xu C, Koch P, Corse JJ (2017) Image caption generation with text-conditional semantic attention. Robotics Institute, University of Michigan, Department of Computer Science, University of Rochester, Electrical and Computer Engineering, University of Michigan, 2017
 - 5. Asakawa S, Ogata T (2017) Comparison between variational autoencoder and encoder-decoder models for short conversation. Proc Int Conf Artif Life Robot 22:639–642. <https://doi.org/10.5954/ICAROB.2017.OS1-4>
 - 6. Chen J, Dong W, Li M (2017) Image caption generator based on deep neural networks. Department of Computer Science, University of British Columbia. https://www.cs.ubc.ca/~carenni/TEACHING/CPSC503-19/FINAL-PROJECTS-2016/image_caption_generator_final_report.pdf. Accessed 12 Nov 2018
 - 7. Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: IEEE conference on computer vision and pattern recognition
 - 8. Khurana K, Awasthi R (2013) Techniques for object recognition in images and multi-object detection. Int J Adv Res Comp Eng Technol (IJARCET) 2(4):1383

Chapter 60

Genre-Based Indian Viewer Movie Reviews—A Sentiment Analysis Classification of Text and Emoticons with a Supervised Machine Learning Approach



Ashish Modi and Elizabeth L. George

1 Introduction

Indian movie viewers are huge in number as Bollywood adds color and drama in its films. The Indian audience demands drama and color as it has been a favorite pastime before and for the new millennium as well. With globalization, we also saw the advent of swanky multiplexes and malls that completely changed entertainment for many Indians. With India now as a “connected country,” the entertainment industry is able to actively engage with the young and the old. The Indian film industry has now matured and it’s high-quality, content-driven cinema can compete with any marquee names and a major stars past record cannot guarantee box office success.

Young viewers today, having been exposed to international cinema, television, and arts from an early age want relevant, high-quality entertainment avenues to satisfy their evolved sensibilities. The earlier “standard formula” will not do anymore. Indian films (even the vernacular films and even a few English language films have now been modest commercial successes) now frequently have differentiate interpretations of the subject, which arises from a diversity of world views and from exposure to life-worlds different from their own.

A. Modi · E. L. George (✉)

Department of IT/CS, Nagindas Khandwala College, Malad-W, Mumbai, India
e-mail: e.leahgeorge@gmail.com

A. Modi
e-mail: ashishmodi536@gmail.com

Many research studies have taken on the movie genres and prediction of box office hits, but very little analysis has been done on the views of Indian cinema genres and the audience/viewers themselves. As per the many studies and reports conducted in this field, the genres are widespread, the viewer first selects vernacular/regional language movies and then movies in Hindi language. The success of Bollywood movies still mainly depends on the actors, storyline and the message the movie conveys.

It is a real challenge for the director and the production/technical team (for the film industry) to understand the viewer's style and the chances of the box office success of the movie.

BookMyShow.com is a business website, that is India's largest online movie and event ticketing platform, and has always been an early adopter and pioneer of *technology* for the film/mass audience field. This website has a major impact on the business of films as most viewers nowadays book their tickets online (barring rural and backward areas, where manual ticketing still rules). Also the movie selection process and the availability of tickets of that particular movie can be easily done with this service.

Opinion mining is a growing field that identifies the thoughts and sentiments of people, which they express in form of their feedbacks or reviews on various matters. Sentiment analysis (SA), often referred as opinion mining, is the extraction, identification or characterization of the sentiment from text using Natural Language Processing (NLP), statistics or machine learning (ML) methods [1].

In this paper, the main focus is on sentiment analysis that identifies the exact sentiment expressed in a text and then analyses it. Therefore, the target of SA is to find opinions, identify the sentiments they express, and then classify their polarity. Document-level SA aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document as a basic information unit (talking about one topic). Sentence-level SA aims to classify sentiment expressed in each sentence. Sentence-level SA will determine whether the sentence expresses positive or negative opinions.

The main aim of this paper is twofold: The first part is to present a Python script that chooses the comments from the webpage, loads them into an Excel file using grabber.py.

The second part of the paper is on sentiment analysis of comments. This classifies (with machine learning supervised approach) the viewer rating of the movie and which kind of a movie is liked by the target audience.

The remainder of this paper is organized as follows:

- Section 2 introduces related work and methodology used.
- Section 3 describes the associated dataset, experiments and evaluation.
- Section 4 presents the results, conclusion, and future work.

2 Related Work and Methodology Used

2.1 Literature Review

Sentence-based sentiment analysis is a much researched topic with many research papers focusing on the IMDB dataset and the opinion reviews predicting the sentiment polarity.

Similar research ideas are retrieved from Twitter dataset for movie reviews as in the reference paper [2] that analyzed different kinds of tweet classification algorithm and movie rating algorithm.

In [2] movie sentiment analysis with tweets, Python programming language is used with the NLTK library for a different types of sentiment classification based on the frequency of words that compares the results with machine learning techniques.

In the paper Movie Prism [3], the system automatically downloads movie reviews and creates an opinion summary for movie recommendation systems. The sentiment polarity is then computed for aspect identification that was measured through precision, recall, and *F*-measure.

We have used a similar system—but have created a script that captures data from BookMyShow for purely research purpose and analyses it as below. The movies for review were selected based on the classification of different movies, on the age group that can give a mixed background of data interpretation. The research paper MovReC [4] tailors a user's needs of choosing a movie by learning his interest based on the existing metadata, which include movie ratings, genres, personal ratings provided by either parents, guidance, or children, movie descriptions, movie reviews with providing children with movies that can enhance their educational, social, and emotional development.

In any recommendation system, collaborative filtering is used that takes the input of genres and users ratings to harness the expertise of both the movie experts as well as the common consensus of the crowd.

In the study conducted for the movie genre in making movies based on the viewer's likes, major types of movies were made on the following five types of genre such as **action, comedy, drama, horror, thriller, and romance**.

The movie release on popular demand as mentioned in [5] indicated that comedy and romance topped the list and thriller and drama are moderately popular among the children and adults. Also the movie rating improved or had higher frequency for family-based movie story line.

2.2 Methodology

This section emphasizes on the following phases. Creating a dataset of movie reviews from the popular movie booking website BookMyShow that takes an integrated framework of a movie name from the user, crawls the information about movie and its reviews which can be both textual and emoticon based from the website using the Python script Grabber.py, the comments classifier pre-processes the data by tokenizing the comments with stop words removal, moving to the next phase of detecting the sentiment polarity about that movie user comment. The system finally presents a summary of overall sentiment strength in terms of all major aspects identified (Fig. 1).

2.2.1 Dataset Collection

The overall idea of choosing a dataset that is freely available with the live user comments rather than doing a survey or collecting data was the biggest and the easiest challenge that one could face. The BookMyShow website provided an easy method of retrieving opinion polls, audience movie reviews and ratings. The dataset created with user comments of the movies released in the year 2019 based on various genre classified from Hollywood to Bollywood movies that had mixed liking of the

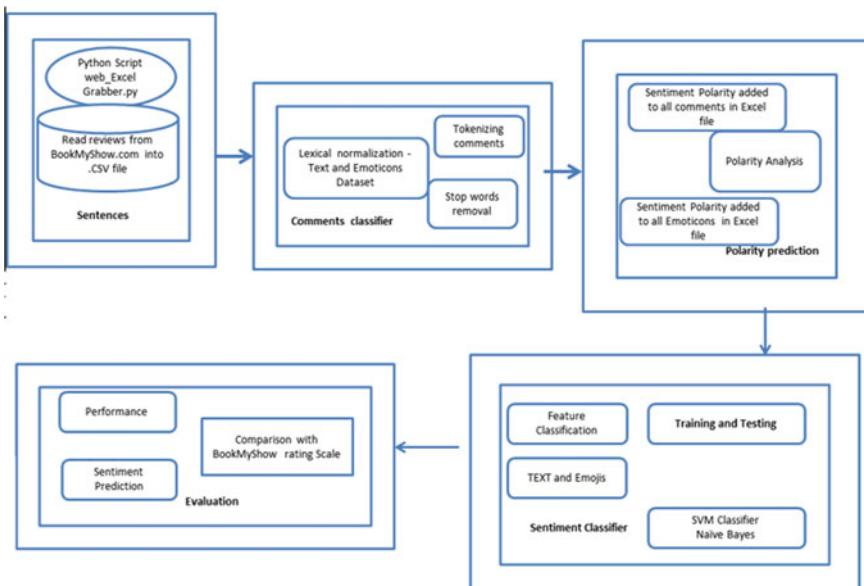


Fig. 1 System model—experimental layout of movie comments

movies. The age group of viewers was also considered for the type of movie selected (Table 1).

The website BookMyShow shows all the user reviews and movie rating related to a particular selected movie. The review data that has both text movie comments in English, Hindi, and other regional languages including emoticons from BookMyShow.com. The reviews of a movie are directly saved as html source code from the browser. The data is saved in the chronological sequence by adding to the latest comment at the end of the day. Multiple information can be drawn from the html source code like review, rating, movie genre, actor rating, storyline, etc., along with the date (Fig. 2).

Table 1 Overview of movies targeted as a data

Movie_title	Language	Genre	Target audience
Spiderman—far from home	English	Action, adventure, comedy	Kids, teens, young adults, family
Super 30	Hindi	Biography, drama	Young adults, family
Malaal	Hindi	Drama, romantic	Teens, young adults
Jhootha_Kahin_Ka	Hindi	Comedy, romantic	Family
The_Lion_King	English	Adventure, animation, fantasy	Kids, teens, young adults, family

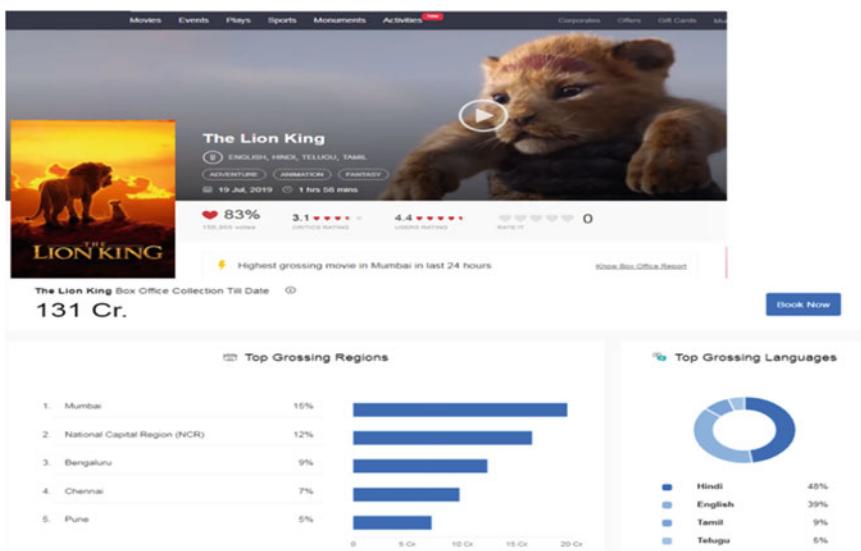


Fig. 2 Lexicon of movie characteristic features from BookMyShow website

Table 2 Dataset collection of movie reviews—user comments

Movie_title	Language	Genre	User reviews comments—as on July 2019
Spiderman –far from home	English	Action, adventure, comedy	4,606
Super 30	Hindi	Biography, drama	8,907
Malaal	Hindi	Drama, romantic	1,412
Jhootha_Kahin_Ka	Hindi	Comedy, romantic	664
The_Lion_King	English	Adventure, animation, fantasy	4,352
Total reviews			19,941

2.2.2 Lexicon of Dataset

The script Grabber.py helps in loading movie comments and ratings to an excel sheet to create a dataset for any number of movies all around India. As BookMyShow being an Indian company with access of online booking platform where user comments and movie rating can be retrieved [6,8]. Our dataset consist of five movies with a total of 19,941 reviews which we crawled from the movie ticket-booking website (Table 2).

The features of the comments for the Lexicon of the movie comments are analyzed both textual and contextual as in [5],

- words: binary feature for each word;
- emoticons: binary feature for each positive/negative emoticon;
- N-grams: binary feature for each n-gram (we only used bigrams).
- Lexicon: We have included two features based on our automatically generated movie reviews lexicon.

They represent the negative/neutral/positive and overall score of the movie review, obtained by aggregating the lexicon scores of each word in the review text. Therefore, the final feature set for the baseline system only used bag of words, emoticons, and the lexicon features.

The contextual features [9,10] include the contextual (metadata) features:

- movie length: numeric feature indicating the run length of the movie;
- Language: featuring which language the movie is catering to;
- Genres: indicator feature for each genre;
- Average user rating: numeric feature with the user's average movie review score;
- Critic score: numeric feature, current average score for this movie in box office;
- BookMyShow User score: numeric feature, current average score for this movie in BookMyShow.

3 Experiments and Evaluation

3.1 Pseudo Code of the Experimental Implementation

Algorithm 1- Genre based Movie comments features and review

- 1 Create Dataset
 - 1.1 Open the website and see all reviews
 - 1.2 Copy the html code from developer mode in browser to notepad and save with Encoding type UTF-8
 - 1.3 Save all the comments in an excel file
- 2 Loading raw data dictionary and returning array of words
 - 2.1 Tokenize the comments
 - 2.2 Filtering and removing non-alphabetical and stop words
- 3 Define Polarity Analysis for each comment
 - 3.1 if neg ≥ 0.4 :
 value = -1
 elif pos ≥ 0.4 :
 value = 1
 elif (neg > pos) and (neg - pos) ≥ 0.15 :
 value = -1
 elif (pos > neg) and (pos - neg) ≥ 0.15 :
 value = 1
- 4 Define SVM function, implementing SVM algorithm for both textual and emoticons comments.
- 5 Define Naive Bayes function for getting the sentimental result of the particular comment.
- 6 Compare the result for textual and emoticons analysis.
- 7 Compare the result with success percentage declared on Bookmyshow.

3.2 Screenshot of the Python Review Analyzer

The Python coding as shown below characterizes the result implementation; For all of the above models, we used sklearn modules along with NLTK library for sentiment classification (Fig. 3).

The comment received for the Hindi movie **Super30**.

Awesome movie, motivating, please go and watch...Super 30 is super. Hrithik Roshan played a super role as a Anand Kumar...dont be late, book your tickets...

```

#Loading raw data dictionary and returning array of words
def preprocessing(data):

    stop_words = set(stopwords.words('english'))
    processed_data = []
    for string in data:
        words = word_tokenize(string)
        words = [word for word in words if word.isalpha()] # tokenizing comments
        words = [w for w in words if not w in stop_words] # Checking if character is not alphabet remove it
        words = [w for w in words if not len(w) == 1] # Removing all stops words
        words = [w for w in words if len(w) == 1] # removing words of length 1
    processed_data.append(words)
return processed_data

#loading list of words and returnig...
def analysis(words,excel_location,sheet_name=""):
    excel_book = openpyxl.load_workbook(excel_location)
    sheet = excel_book[sheet_name]
    polarity_result = [] #it stores the result of polarity of every comment
    data = []
    neg = neu = pos = com = value = 0
    string = ''
    sid = SentimentIntensityAnalyzer()
    for i in range(1,sheet.max_row):
        if sheet.cell(i,2).value == 'Comment':
            initial_row_index = i+1
            break
    for i in range(len(words)):
        string = ''
        for j in words[i]:
            string = string + j + " "
        ana = sid.polarity_scores(string)
        polarity_result.append(ana['compound'])
        neg += ana['neg']
        neu += ana['neu']
        pos += ana['pos']
        com += ana['compound']

    neg, neu, pos, com = polarity_result

```

Fig. 3 Python implementation—preprocessing of movie reviews

Negative	Neutral	Positive value
0	0.288	0.712

Has a rating with Positive Polarity.

The comment received for the Hindi movie **Malaal**.

No acting skills and above that both of them look soo bad...here comes nepotism.
Feel so sorry for us who still went and watched the movie.

Negative	Neutral	Positive value
0.304	0.696	0

Has a rating with Negative Polarity.

The comment received for the Hindi movie **Jhootha Kahin Ka**.

“Awesome movie... masterclass by Rishi Kapur...Truely enjoyed his acting with my boys...loved it...0.5 star for many reasons...Go with your family to watch it.”

Negative	Neutral	Positive value
0	0.619	0.381

Has a rating with Neutral Polarity.

3.3 Supervised Learning Classification

The overall task in this project is for classification of reviews for a positive or negative feedback on the movie goers. The main classifier we have used is the SVM classifier as many research studies of SA has proved it to be the best classifier. Also for a supportive study, we have used the feature representations on NaïveBayes' classifier as a primarily case of text mining in combination with Bag of Words and N-Gram Modeling. The cross-validation experiments that were initiated have shown that not all features that we have introduced above were really relevant; thus, we created a selected set of highly-relevant features: BookMyShow Reviewer Score and for contextual analysis of words, emoticons, lexicons, N score, and average user rating.

We used this feature set when comparing the two machine [7] learning algorithms in classification such as SVM classification and Naive Bayes classifier.

Support vector machines use a hyper-plane isolating plane to make a classifier. SVM offers high probability of finding a precise answer for any principal information in a high dimensional feature space separating the hyper plane.

The Naive Bayes classifier is a likelihood classifier, it helps to assess likelihood that a record is sure or negative, in a specific setting, or the probability that an occasion to happen in the event that it was foreordained to be certain or negative.

$$P(\text{classification}_{\text{class}}|\text{element}) = \frac{(P(\text{element}|\text{classification}_{\text{class}}) \cdot P(\text{classification}_{\text{class}}))}{P(\text{element})}$$

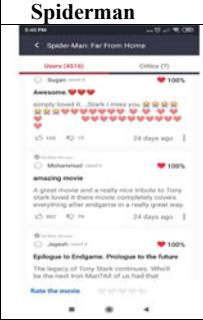
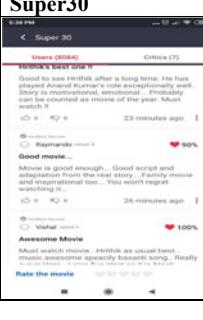
As the algorithm in Fig. 1 states that the sentences in the dataset are further tokenized, filtered tokens and added stop words that the text processing operators are used in this step. Tokenize splits the texts of a document into a series of words or tokens.

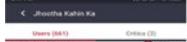
Stemming is a very important concept in natural language parsing. It allows one to reduce words to a common base form. The stopwords operator removes noisy common English words such as “a” and “the.” as a word vector to a document and the information set is separated into two sections, a preparation set and a test set as seen in Fig. 2.

The model is thus prepared on the preparation set just and its exactness is assessed on the test set. This is rehashed n number of times. The use of linear SVM and Naïve Bayes helps to quantify the exactness and check the accuracy of the review values.

4 Results and Discussion

4.1 Comparison Results of Website and Implementation of Sentiment Polarity

 <p>Spiderman</p> <p>Spider-Man: Far From Home</p> <p>Users (80110) Critics (2)</p> <p>100% 100%</p> <p>Awesome ❤️❤️</p> <p>simply loved it... Stark I miss you 😊😊</p> <p>100% 100% 24 days ago</p> <p>amazing movie</p> <p>A great movie and a really nice tribute to Tony Stark loved it there movie completely covers every aspect of the story and characters in a very good way</p> <p>100% 100% 24 days ago</p> <p>Epiclogue to Endgame. Prologue to the future</p> <p>The legacy of Tony Stark continues. Will be the next Iron Man/Md of us had that</p> <p>Rate the movie</p>	<pre>[MOVIE NAME : Spider-Man: Far From Home] [SVM - SVC ALGORITHM] -----[TEXT]----- >>> The movie is positive <-----[Emoji]----- >>> The movie is positive -----[NB-TRAIN] -----[TEXT]----- >>> Positive Percentage : 90.18835382755422 >>> Negative Percentage : 9.811644617244578 >>> Excellent -----[EMOJI]----- >>> Positive Percentage : 92.313593479234653 >>> Negative Percentage : 7.686406320763467 >>> Excellent</pre>
 <p>Super30</p> <p>Super 30</p> <p>Users (20044) Critics (7)</p> <p>90% 100%</p> <p>Hirsh's Best Movie</p> <p>Good to see Hirsh after a long time. He has given a great performance. Story is gripping, dialogues are sharp, music is great. Story is motivational, emotional... Probably one of the best movies of the year. Must watch it</p> <p>100% 100% 23 minutes ago</p> <p>Good movie...</p> <p>Movie is good enough... Good script and adaptation from the real story... Fairly movie and good entertainment... You would enjoy watching it.</p> <p>100% 100% 26 minutes ago</p> <p>Awesome Movie</p> <p>Must watch movie. Hirsh as usual best. Story is gripping, dialogues are sharp. Finally movie keeps you hooked till the end.</p> <p>Rate the movie</p>	<pre>[MOVIE NAME : Super 30] [SVM - SVC ALGORITHM] -----[TEXT]----- >>> The movie is positive <-----[Emoji]----- >>> The movie is positive -----[NB-TRAIN] -----[TEXT]----- >>> Positive Percentage : 92.01355628300596 >>> Negative Percentage : 7.5864437169949465 >>> Excellent -----[EMOJI]----- >>> Positive Percentage : 98.39606752110377 >>> Negative Percentage : 11.603132478896223 >>> Excellent</pre>

Malaal  Stars (147) Critics (4) 70% Malaal Malaal is reminiscent of sweet romances when hand written notes were actually a thing and the sentiments like these that sets the film apart. 100% Agree 50% Malaal Meeshaan Jassieh, Sharmeen Segal grow on you in this romantic drama. 100% Agree 40% Malaal Meeshaan, Sharmeen Segal debut in a vaginal romance about a rough diamond and the things that will set him straight, all set in 90's Mumbai. 100% Agree 60% Malaal Rate movie 40%	<pre> ----- MOVIE NAME : Malaal ----- SVM - SVC ALGORITHM -----[TEXT]-----> >>> The movie is positive -----[Emoji]-----> >>> The movie is positive -----[NB-TRAIN]-----> -----[TEXT]-----> >>> Positive Percentage : 66.2397862441095 >>> Negative Percentage : 33.76021373558905 >>> Good -----[EMOJI]-----> >>> Positive Percentage : 86.57653109575519 >>> Negative Percentage : 13.4254669042448015 >>> Excellent </pre>
Jhootha kahin ka  Stars (641) Critics (2) 100% Awesome movie This movie is really very amazing and I am very happy to see it. For me jhootha kahin ka is my favorite movie. I am a huge fan of this movie so it is very interesting <input checked="" type="radio"/> 1 <input type="radio"/> 2 6 days ago 100% Agree Extremely funny Overload of humor. Loved the film. A complete family entertainer. A must watch for the audiences. The punchlines boys are amazing. And the story line is the breakdown. <input checked="" type="radio"/> 1 <input type="radio"/> 2 6 days ago 100% Agree Time Pass Movie don't expect too much from this movie. this is just an ordinary movie with some light jokes and some nice songs. I would not recommend this. Rate the movie 40%	<pre> ----- MOVIE NAME : Jhootha Kahin Ka ----- SVM - SVC ALGORITHM -----[TEXT]-----> >>> The movie is positive -----[Emoji]-----> >>> The movie is positive -----[NB-TRAIN]-----> -----[TEXT]-----> >>> Positive Percentage : 80.71042760132357 >>> Negative Percentage : 19.289372398767433 >>> Very Good -----[EMOJI]-----> >>> Positive Percentage : 82.90598290598291 >>> Negative Percentage : 17.094017094017094 >>> Very Good </pre>

4.2 Conclusion and Future Scope

In this paper, we have analyzed the sentiment polarity and it's of user reviews about movies. We evaluated the linear SVM and Naïve Bayes algorithms on the movie user review dataset to check the accuracy of the different genre and opinions.

On the basis of analysis, the result demonstrated that the movie drama has the high accuracy rate among the different genre of the movies.

The sentiment orientation describes that the user prefers to watch drama type of movie. The graph shows the polarity of the different words.

The future scope of the work is to analyze the trend of the movie goers and provide a predictive analysis system for the movie crew of directors on analyzing the viewer recommendations of movie choices.

References

1. Bhaskar MS, Ranjana K (2016) Recent genre based categorized comparisons of bollywood movies through radar plots. (IJITR) Int J Innov Technol Res 4(2):2844–2851
2. Kesharwani A, Bharti R (2017) Movie rating prediction based on twitter sentiment analysis. J Adv Comput Commun Technol 5(1). ISSN 2347-2804
3. Blatnik A, Jarm K, Meza M (2014) Movie sentiment analysis based on public tweets. Elektrotehniški Vestnik 81(4):160–166
4. Piryani R, Gupta V, Singh VK (2017) Movie prism: a novel system for aspect level sentiment profiling of movies. J Intell Fuzzy Syst 32(5):3297–3311. ISSN 1875-8967
5. Ng Yiu-Kai (2017) MovReC: a personalized movie recommendation system for children based on online movie features. Int J Web Inf Syst 13(4):445–470
6. Film Industry in India—Statistics and Facts (2018) Statista Research Department, 10 Dec 2018. <https://www.statista.com/topics/2140/film-industry-in-india>
7. Barnes J, Velldal E, Øvrelid L (2019) Improving sentiment analysis with multi-task learning of negation. Nat Lang Eng 1(1):1–25
8. Mittal S interview with Viraj Patel, VP-Technology, BookMyShow. How IT helps BookMyShow provide a seamless online ticketing experience. Newsletter 27 May 2016 <https://cio.economictimes.indiatimes.com/news/strategy-and-management/>
9. Kapukaranov B, Nakov P (2015) Fine-grained sentiment analysis for movie reviews in Bulgarian. In: Proceedings of recent advances in natural language processing, Hissar, Bulgaria, 7–9 Sep 2015, pp 266–274
10. Rana S, Singh A (2016) Comparative analysis of sentiment orientation using SVM and Naïve Bayes techniques. In: 2016 2nd international conference on next generation computing technologies (NGCT) comparative analysis of sentiment orientation using SVM and Naïve Bayes Techniques

Chapter 61

Detecting Offensive Text on Facebook Using Natural Language Processing and Machine Learning



Ameya Kasbekar, Rashmi Rana, Vidhi Shah and Abhijit R. Joshi

1 Introduction

Facebook has become one of the most widely used social networking sites across the world. It provides a platform for its users to connect with their colleagues, family and friends residing in any part of the world. According to recent statistics, Facebook currently has more than 2.13 million daily active users [3]. The statistics reveal that people spend an average of 35 min on Facebook every day [3]. Approximately, 400 users sign up for Facebook every minute [3]. The hype of using Facebook has increased immensely among adolescents. Several times, it may happen that while accessing Facebook, an adolescent might come across something, which is inappropriate. Such content may have a negative and long-lasting effect on the mind of a teenager. According to studies, teenagers, who come across profane content, are more likely to use swear words and become aggressive toward others.

Facebook owns a community standard group, which takes care of any profane content being posted and simply removes it. This system has two drawbacks. The first drawback is readability and the second drawback is it fails to detect offensive content while looking for its meaning. Elaborating on the first issue, let us consider a Facebook post written as “Happy women’s day to all the beautiful women out there. Ladies this is your day enjoy it. All the moron men you guys are assholes.”

A. Kasbekar · R. Rana · V. Shah · A. R. Joshi (✉)
Dwarkadas J Sanghvi College of Engineering, Mumbai, India
e-mail: abhijit.joshi@djsce.ac.in

A. Kasbekar
e-mail: kasbekarameya@gmail.com

R. Rana
e-mail: rashmi28.rana@gmail.com

V. Shah
e-mail: vidhishah856@gmail.com

In this case, the existing system used by Facebook will tag this post as abusive and entirely delete it from user's timeline. This seems to be a serious issue because the message wishing women's day had good intentions and still would be deleted along with the offensive content. This affects the readability of the post and its purpose, due to the presence of some profane words. Coming to the second drawback, consider the example "Mona you are such a cry baby, stop being a cry baby." In this case, the existing system fails to identify this sentence as offensive, because it lacks the presence of any profane or abusive word. However, the statement "Mona you are such a cry baby, stop being a cry baby," may sound offensive to the targeted person.

To overcome these issues in the existing system, we have proposed a system wherein lexical-based and semantic-based methods will be combined.

The rest of the paper is organized as follows. Section 2 throws light on the related work done in this area and the issues in the existing system. Section 3 explores the proposed approach along with design and implementation details. Section 4 takes a walk-through of our system as a user. The paper ends with conclusion and direction for future work.

2 Related Work

In this section, the prominent existing systems and some of the useful algorithms are discussed. The section ends with the observations on these systems. Now, let us see first the useful algorithms.

2.1 AHO–Corasick

AHO–Corasick, a string pattern-matching algorithm, which is used to detect offensive word exists in a given text or not [1]. The algorithm searches for the pattern (here abusive words) in the given text. If the pattern/word is found, then the occurrences of pattern along with the index numbers are stored in an array. Now, the index number is used to trace back the location of the pattern in the given text and replaces this pattern by some special characters.

2.2 N-Gram

Given a text, in which a group of continuous sequence of n -items is called as N-grams [2]. N-grams are useful in dividing text and words into n chunks. Generally, 2, 3, 4 and 5 N-grams are used to find the number of occurrences of each of these keywords in the given text. If the related words are separated by too long text, in that case N-grams find it difficult to explore such extreme cases [3]. One can address this

issue just by increasing the value of N. But it will affect the outcome of the overall process resulting in more false positives and also decreases the processing speed of the system.

2.3 *Bag of Words (BoW)*

BoW is an approach generally used in the case of an automated system when one has to classify the document on the basis of insulting or abusive content [3]. In terms of Linguistics expression, since insulting or abusive words have extreme nature, such contents or words are generally considered as a part of extreme subjective language. To identify such extreme words from a given document, a set of rules are formed. These rules are used to extract the meaning/semantic information of a given sentence. In this extraction process, the general semantic structure of a sentence is used to separate out abusive contents from the given document. But the issue with such system is it can only identify and detect those abusive or insulting sentences having related words or phrases, which are currently being part of the lexicon list [1].

Currently, there is no dedicated automated system exists for detecting offensive content posts on social media sites. The following manual systems/processes followed for detecting offensive content.

2.4 *Manual Check by Content Reviewers*

Facebook has a group of content reviewers, a separate department for dealing with cyberbullying cases. The aim of this department is to prevent Facebook users from cyberbullying. Currently, there are approximately 7,500 employees working in the content reviewing department of Facebook [4]. If a Facebook user comes across any inappropriate comments or posts being posted on his/her timeline, he/she can report a complaint to Facebook by flagging the respective post or comment. The content reviewing department manually checks every flagged comment or post by any of the Facebook users. There are more than 2.13 billion monthly active Facebook users, who make millions of posts and comments every hour out of which thousands of posts and comments are flagged as inappropriate every day. As a result, one content reviewer spends approximately 10 s on each flagged comment/post [4]. Depending on the Facebook terms and conditions, the decision of keeping or removing a post/comment is made. It is not necessary that all the reported posts will be removed by Facebook; a post/comment may be inappropriate for a user but may be appropriate as per Facebook terms and conditions. In fact, Facebook does not remove most of the flagged content. Another drawback is, by the time a flagged post/comment is reviewed by any content reviewer, the post goes viral and the damage is being already caused.

2.4.1 Pitfalls in Manual Check by Content Reviewers

- i. The time invested by content reviewers on a single post is less than 10 s; as a result, the judgment made by the reviewers is not accurate.
- ii. Every day millions of complaints are being registered, since there is no automated system being used; no matter how many people Facebook hires, they will never be able to look after everything.
- iii. Since Facebook is used by people of different age groups ranging from kids to adults, adolescents often come across certain inappropriate or offensive content.

2.5 *Dictionary-Based System*

Dictionary-based system is an automatic system used for dealing with abusive content in a given document. In this method, a dictionary of abusive words is used for detecting abusive words in given document. A set of rules is created to extract the semantic information of a given sentence in order to distinguish between abusive and non-abusive sentences. Appen and Internet Security Suite are some of the software packages, which are based on the said approach. These software packages are used for detecting and filtering online abusive and insulting contents. These software packages simply blocked web pages and paragraphs having abusive words [5]. The document is scanned for the words blacklisted in the dictionary. If a match is found, the respective word is either completely hidden by using special characters (e.g., ****) or partially substituted by special characters (e.g., f***). However, this method badly affects the readability of the document. Brutally hiding or removing words makes the sentences meaningless. In addition, partially hidden words are often easy to guess. Thus, we can say that although bag of words seems an easy approach on the scale of accuracy, it gives quite disappointing results.

2.5.1 Pitfalls in Dictionary-Based System

- i. This system affects the readability of the website, since it simply blocks abusive words as well as paragraphs.
- ii. The system takes into consideration just the presence of abusive words and does not evaluate the context in which it is said.
- iii. For example, the sentence “you are such a cry baby” will not be recognized as offensive because there is no presence of any abusive word, but it is offensive.

To address these issues in existing system, we decided to develop a system addressing these concerns. Let us see our approach for addressing these issues in the next section.

3 The Proposed Approach

The system is built up of two stages. In the first stage, the corresponding comment or textual post is checked for the presence of any profane words. We have constructed two dictionaries, one with high-intensity profane words, and the other with low-intensity profane words. The words in a comment or post are matched with the words in both the dictionaries. If a match is found, then the corresponding sentence in a post or comment is deleted. Let us consider the post “Happy women’s day to all the beautiful ladies out there. Ladies this is your day enjoy it. All the moron men you guys are assholes.” In this case, after filtering the particular post would be presented to the viewer as “Happy women’s day to all the beautiful ladies out there. Ladies, this is your day enjoy it.” In the second stage, if no match is found between the words in a post or comment and the words present in any of the dictionaries, the semantic analysis of the post or comment is carried out. If the sentence sounds appropriate semantically, then it is displayed as it. However, if the sentence is semantically offensive, then the corresponding content will be removed. Take the example “Mona you are such a cry baby, stop being a cry baby.” In this case, since the sentence is semantically offensive it will be deleted. Now, let us see the design details of the system.

3.1 Architecture of the System

The system has a three-tier architecture (see Fig. 1); the User Interface is at the client side which is the first tier, the middle tier performs the profane content detection task and the third tier consists of all the databases.

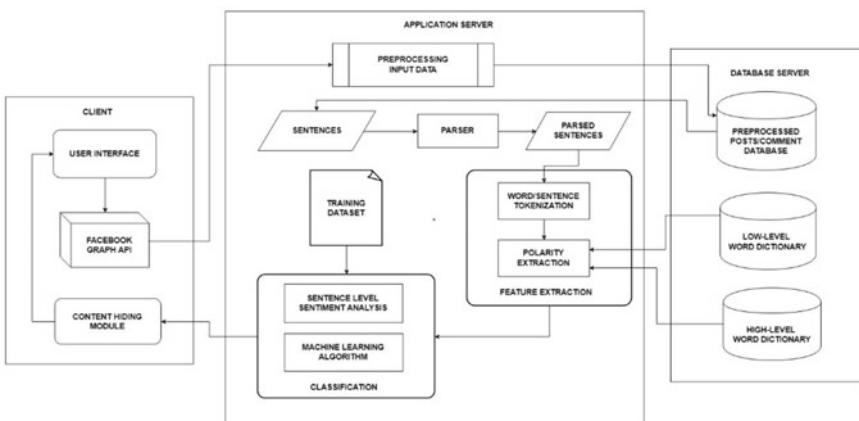


Fig. 1 System architecture

A user logs into the system using the Web application. As soon as the user logs into the system, a PHP call is made to the Facebook using Graph API on behalf of the user by the system. All the data from the user's profile is collected; this collected data is then pre-processed by the middle tier and is stored into pre-processed post/comment database. The system retrieves this data from the database and broken down into sentences and supplied to the parser. The parser performs sentence-level parsing on the received data and forwards it to the tokenizer. After tokenization, polarity extraction is performed using low-level and high-level dictionaries stored in the database server.

After extracting features of the received content, sentence-level sentiment analysis is performed using the training data set. Machine learning algorithm is applied in order to detect profane content. Once the task of checking each and every sentence for profane content is over, the obtained results are stored in the database.

The final step of hiding the profane content takes place in the client tier, the stored results are retrieved from the database by the content hiding module and it only displays data, which is free from all sorts of profanity. Now, let us see the functionality of various modules of the system.

- i. **User Interface:** This module is developed using HTML, PHP and CSS Web technologies. The user logs into his/her Facebook account using User Interface module. As soon as the user logs into the account, this module makes a PHP call to the Facebook Graph API module. The User Interface primarily instructs the Facebook Graph API to extract the comments and posts from user's Facebook timeline. The User Interface module is responsible for displaying the filtered Facebook profile content from the user's timeline on the Web application.
- ii. **Facebook Graph API:** The system uses version 12 of Facebook Graph API to extract data from user's Facebook timeline. Facebook Graph API is a primary way for websites and apps to read and write to the Facebook social graph. Data provided by the Facebook Graph API has a structure of Facebook social graph. The Facebook social graph is a data structure designed by Facebook that works like a connected graph data structure. Each element of graph is hence linked using edges and nodes containing all the data. For example, a post or a comment can be considered as a node and each post is connected to its comments using edges or links. The data extracted using this module is provided as an input to the data pre-processing module.
- iii. **Pre-Processing:** The pre-processing module takes the Facebook Graph API data as its input. This module performs the task of removing Not A Number (NAN) values, NULL values and stop words from the extracted data. After the data is pre-processed, it is stored in the third tier, i.e., in the database server. Further, the Natural Language Parser for syntactic feature extraction uses this data.
- iv. **Natural Language Parser:** As soon as the task of the pre-processing module is completed, it saves all the data to the Facebook User Database module. The Natural Language Parser then extracts this data from the Facebook User

Database module to study the grammatical structure of each sentence present in each post and comment.

In order to have a definitive meaning for any sentence, it should consist of a combination of components such as nouns, verbs, adverbs, adjectives, pronouns, conjunction and their subcategories [6]. This module uses a Natural Language Parser to determine the relation between these components of every sentence. It is a prerequisite step for extracting the syntactic features and construction of a parse tree.

- v. **Word and Sentence Tokenization:** Once the parse tree is constructed and the Natural Language Parser module derives relation between each component of every sentence, the parsed sentences are provided as an input to the Word and Sentence Tokenization module.

This module treats each word from the database as an independent entity. As we know that tokenization is the process of dividing each word of an input text into small pieces called tokens. The resulting tokens are then passed to PoS tagging, wherein each token is tagged with parts of speech. These Pos tagged tokens are used by parser.

Consider an example: “**The crying black baby falls in the big pit.**”

In this example, the string is split into the nine tokens (words) on the basis of “space” as a delimiter.

The word tokenizer represents an input sentence using XML format. Hence, the tokenizer in the following format will represent the above example:

```
<sentence>
<word>The</word>
<word>crying</word>
<word>black</word>
<word>baby</word>
<word>falls</word>
<word>in</word>
<word>the</word>
<word>big</word>
<word>pit</word>
</sentence>
```

- vi. **Polarity Extraction:** The Word and Sentence Tokenization module is responsible for providing an input to the Polarity Extraction Module. This input contains posts and comments but in the form of a list of words for each sentence and list of sentences for each paragraph.

The Polarity Extraction Module treats each word from the database as an independent entity. Each word is searched and compared with the words in an abusive language dictionary. Based on the comparison, if any word from the dataset matches any of the words from both of the dictionary, the word is considered an offensive polarity for the user.

To detect offensive words, lexical features do still well even in the absence of syntactical structure of the whole sentence. But when a sentence has same

words appear in different orders, in such cases lexical feature failed to detect offensiveness of the sentence. Here, the relation tree that is obtained from the Natural Language Parser is used to consider the structure of the sentence. This helps the classification module to further classify the sentences of a post or comment into an offensive or non-offensive category.

Let us now have a look at the algorithm used by the Polarity Extraction Module and how it processes each post/comment.

Input: Paragraph of the post/comment

Calculate number of sentences N in a paragraph

```

For i = 1 to n do
    Split Sentence i in list of words (L).
    Calculate length of L as m.
    For j = 1 to m do
        Compare jth word with each word of Offensive Word Dictionary.
        If jth word in Offensive Word Dictionary do
            Flag = 1
        If Flag is equal to 1 do
            Consider the Sentence i as offensive and remove it from the Para-
            graph.
        Update the result of the algorithm in the Post/Comment Database.
    End

```

Output: The result obtained from the algorithm will be a paragraph without offensive sentence. The generated result is stored in database and used for displaying the post/comment at the front end.

- vii. **Facebook User Database:** All the pre-processed comments/posts are stored in Facebook User Database. It is also used to store details of each user such as user id, user name and email id. In order to relate a post to its comments and replies of that comment, all the tables are combined into a universal table using MySQL joins.
- viii. **Classification:** This module is the heart of the architecture of our system as it performs the core function of classifying offensive posts and comments from non-offensive ones. The classification module performs its operation based on the processed data provided by the Polarity Extraction Module. As this module performs such a fundamental task of the system, it is necessary that an input to this module should be completely pre-processed by all the preceding modules.

The classification module is primarily divided into two major components. The sentence-level sentiment analysis module is designed to work on individual sentence and its position within a paragraph. This module detects all the offensive sentences in a paragraph and removes those offensive sentences only. To do so, first it recognizes the relationship and meaning between two or more sentences and removes an offensive sentence in such a way that the neighboring sentences are not affected by its removal.

The second major component of the classification module is a machine learning algorithm. This module uses a support vector machine algorithm to

classify offensive posts and comments. It is trained on a dataset of 10,000 sentences and paragraphs combined. Each sentence in this dataset is an offensive and insulting sentence.

- ix. **Offensive Content Hiding Module:** The job of Offensive Content Hiding module is to fetch the result generated by the combined efforts of all the preceding modules and perform the operation of hiding the offensive content for the same. This module hides only those sentences in comments and posts that are tagged as offensive in relation to the user. It is a part of the User Interface module as it works in coordination with the User Interface.

4 Working with the System

Having looked at the system design and implementation details, let us take a walk-through of the system as a user. Figure 2 shows the login page of the Web application, through which a user logs into the system. The user will then be redirected to the Facebook login page.

Figure 3 shows a snapshot of the actual Facebook login page. The user will enter his/her login credentials on this page, and after authentication, the user will be logged into his/her profile.

Once the user gets logged into his/her profile, all the user's data is fetched from his/her timeline through Facebook Graph API call and displayed on the Web application. Figure 4 shows the user's Facebook data being displayed on the Web

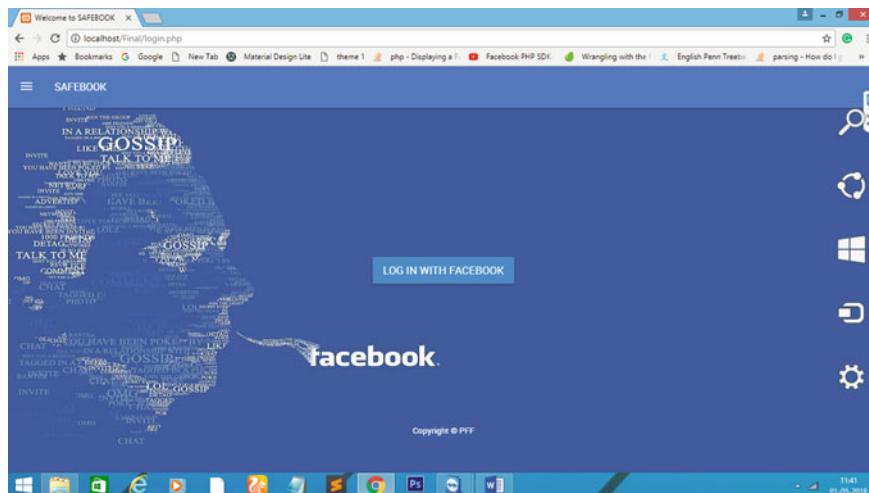


Fig. 2 Web application login page

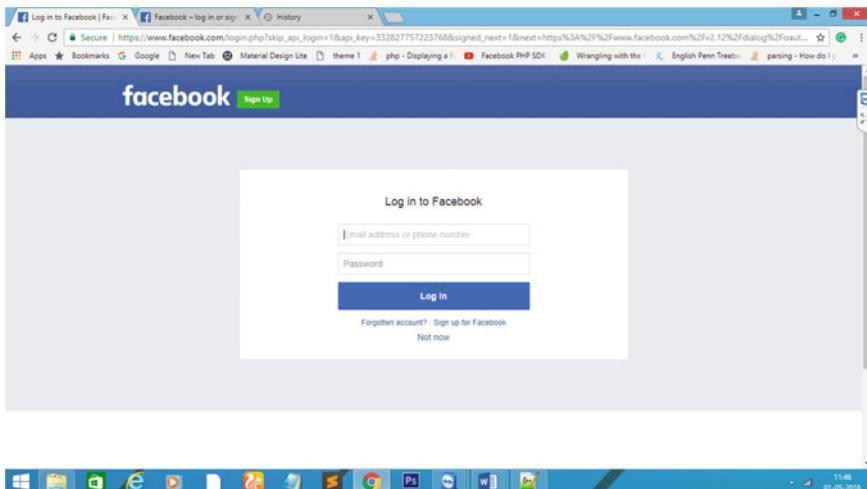


Fig. 3 Facebook login page

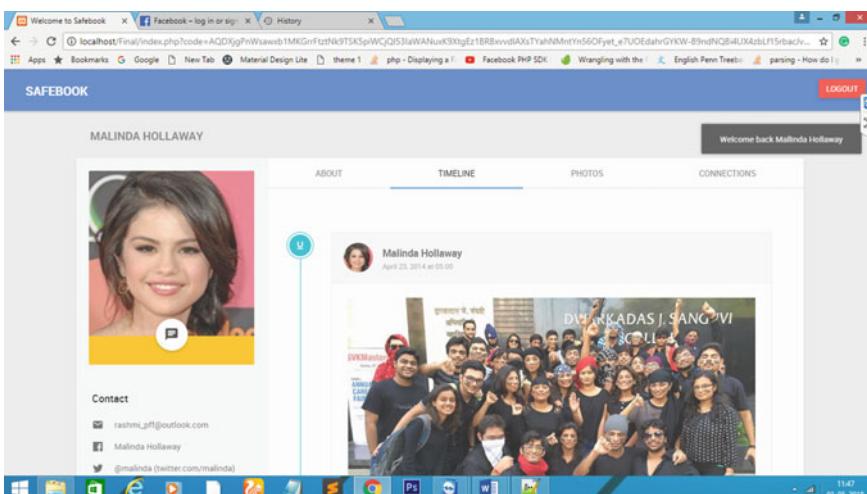


Fig. 4 User's Facebook timeline displayed on Web application

application. The data displayed on the Web application is filtered data, which is free from all textual profanity.

Figure 5 shows the snapshot of a post posted on the logged-in user's Facebook timeline. The sentence "**All the moron men, you guys are assholes**" in the post is highly profane. Such kind of post when posted on a globally used platform like Facebook goes viral in very few seconds and leads to many controversies, internet war, cyberbullying, etc.

Fig. 5 A profane post displayed on user's Facebook timeline



Figure 6 shows the snapshot of the filtered post displayed on the Web application. This snapshot shows the filtered version of the same post referred in Fig. 5. By comparing Figs. 5 and 6, we can see the difference between the post, which is displayed on Facebook and on our Web application. The sentence "All the moron men, you guys are assholes," is displayed as it is whenever a user logs into his/her profile using Facebook. Hence, whenever a user logs into his/her profile using our Web

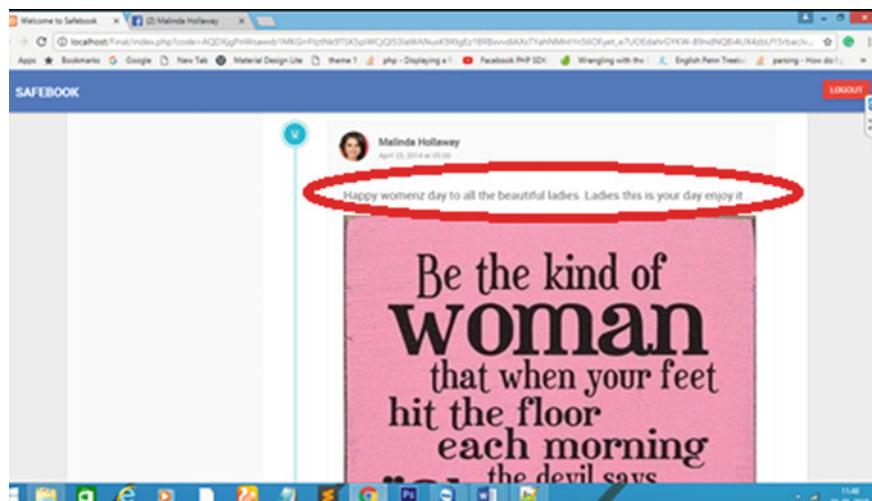


Fig. 6 A profane post displayed on our Web application

application, such profane sentences are hidden and it is ensured that the readability of the user's profane free content does not get affected.

5 Conclusion and Future Scope

We have developed a paradigm, which can be used by any Facebook user for hiding profane and offensive content. Thus, they will be able to access their Facebook posts and comments without being a victim of cyberbullying without affecting the readability of the content.

In future, one can expand our system by detecting racism, nationalism and high-level sarcasm. One can also plan to enhance the system by detecting the presence of profane content in images and videos.

Accuracy of detection can be improved by refining the lexical and machine learning modules. In order to make the system capable of detecting the latest profane content, the dictionary can be continuously updated with most recently used profane words, in turn increasing the data set of offensive content.

One can also improvise the User Interface and Interaction with the Web application. The overall accuracy of the system can be improved by testing the system on more number of users.

References

1. Manwatkar PM, Yadav SH (2015) An approach for offensive text detection and prevention in social networks. In: IEEE sponsored 2nd international conference on innovations in information embedded and communication systems, ICIIIECS'15
2. Chavan VS, Shylaja SS (2015) Machine learning approach for detection of cyber-aggressive comments by peers on social media network. International conference on advances in computing, communications and informatics
3. Chen Y, Zhou Y, Zhu S, Xu H (2012a) Detecting offensive language in social media to protect adolescent online safety. In: Privacy, security, risk and trust (PASSAT), 2012 international conference on and 2012 international conference on social computing (SocialCom). IEEE, Sept 2012, pp 71–80
4. Facebook's Fight to Block Violent and Hateful Content. <https://learningenglish.voanews.com/a/facebook-faces-difficult-task-of-blocking-violent-and-hateful-content/3882423.html>
5. Spertus E (1997) Smokey: automatic recognition of hostile messages. In: AAAI'97/IAAI'97 proceedings of the fourteenth national conference on artificial intelligence and ninth conference on innovative applications of artificial intelligence, pp 1058–1065
6. Mahmud A, Ahmed KZ, Khan M (2008) Detecting flames and insults in text. In: Proceedings of 6th international conference on natural language processing (ICON-2008), CDAC Pune, India, 20–22 Dec 2008

Chapter 62

A Proposed Approach for Financial Investment Recommendation and Decentralized Account Management



Harshita Khandelwal, Harsh Jain, Shreeyaa Agrawal and Vinaya Sawant

1 Introduction

For a person without a financial background, the ambiguity in choosing among the wide range of investment options is a major problem. These options must be thoroughly assessed in accordance with the goals and available capital on factors such as risk involved. A misstep could result in expensive mistakes such as high hidden fees or poor investment choices.

People generally turn toward a financial planner for aid. A financial planner helps in navigating through complex financial situations, answer questions as they come up and may provide continuous comprehensive financial planning which is prepared for market fluctuations. Asset management also includes the management of the risks connected with the use and ownership of the assets. Proper assessment of the assets can help to identify the risks involved and allocate current assets across investment opportunities based on the risk preferences and the desired goals.

Asset management requires user's data based on which appropriate recommendations must be made. However, there are often cases where people tend to hide their monetary details in order to get away with paying taxes [1]. Several details are hidden in order to protect personal interests. People tend to not trust the banking authorities with their data. This problem has gathered attention due to recent exposures of data being leaked from the banks [2]. Banks need people to agree to certain policies, which can sometimes be time-consuming and inconvenient. It is a general observation that people possess accounts at different banks in order to tackle this problem. The hectic task of managing several accounts can result in confusion and chaos.

H. Khandelwal (✉) · H. Jain · S. Agrawal · V. Sawant
Dwarkadas J Sanghvi College of Engineering, Mumbai, India
e-mail: harshitakh@gmail.com

In this paper, we try to address these issues by proposing a decentralized account management system using blockchain technology and an investment recommendation model. The rest of the paper is organized as follows. In Sect. 2, we present the findings of the literature survey. In Sect. 3, we describe our proposed system. The paper ends in Sect. 4 with conclusion.

2 Literature Survey

This paper [3] takes the personal information of the customers such as their spending capabilities, their knowledge of the financial domain and risk tolerance and compares it with the other users having a similar portfolio and recommends assets based on the investments made by those customers.

A group of banks in Singapore has also been developing a payment system prototype using DLT in which bank users can exchange currency with one another without lengthy processing times, expensive processing fees or intermediaries. This initiative is known as Project Ubin [4].

In this research [5], the authors proposed the advantages of using blockchain technology in payments system for a competitive advantage in the banking industry as follows: easy to use, privacy, low transaction fees and real time.

This paper [6] makes use of recurrent neural network (RNN) followed by convoluted neural network (CNN). It uses RNN for prediction of stock prices using sentimental analysis of news data and historic data. These results are represented in a graphical format which is fed to the CNN for analyzing its nature. These predictions are then applied to a knapsack algorithm to predict the appropriate stocks for investment.

This paper [7] focuses on developing a recommender system for suggesting distinguished investment plans for different individuals. Firstly, it calculates the risk level of securities and risk preference of investors using VaR method. It then takes into consideration the risk preference and historic behavior of the user using collaborative filtering and matches it with the nearest neighbors.

This paper [8] makes use of the support-confidence framework over the traditional association rule mining techniques to generate association rules as they overcome the traditional disadvantages of ARM using domain-specific techniques. These rules are then employed to suggest investment options. Also, they have implemented novel methods like using fuzzy logic and the concept of time lags to generate datasets from actual data of stock prices. Table 1 represents the summary of the literature survey.

Table 1 Summary of literature survey

S. No.	Name	Description
1	Recommending personalized asset investments through case-based reasoning: the SMARTFASI system	This approach makes use of a combination of different distance calculating measures on various parameters such as age, marital status, asset allocation to produce a ranking of portfolios who possess similarities with the current user
2	Project Ubin	The whitepaper on Project Ubin describes the implementation of an efficient inter-bank transaction system
3	The affecting factors of blockchain technology adoption of payments systems in Indonesia banking industry	This paper highlights the need and benefits of a blockchain-based payment model by conducting research in Indonesia's banking industry
4	An ensemble stock predictor and recommender system	This system combines the power of recurrent neural network (RNN) and convoluted neural networks (CNN). It makes use of a RNN model for news analysis of stocks, another for historical data analysis of stocks. Their results are fed to an ensemble predictor whose results are used by the CNN to read peaks and dips which helps in recommending stocks
5	An investment portfolio recommendation for individual e-commerce users	This paper focuses on different collaborative filtering algorithms such as user-based, item-based and hybrid collaborative filtering along with VaR method to recommend appropriate portfolio options
6	An association rule mining based stock market recommender system	This system combines association rule mining along with fuzzy logic to implement weighted fuzzy ARM to predict relations in BSE stocks on the basis of weights and other factors, thereby deriving suitable conclusions about these stocks

3 Proposed System

After an extensive literature review of the existing systems, algorithms and technologies, we propose to build a system that is efficient, cost-effective and time-saving. The management of several accounts can be made easy by forming a consortium/permissioned blockchain of banks. This consortium would comprise of all the banks connected through a channel. Each bank can then form a separate channel for its users. The new users would be added to the consortium after going through a

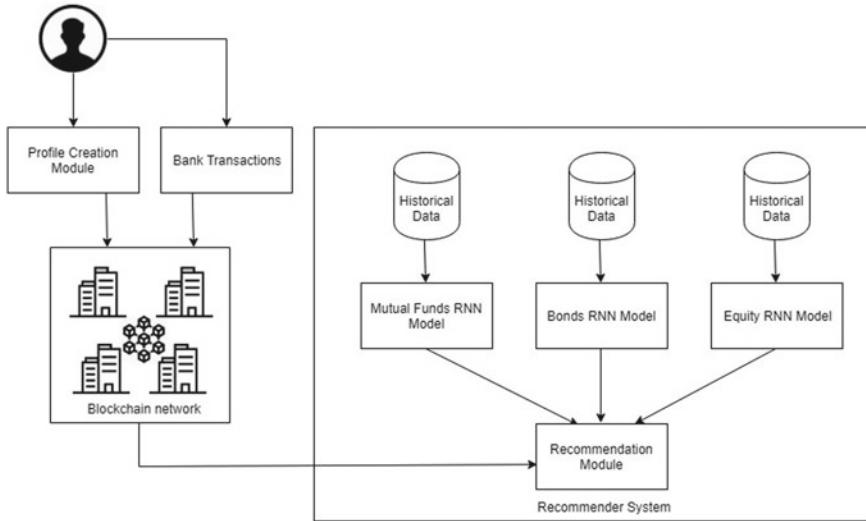


Fig. 1 Proposed system architecture

one-time verification process. This consortium would ease the process of cross-bank transactions and reduce the time as well as the cost consumed in the process. The customers can only view their account details and transactions, while the bank can view transactions of the customers. The user would not be able to commit any tax fraud.

The proposed system would recommend among the three asset classes, which are bonds, mutual funds and equity. These three asset classes are one of the most popular investment options. We intend to analyze data from these asset classes, combining them with the user's profile data obtained from the blockchain to suggest appropriate choices for investment. Some of these parameters like lock-in period would be assumed/taken from user. Fig. 1 shows the architecture of the proposed system. It includes all the rudimentary modules.

The proposed system shall have two major steps. First, analysis of the historical data from available datasets to predict future values of the asset classes. We intend to do so by creating a recurrent neural network (RNN) model since it uses long short-term memory (LSTM) which provides better accuracy. An RNN (see Fig. 2) [9] is a type of neural network famous for its ability to use previous (historical) data to predict values. As seen in the figure, it not only considers data from its current iteration but also from all its previous ones as well. This approach hits a roadblock as it leads to the vanishing gradient problem, i.e., when the gradient value for the error becomes vanishing small due to which the network is not trained further. So, to overcome this problem, we use the RNN-LSTM model (see Fig. 3). Here, each layer has a memory unit attached to it. Hence, data stored in these units are also utilized which help us in forming long-term dependencies. This proves as an added advantage in predicting values.

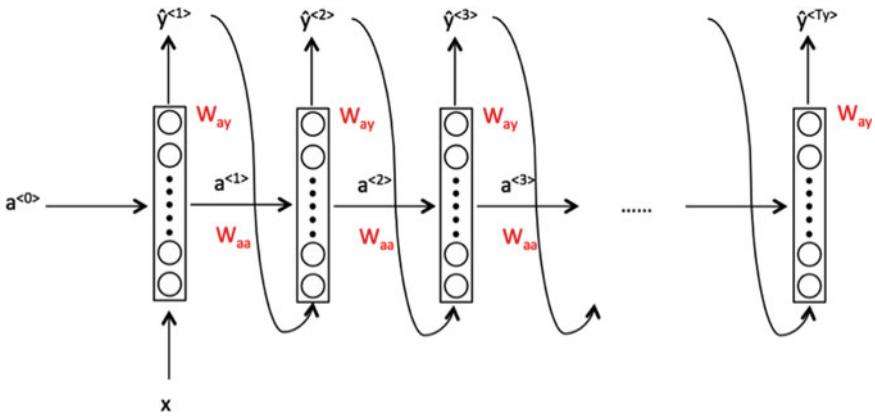


Fig. 2 Recurrent neural network—RNN

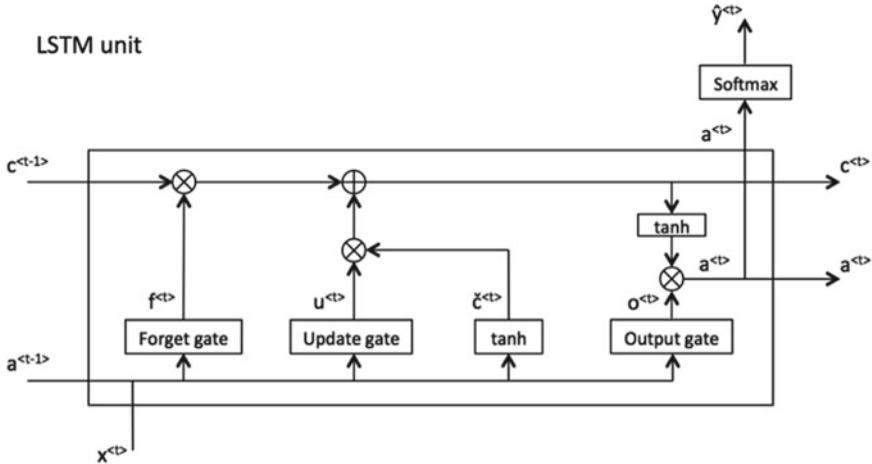


Fig. 3 Long short-term memory unit

In the second step, the predicted stock, equity and mutual fund prices would be fed into a recommender system. The recommender system module would consider factors such as age, risk, lock-in period which would be obtained from a permissioned blockchain. The permissioned blockchain network would be used by the user to perform transactions, involving either a single or multiple bank. All the banks shall be a part of this consortium. Since all these transactions will be performed on a single blockchain network, it would eliminate the need of intermediaries for any transaction involving multiple banks. Managing multiple accounts would become easier via a single interface. The use of blockchain would ensure non-repudiation and increased security of the system. Since all the transactions would be published on the blockchain, the transaction history could also be easily obtained. Transaction

history will be used in calculating risk that a user might be willing to take. This calculated risk along with other factors obtained from the user profile would be fed into the recommender system. The recommender system will use logistic regression to predict the best possible investment option for the user.

4 Conclusion

In this paper, we identified problems in the current scenario, performed a literature survey based on which we formulated an approach to design our system. The system proposes a technique to design and implement a model to predict the most suitable investment option from three asset classes for a user. The approach deals with the usage of historical data obtained from over a 20-year period. The proposed model will use recurrent neural networks (RNN) and long short-term memory (LSTM) to learn and predict the future trend for the three asset classes. The system would personalize the experience of asset recommendation by using users' data obtained from a permissioned blockchain. By leveraging properties of blockchain such as its tamper-proof nature and non-repudiation, we will ensure reliability and robustness of the system. The data extracted from the blockchain along with predicted prices of the various asset classes will be fed into the recommendation model to provide suitable asset investment options.

References

1. Das PK (2018) An insight into black money and tax evasion—Indian context. *J Int Bus Res Mark* 3(4):30–39
2. Capital one data theft impacts 106 M people [online]. <https://krebsonsecurity.com/2019/07/capital-one-data-theft-impacts-106m-people/>. Last accessed on 31 July 2019
3. Leonardi G, Portinale L, Artusio P, Valsania M (2016) Recommending personalized asset investments through case-based reasoning: the SMARTFASI system. In: 2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI), San Jose, CA, pp 804–811
4. Project Ubin, Monetary Authority of Singapore
5. Taufiq R, Hidayanto AN, Prabowo H (2018) The affecting factors of blockchain technology adoption of payments systems in Indonesia banking industry. In: 2018 international conference on information management and technology (ICIMTech), Jakarta, pp 506–510
6. Hegde MS, Krishna G, Srinath R (1985) An ensemble stock predictor and recommender system. In: 2018 international conference on advances in computing, communications and informatics (ICACCI), Bangalore, pp 1981–1985
7. Li X, Yu C, (2017) An investment portfolio recommendation system for individual e-commerce users In: 24th international conference on production research
8. Paranjape-Voditel P, Deshpande U (2011) An association rule mining based stock market recommender system. In: 2011 second international conference on emerging applications of information technology, Kolkata, pp 21–24
9. DeepLearning series: sequence models [online]. <https://medium.com/machine-learning-bites/deeplearning-series-sequence-models-7855babeb586>. Last accessed on 31 July 2019

Chapter 63

User-Based Personalized Text Summarizer



Pratik Nalage, Jay Parekh, Arjav Metha and Abhijit R. Joshi

1 Introduction

It has always been a next to the impossible and very time-consuming task to summarize and sort through data on the Internet manually and generate a summary adhering to all the semantics. There are various types of text documents used by teachers, students, authors and many more end users. They do not have the time to sit and read every line and then decide what the essence of the document is. The amount of information available is vast and increasing rapidly. To help the end users, our model summarizes the document according to their needs. It identifies the relevant information and presents it to the users as per his/her needs in a simplified way. Based on the type of user selected, our model uses different algorithms to summarize the document. It uses the extractive or abstractive model to summarize the document based on the needs of the user. A user is also given an additional function where he/she can decide the percentage the document should be summarized to. This gives the end user the power to make an informed decision based on the summary percentage set according to their choice.

The rest of the paper is organized as follows. Section 2 gives a gist of the existing work, which is mainly focused on extractive summarization. In Sect. 3, the idea and the logic behind the implementation are discussed. The scope, various assumptions

P. Nalage (✉) · J. Parekh · A. Metha · A. R. Joshi
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: pknalage@gmail.com

J. Parekh
e-mail: jay.parekh7689@gmail.com

A. Metha
e-mail: arjav12352@hotmail.com

A. R. Joshi
e-mail: abhijit.joshi@djsce.ac.in

and constraints related to our system are also mentioned. Section 4 presents the architecture of our system. Section 5 provides the implementation details of extractive and abstractive summarization techniques. Section 6 presents the results obtained and their analysis. The paper ends with the conclusion and future scope.

2 Literature Review

In this section, we discussed the systems currently being available for text summarization, various useful tools for implementation and different approaches followed for text summarization. This section ends with our observations on existing work and major challenges faced in the realization of our system.

2.1 *Literature Related to Existing Systems*

Now let us see few text summarization systems.

2.1.1 Naive Summarizer

It is based on word frequency scoring. This summarizer works in the following manner:

1. Read the input document.
2. Preprocess the document by stripping the extra spaces.
3. Break the document into individual words for tokenization.
4. Count the frequency of each word in the input and rank them in descending order of the score.
5. Pick the first N sentences based on their score.

2.1.2 Wikipedia Summarization

Wikipedia—hosted by Wikimedia, a non-profit organization—is a free online encyclopedia. Wikipedia Summarizer was developed to summarize the pages on Wikipedia. It was developed in a Python framework Django. It uses Google search result count, sentence position and word similarity using WordNet for summarization. It also features a custom summary option where user can give keywords which will be excluded from the summary.

2.2 *Literature Related to Tools*

Now let us see some of the useful tools used in the development of our model.

2.2.1 Gensim

The summarization module present in Gensim uses TextRank, an unsupervised algorithm based on weighted graphs [1]. It builds on Google’s popular PageRank algorithm for ranking web pages. TextRank works as follows [7]:

1. Preprocess the text by deleting stop words.
2. Make a diagram that takes the sentence as a vertex.
3. Connect sentences by the edges. The degree of similarity between the two sentences is determined by the edge weight.
4. Run the PageRank algorithm on the chart.
5. Select the vertex with the highest PageRank score (sentence).

2.2.2 Pyteaser

It is used to generate an extractive summary by ranking the sentences of the article. The top N sentences form the summary [2]. These summaries are created by ranking sentences in a news article according to how relevant they are to the entire text. PyTeaser is the Python implementation of TextTeaser—an automatic summarization algorithm that combines the power of natural language processing and machine learning to produce good results.

The ranking is based on the following criteria:

1. Relevance to the title
2. Relevance to keywords in the article
3. Position of the sentence
4. Length of the sentence.

2.3 *Literature Related to Methodology*

Some of the parameters, which are useful in devising the methodology, are sentence position, term frequency-inverse document frequency, word similarity, etc. Let us see them in brief now.

Table 1 TF-IDF illustration

Word	Frequency value	Score
He	299	0.004
Is	829	0.002
An	460	0.002
Honest	2	0.5
Person	77	0.013
	Total score	0.025
	Final score	0.025/5 = 0.005

2.3.1 Sentence Position

It is used to provide scores to the sentences. Based on the relative position in its section, each sentence is assigned a score. More weight is assigned to the sentences appearing initially of the section as per Eq. 2.3.1.1.

$$\text{Score}(s) = \frac{1}{\text{position in the section}} \quad (2.3.1.1)$$

2.3.2 Term Frequency-Inverse Document Frequency

TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The TF-IDF score for a word present in a sentence is inversely proportional to the number of sentences that contains the same word. Words with high scores imply a strong relationship with the sentence they appear in, i.e., the word might be of interest to a user [3]. Data frequency values are made using 1000 Wikipedia pages which are selected randomly. A high-frequency word in the corpus will have a lower value. The metric for calculating the score of sentences is given in Eq. 2.3.1.2

$$\text{Score}_{f4}(s) = \frac{\sum_{\text{word}_i} \frac{1}{\text{Occurance of the word}_i \text{ in the corpus}}}{\text{No of words contained in the sentence}} \quad (2.3.1.2)$$

where “word” represents the list of words present in the sentence. An example of “He is an honest person.” is shown in Table 1.

2.3.3 Word Similarity

This parameter provides an option for the user to exclude certain words from the summary. Then, the system compares the word similarity with every word in the sentence with the help of the WordNet dataset. The summarizer then assigns scores to

Table 2 Word similarity

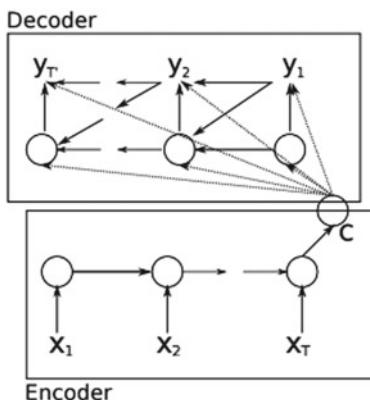
Word 1	Word 2	Score	Explanation
Cat	Tiger	0.5	Both belong to <i>Felidae</i> family
Apple	Tiger	0.1	Less related
Apple	Orange	0.25	Both are fruits
Cat	Cat	1.0	Same words
Cat	Feline	0.5	Synonyms
History	Past	0.5	Both referring to lime

the sentences according to the user preference. It increments the score for sentences containing similar words for positive keywords, while for negative keywords, it decrements the score for sentences containing similar keywords. Word similarity examples are shown in Table 2.

2.3.4 Sequence 2 Sequence

It is a popular deep learning method in the field of artificial intelligence. In this case, the RNN is trained to map an input sequence to an output sequence which may not be of the same length. Figure 1 shows the encoder-decoder architecture in which the encoder RNN reads the input sequence and generates the fixed-size context vector. This fixed-size context vector is then given as input to the decoder RNN. One limitation of this architecture is that it is difficult to summarize a long sequence with a context vector that has a small dimension.

Fig. 1 Seq-2-seq model
(source <https://www.analyticsvidhya.com/blog/2018/03/essentials-of-deep-learning-sequence-to-sequence-modelling-with-attention-part-i/>)



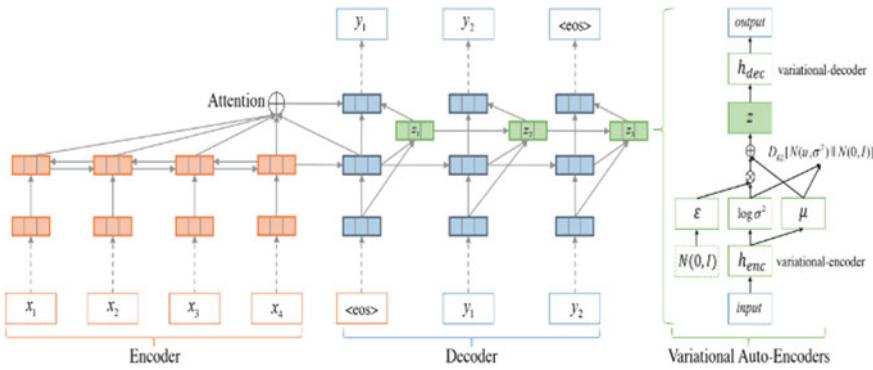


Fig. 2 Deep recurrent generative decoder seq-2-seq model (*source* <https://www.slideshare.net/tomo1414/deep-recurrent-generative-decoder-for-abstractive-text-summarization>)

2.3.5 Deep Recurrent Generative Decoder

It is a new framework based on the popular sequence-to-sequence oriented encoder-decoder model as shown in Fig. 2. It is equipped with a latent structure modeling component. Abstractive summaries are generated based on both the latent variables and the deterministic states [4]. Variational Auto-Encoders (VAEs) show strong capability in modeling latent random variables. It also improves the performance of tasks in various fields like sentence generation. Hence, VAEs are used in the decoder component of the model.

2.4 Observations on Existing Systems

Today's popular and versatile methods that are currently used in summarization are implemented using extractive methods: The vital step is selecting the top N sentences by assigning a score based on some algorithm.

1. Lack of readability is a major drawback of extractive methods. In most of the systems, sentence scoring is based on simple heuristics (e.g., frequency of the sentences) that are not sufficient to produce a coherent text. Current researches aim at exploring new techniques to produce a more coherent text.
2. Another drawback of the existing text summarization system is duplication. Most of the existing summarization system produces a summary that contains duplicate sentences.
3. Sometimes the summaries produced by the abstractive methodologies contain a lot of grammatical errors or are often absurd.
4. Also, the existing systems do not consider the synonymy and polysemy factor.
5. The evaluation of the summaries is still a major issue. The latest techniques are more consistent over the human-based evaluation. However, a lack of consensus

between humans is the major caveat is when evaluating the summaries. The development of more focused summaries will make evaluation more consistent, and it may achieve better convergence between human and automatic summary evaluation methods.

6. Most of the implemented abstractive methods are used to generate titles for the input document. It is very lucrative for generating newspaper headlines and online blog titles, but it does not generate long summaries.
7. Abstractive methods require preprocessing of the input document such as removing stop words, punctuation and numbers. Without preprocessing, the machine would not be able to interpret the input document.
8. Most of the abstractive methods are efficient on Linux operating systems rather than Windows or Macintosh.
9. ROUGE-N scores are used to measure the quality of a summary. G-BLEU scores are used less often but are more fitting for abstractive methods. ROUGE-N scores are more fitting for extractive methods [6].

2.5 *Major Issues and Challenges*

In the realization of our summarization system, the major issues and challenges that need to be tackled are:

1. Long document summarization: The input document could have multiple input sentences, i.e., it could be a long article. These articles could be difficult to summarize as a few key points could be overlooked by the system.
2. High reduction rates: The critical part of a single document is usually five to 30% of the source length. However, compression targets are very small for summarizing from multiple sources. These high reduction rates usually pose a challenge because they are hard to attain without a reasonable amount of domain knowledge.
3. Summary cohesion: There is a high probability that there exists no smooth transition between ideas/concepts/topics in different sentences of the summary. Hence, the summary is likely to make no sense. This can occur due to a range of reasons from lack of domain knowledge to the long input document.
4. Deep linguistic knowledge: In order to maintain the proper language constructs and reformulating the text, the system will require a deep linguistic knowledge of the English words. This knowledge includes all the words of the English language, their meanings, synonyms, antonyms and type (i.e., adjective or verb or noun or pronoun).
5. Non-textual data: The input document might consist of numbers, statistics, graphs, diagrams, etc. The system has to interpret such non-textual information and provide the appropriate summary. For example, economic articles consist of a lot of numbers that are important and relevant to the article.

6. Laboratory testing: Testing the huge dataset requires a large amount of GPU. In order to run the model, a minimum of 6 GB of RAM is required as well.

3 Our Approach

After careful evaluation and review of several approaches, we plan to incorporate features of different methods considering the advantages offered by them and analyzing the suitability of those approaches for implementation of our system.

1. The system provides an interface to the user to upload the documents to be summarized.
2. The system also provides an interface to choose the type of user.
3. The system also provides the feature to choose the compression ratio.
4. Based on the type of user, the system should select the appropriate algorithm to be applied on the input document.
5. The document is preprocessed first to strip the redundant spaces around the words and convert all characters to lower case to maintain consistency. The preprocessing phase also segregates words and sentences by splitting the input document based on spaces and full stops.
6. The system executes the selected algorithm based on step 4.
7. Our system uses a custom metric (a combination of ROUGE-N and G-BLEU) for automatic evaluation of the summary.
8. Based on a certain value (threshold value) of our metric, the text is fed again to the system if it does not meet the threshold value. The threshold score is calculated based on the training dataset.

Now let us see the architecture of our system.

4 Architecture

Figure 3 shows the architecture of our system, which is divided into multiple layers.

Parser: It is the initial layer responsible for parsing the given input document. It helps with data preprocessing as it breaks up the document into smaller chunks for parallel execution which is fed into the summarizer.

Summarizer: The summarizer can be an abstractive or an extractive based on the user's input type. It summarizes the given document with the help of an appropriate algorithm and then evaluates the output summary by using a combination of ROUGE-N and GLEU.

ROUGE-N: After extracting the final summary, the summary is evaluated and its semantic similarity with the original text is examined [3]. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It calculates the number of units that

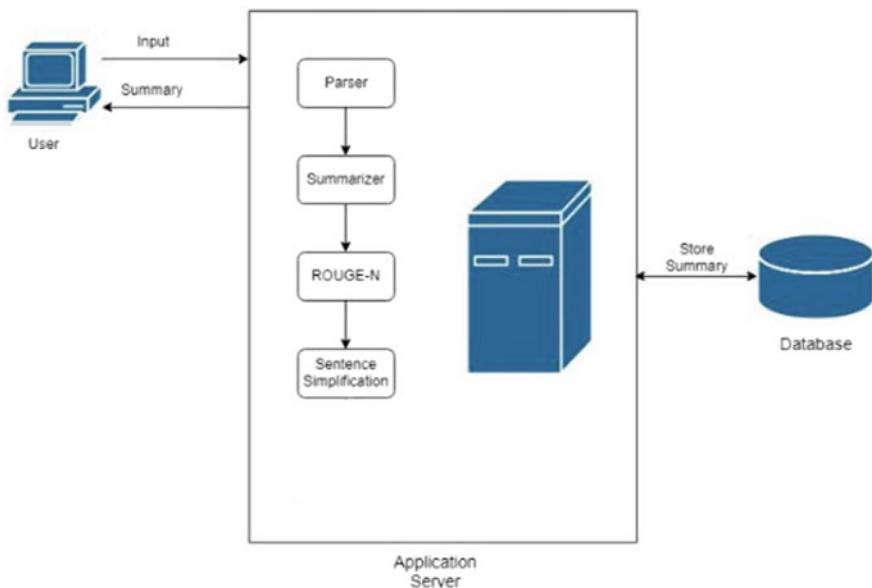


Fig. 3 System architecture

overlap between human and system summary like n-gram, word sequence and pairs of words.

Sentence simplification: It aims to make the sentences easier to read and understand. The main goal of sentence simplification is to reduce the linguistic complexity of the text with the help of DRESS [5], while retaining its original information and meaning [8].

GLEU: It stands for Google bilingual evaluation understudy which is an algorithm for evaluating the quality of the summary. GLEU score correlates quite well with the BLEU metric on a corpus level but does not have its drawbacks for our per sentence reward objective.

Custom metric: In addition to ROUGE-N and GLEU, we have taken the harmonic mean of the two to get the accurate F1-measure to evaluate the summaries so that they contain all the important points. If the custom metric score is less than the threshold score, then the output of the summarizer is fed as an input to the system and the whole procedure is repeated.

User: The user will input the text document and also select the type of user he/she is based on which our system will employ an algorithm to summarize the document and then deliver it to the user. The user shall have the following options as user type:

1. Student
2. Author
3. Teacher
4. Foreign language student.

Database: Different summaries and their respective scores are stored in the database, which aids in calculating the threshold score for our custom metric.

5 Implementation

The process of getting a summarized document from the input document is as follows.

- Step 1 Take a text document as input and convert it to a string.
- Step 2 Convert the whole string to lower case.
- Step 3 Segment the string into sentences and words.
- Step 4 Remove the stop words from the string and convert the words with an apostrophe into words without an apostrophe. For example, the word “*doesn’t*” is converted to “*does not*.”
- Step 5 Based on the type of user selected, execute the algorithm and apply the compression ratio if specified by the user, else use a default compression ratio of 40%.
- Step 6 Store the summary generated from the above step in the database.
- Step 7 Remove the unnecessary tokens from the summary.
- Step 8 Apply the evaluation metric to gauge the quality of the summary. If the score is above the threshold, prompt the user to increase the compression ratio.
- Step 9 Output the summary to the user.

6 Evaluation and Results

The results obtained from our system are presented in this section.

6.1 System Interface

Our system interface is minimal and consists of a drop-down list to select the type of user, a text box to input the summarization ratio and a text box for the input text to summarize (Fig. 4).

Consider some sample text given as an input by the user as shown in Fig. 5.

The preprocessed text is shown in Fig. 6.

On selecting “Student” (extractive—PyTeaser) as the user and 30% as compression ratio, we get the summarized output as shown in Fig. 7.

The various scores obtained for the summarized output (from Fig. 7) are presented in Table 3.

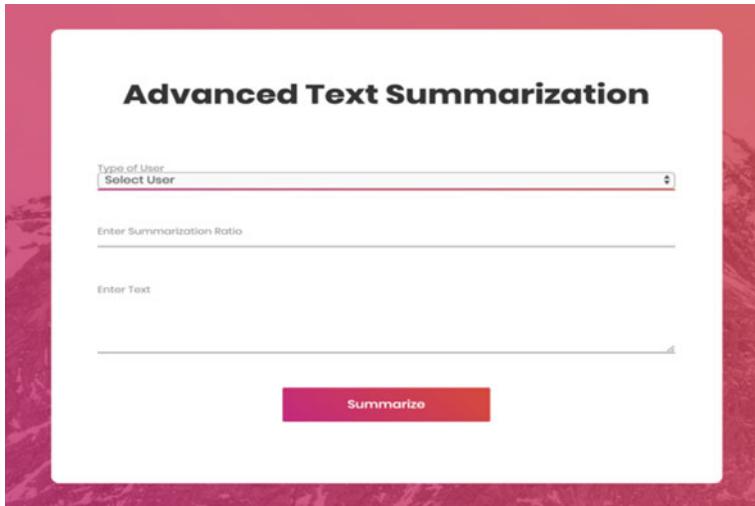


Fig. 4 System UI homepage

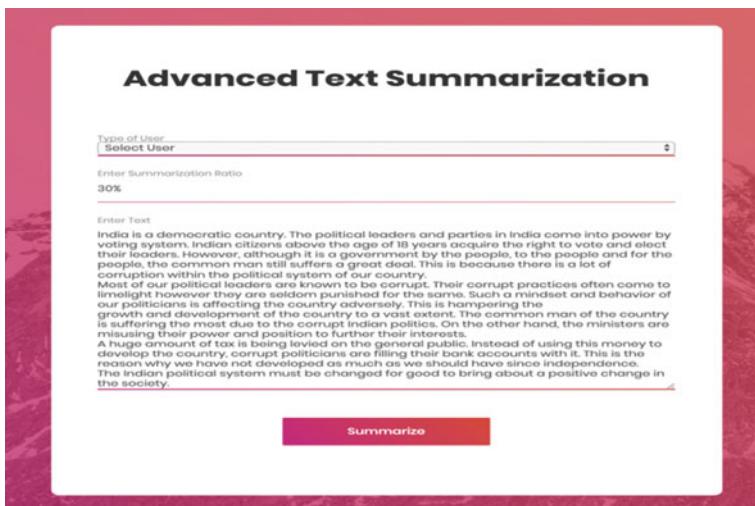


Fig. 5 System UI input

On the same input but setting “Foreign Language Student” (extractive—Gensim) as the user and 30% as compression ratio, we get the summarized output (see Fig. 8).

The various scores obtained for the summarized output (from Fig. 8) are presented in Table 4.

On the same input but setting “Teacher” (abstractive—sequence 2 sequence) as the user, we get the summarized output (see Fig. 9).

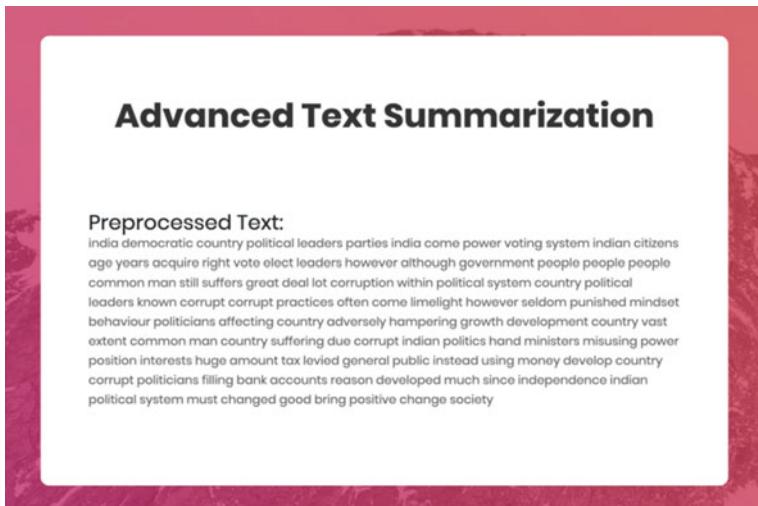


Fig. 6 Preprocessed text

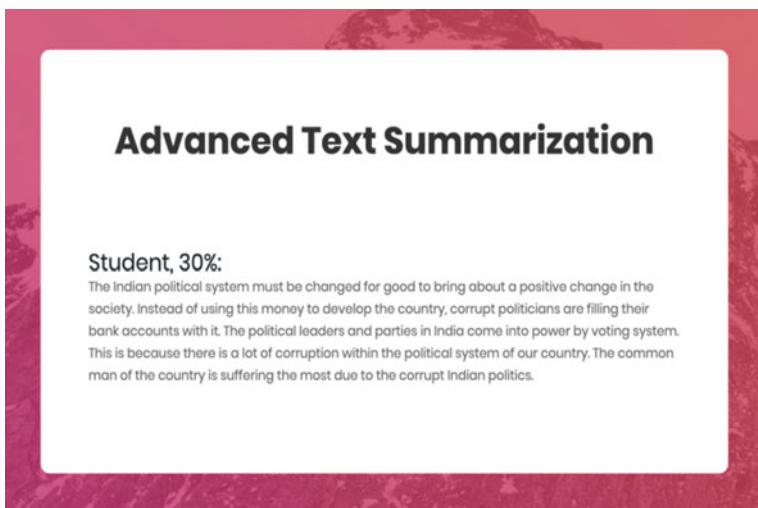


Fig. 7 System UI output for Student at 30% compression ratio

Table 3 Scores of user—student

ROUGE-N score	0.634408597819
GLEU score	0.314465408805
Our metric	0.420497468937



Fig. 8 System UI output for Foreign Language Student at 30% compression ratio

Table 4 Scores of user—foreign language student

ROUGE-N score	0.573033703777
GLEU score	0.34965034965
Our metric	0.434301284698

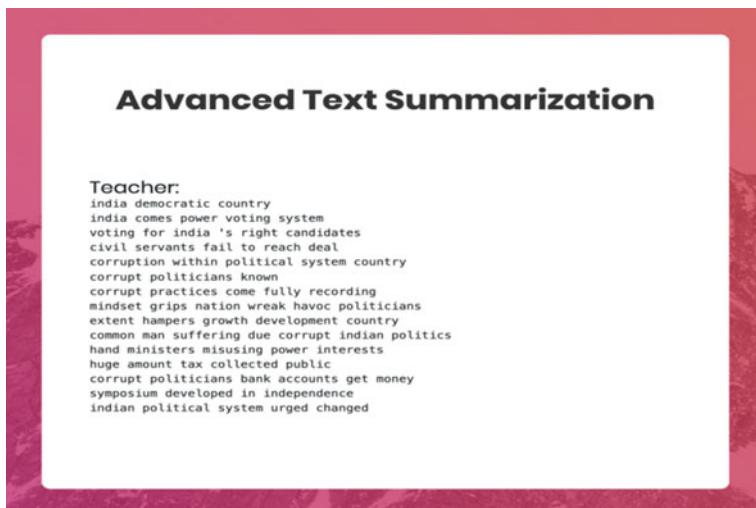


Fig. 9 System UI output

Table 5 Scores of user—teacher

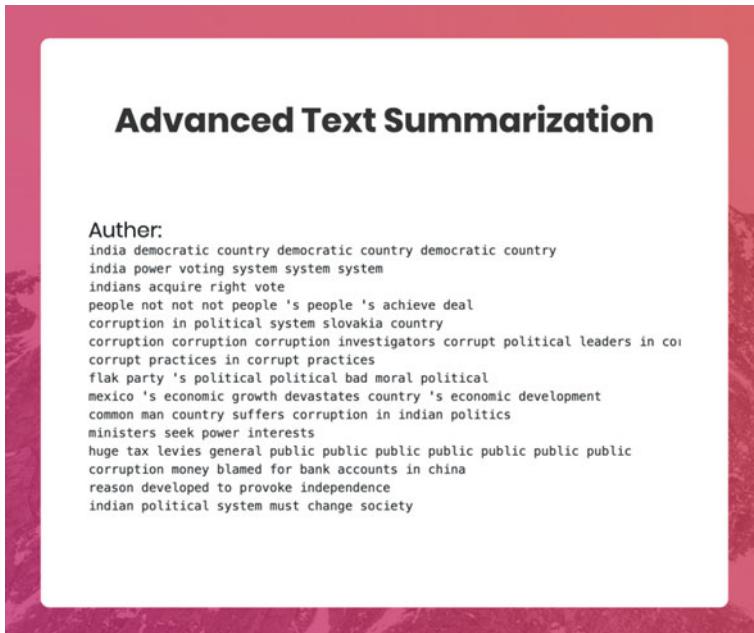
ROUGE-N score	0.461538457082
GLEU score	0.0446030330062
Our metric	0.0813449023169

The various scores obtained for the summarized output (from Fig. 9) are presented in Table 5.

On the same input but setting “Author” (abstractive—deep recurrent generative) as the user, we get the summarized output (see Fig. 10).

The various scores obtained for the summarized output (from Fig. 10) are presented in Table 6.

From the scores obtained for various types of cases (student, foreign language student, teacher and author), we can say that the extractive algorithm for the case (student and foreign language student) achieves a much higher score compared to the abstractive algorithm for the case (teacher and author). Also, the difference between

**Fig. 10** System UI output**Table 6** Scores of user—author

ROUGE-N score	0.464088393358
GLEU score	0.0315855969678
Our metric	0.0591457661129

the abstractive and extractive ROUGE-N score is less significant as compared to the GLEU score.

7 Conclusion and Future Scope

As we have already seen there is a dire need for a system to summarize the tremendous information available. The need is observed in many different target domains in order to reduce the size of the data, ranging from students to teachers to editors of newspapers. The text summarization software would make their work considerably easier and quicker. Traditional systems have a variety of flaws that our system intends to overcome. We have created a system that employs abstractive methods and trained the model in such a way that it has a refined understanding of the text in order to have an unbiased opinion in the summaries that it creates. Our system measures redundancy using ROUGE, GLEU and a combined metric to evaluate the summary and provide the user with the most efficient summary of the input document. Thus, our system will make use of AI to replace the manual task of summarizing documents. The system will deliver a concise and accurate summary with minimal errors with the help of evaluation metrics. Thus, we have built a system that will improve upon and has overcome some of the issues observed in the conventional methods of summarization to a certain extent and helped in achieving progress in this field of research. However, some efforts are required to improve the quality of automatic summaries as the number of topics is proposed in a different context by the users. The immediate application for this system includes integration into a search engine so that a summary can be shown on the search page instead of the document themselves. Summaries can also be integrated into a document visualization tool. Another application can be brought up a summary instead of the first few lines when a mouse has hovered over an article.

References

1. Text summarization techniques—a brief survey
2. PyTeaser—extractive summarization. <https://arxiv.org/pdf/1707.02268.pdf>
3. Python library for the ROUGE metric. <https://github.com/xiaoxu193/PyTeaser>
4. Deep recurrent generative decoder for abstractive text summarization. <https://arxiv.org/pdf/1708.00625.pdf>
5. Semantic modeling of multimodal documents for abstractive summarization. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.2132&rep=rep1&type=pdf>
6. TextRank—bringing order into texts. <https://github.com/pltrdy/rouge>
7. A survey on methods of abstractive text summarization. <https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf>
8. Sentence simplification with deep reinforcement learning. <https://arxiv.org/pdf/1703.10931.pdf>

Chapter 64

Juxtaposing Deep Learning Architectures for Breast Cancer Classification



Purva Raut, Viraj Mehta and Akshen Kadakia

1 Introduction

Breast cancer has been widely acknowledged as one of the most lethal and frequently occurring cancers across the world. Breast cancer has been the most frequently diagnosed cancer in women throughout all regions of the world, except Eastern Africa which was dominated by cervical cancer [1]. Breast cancer is also the second most incident cancer (11.6%) after lung cancer among men and women [1]. The GLOBOCAN 2018 [1] report estimated that there would be over 2 million new cases of breast cancer worldwide and over 600,000 mortalities in the year 2018. From India's perspective, as per an epidemiology of breast cancer in Indian women [2], breast cancer ranks number one with age adjusted rate as high as 25.8 per 100,000 women and mortality 12.7 per 100,000 women. The same study projects the number of breast cancer cases during the time period of 2020 to rise up to 1,797,900. This serves as an indicator of the deadliness of breast cancer and its outgrowth not just in India, but across the world. It is of prime importance that humanity be equipped with measures which can ably fight against breast cancer, thereby saving thousands of lives.

Deep neural networks or deep learning has engendered fascinating ideas and concepts across all possible domains, be it in the financial industry, the automobile industry, in arts, etc. Two of the most popular applications powered by deep learning

P. Raut (✉)
Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, India
e-mail: purvapraut@gmail.com

V. Mehta
JP Morgan Services India Pvt. Ltd., Mumbai 400076, India
e-mail: viraj.mehta30@gmail.com

A. Kadakia
Media.Net Software Services (India) Private Limited, Mumbai 400069, India
e-mail: akshenk8@gmail.com

in the recent years are of YOLO [3] and neural style transfer [4]. Deep learning has also come to the fore in the field of medicine. Deep neural networks have been extensively used in brain magnetic resonance imaging (MRI) [5]. Deep learning has also been used for text analysis in medical records [6]. This study showcases a novel use of deep learning for classifying relations between medical concepts in clinical records which was extensible to applications in other domains as well. Deep learning has also made inroads in tackling the breast cancer issue. There have been many machine learning and deep learning approaches to classify whether a mammogram or a histopathological image is actually benign or malignant. New state-of-the-art deep learning models have shown to be incredibly accurate and precise, and this is why our objective is to apply these models and figure out if these models are powerful enough to successfully classify histopathological images of breast cancer.

The remainder of this study will be organized in the following manner: Sect. 2 consists of a literature study comprising a variety of algorithms used in classification or detection of breast cancer based on different image datasets. Section 3 describes the architecture overview which will provide an overview on the deep learning models that we have decided for our study. Section 4 represents the experimental setup which talks about the dataset and the training of the deep learning models. Section 5 presents the outcome of our study. Lastly, Sect. 6 concludes the study.

2 Literature Review

For the literature review, we will be looking at four research papers which showcase attempts to provide a solution to this classification problem using different techniques on the BreakHis [7] dataset.

The study by Sudharshan et al. [8] uses multiple instance learning (MIL) [9] methods to classify images on the BreakHis [7] public dataset. MIL provides a framework to counter the issue where labeling the data proves to be an expensive process by organizing instances into bags, without needing to label all the instances. The authors have investigated the pertinence of MIL for a computer-aided diagnosis system based on the analysis of histopathological breast cancer images. The BreaKHis [7] public dataset which consists of 7909 microscopic biopsy images of benign and malignant breast tumors has been adopted for the experiment. The results of this study show that the nonparametric methods outperformed the single-level classification methods. The comparison between MIL and single instance (conventional) classification reveals the relevance of the MIL paradigm.

In double transfer learning for breast cancer histopathologic image classification, de Matos et al. [10] propose an approach for classifying histopathologic images (HI) of breast cancer tumors using transfer learning [11] to extract features from HI using a convolutional neural network (CNN) [12] pretrained with ImageNet [13]. They have also used transfer learning [11] to train a support vector machine (SVM) classifier on a tissue labeled colorectal cancer dataset aiming to filter the patches from a breast cancer histopathological image and remove those images which were

irrelevant. This was shown to improve the accuracy for classifying malign and benign tumors on breast cancer images.

In deep features for breast cancer histopathological image classification by Spanhol et al. [14], they propose a classification approach based on DeCAF [11]. They opine that convolutional neural networks (CNN) [12] achieve higher recognition rates for breast cancer recognition over handcrafted feature descriptors, but they are computationally expensive and quite complex in nature. They suggest DeCAF [11], which is based on reusing a previously trained CNN on a different dataset but only as feature vectors, which can then be used as an input for a classifier trained only for the new classification task. They evaluated the DeCAF [11] features for breast cancer recognition and compared it with the other methodologies. The evaluation showed that DeCAF [11] can be a feasible alternative for highly accurate breast cancer recognition models.

In breast cancer histopathological image classification using convolutional neural networks [15], the authors lay down a method which is based on extracting image patches for training the CNN [12] and then combining them for the final classification. The authors also investigated a combination of different CNNs using simple fusion rules. They used the high-resolution images from the BreaKHis [7] dataset as input to an existing CNN. An existing CNN was used to avoid variations of the model which could potentially be more complex and computationally expensive. The outcome of this study was that the CNN performance was much better compared to results generated by other models trained with handcrafted textural descriptors. By using fusion rules to combine different CNNs, they achieved some further improvement in recognition rates.

3 Architecture Overview

For this comparative study, we have decided to take up two of the most famous deep learning models in the current times, the Inception v3 network [16] and the NASNet [17] by Google Brain.

3.1 NASNet

The authors of this architecture proposed a method to learn the model architectures directly on the dataset of interest. They proposed that since this approach would be quite expensive if it was applied directly on a large dataset, it would be more efficient to first search for a building block on a smaller dataset and then transfer that building block to a larger dataset. In their study, they searched for the best convolutional building block on the CIFAR-10 [18] dataset and then transferred that building block to the ImageNet [13] dataset by creating multiple copies of this particular block and then stacking them together. Each such block would have their own

parameters leading to a convolutional architecture, which was named the NASNet architecture and such blocks were termed as “cells.” The search method used for this architecture was the neural search architecture (NAS) [19]. The architectures of all the convnets are predetermined. They comprise repetitive convolutional cells where each convolutional cell has the same architecture but different weights. They used two types of convolutional cells: normal cell and reduction cell. The normal cell was that convolutional cell which would return a feature map of the same dimension, and the reduction cell would return a feature map where the height and width of the feature map are deducted by a factor of two. This facilitated scalable architectures for images of any size. NASNet architecture for ImageNet and CIFAR-10 datasets as shown in Fig. 1 depicts the usage of normal cells and reduction in NASNet architecture. A controller RNN has been utilized to search for the architectures of the reduction cell and the normal cell. Searching for the remainder of the architecture of the convolutional cells is a 5 step procedure as prescribed by the authors of this study which includes selecting the hidden states, selecting the operations on those hidden states and finally selecting a method to combine the hidden states.

Fig. 1 NASNet architecture for ImageNet and CIFAR-10 datasets [17]

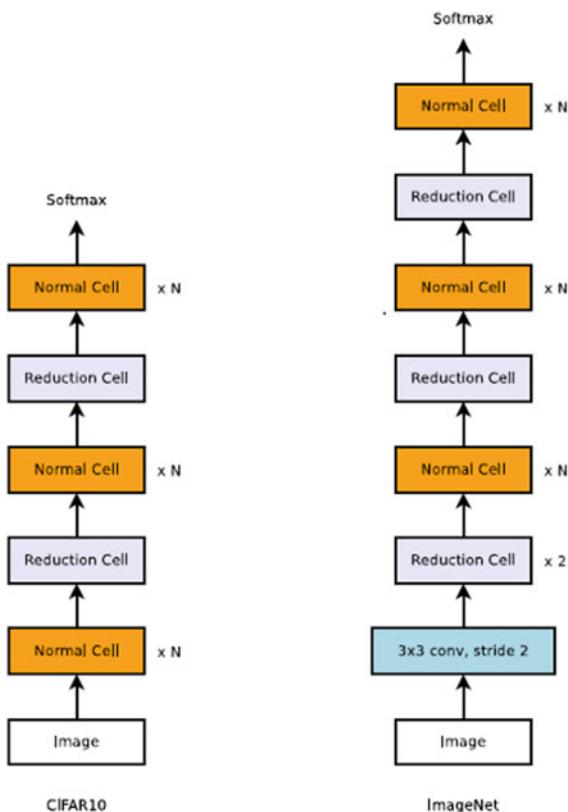


Fig. 2 The Operations space for NASNet cell evaluation [17]

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-separable conv

The potential set of operations that can be selected for the hidden states is shown in Fig. 2. These steps are used for only one cell. The controller RNN is a one-layer LSTM with 100 hidden units at each layer and 10B softmax predictions for the two cells associated with each architectural decision. Each of these 10B predictions is associated with a probability. The product of all probabilities is the joint probability of a child network. This is used for evaluating the controller RNN's gradient. The controller RNN is updated by scaling the validation accuracy of the child network such that low probabilities are assigned for bad child networks and high probabilities are assigned for good child networks.

3.2 Inception v3 Network

The study which proposed Inception v3 [16] has been aptly named so because they were able to achieve computational efficiency and fewer parameters by rethinking the Inception architecture. The Inception v3 architecture is a 42 layer deep architecture comprised of around 7 million parameters. This is the third iteration of the Inception architecture proposed by Szegedy et al. in 2015 which was called GoogLeNet [20]. Later, the Inception architecture was refined in various ways, first by the introduction of batch normalization which came to be known as Inception v2 [16]. The third iteration saw additional factorization ideas which led to the Inception v3 architecture [16]. Some of the salient features of this architecture are described as follows.

Factorizing Convolutions

This was done to reduce the number of parameters without affecting the efficiency. The study suggests factorization into smaller convolutions and factorization into asymmetric convolutions.

Auxiliary Classifiers

The Inception v3 authors argued that the idea of two auxiliary classifiers being used to improve the convergence of deep neural networks is wrong. Their experiments showed that the training progression with or without the auxiliary classifiers looked almost identical. They also tried removing one of the classifiers which proved that it did not adversely affect the quality of the network. They are of the opinion that the auxiliary classifiers are, in fact, regularizers and not catalysts in converging a deep network.

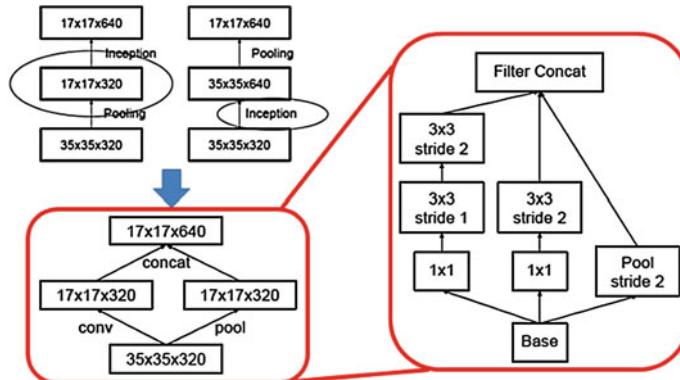


Fig. 3 Grid size reduction as proposed by the authors of the Inception v3 architecture [16]

Grid Size Reduction

Convolutional networks conventionally used some form of pooling to reduce the grid size of the feature maps. To avoid a representational bottleneck as described in the design principles of this study, before applying pooling, the activation dimension of the network filters is expanded by adding a convolutional step before applying max pooling or average pooling. But, this is very expensive computationally. Hence, they suggested a variant as shown in Fig. 3 which removes the computational expense while removing the representational bottlenecks as well.

Model Regularization via Label Smoothing

The authors proposed a mechanism which would regularize the classification layer by estimating the marginalized effect of the label dropout during training. Basically, a regularizing component was added to the loss formula that would ensure that there was no overfitting and also ensure that the model was adaptive. A regularizing component was introduced in the loss calculation so that the network did not become too confident about its own predictions. The Inception v3 architecture can be seen in Fig. 4.

4 Experimental Setup

4.1 The Dataset

We have used the breast cancer histopathological public dataset (BreakHis) [7] for this comparative study. BreakHis is composed of 7909 microscopic images of breast tumor tissue collected from 82 patients using different magnifying factors

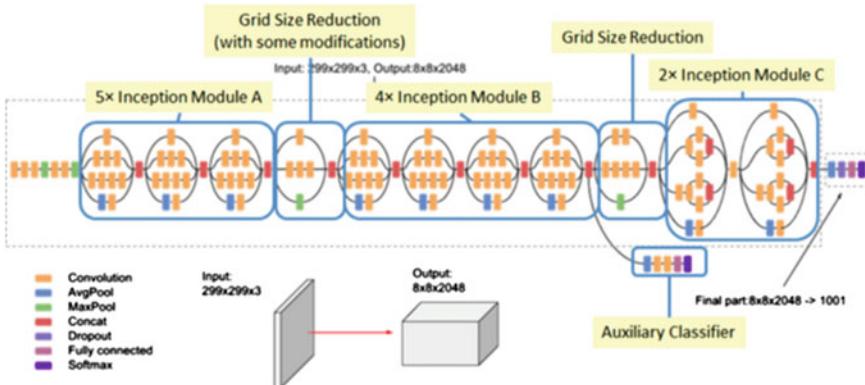


Fig. 4 Inception v3 architecture [16]

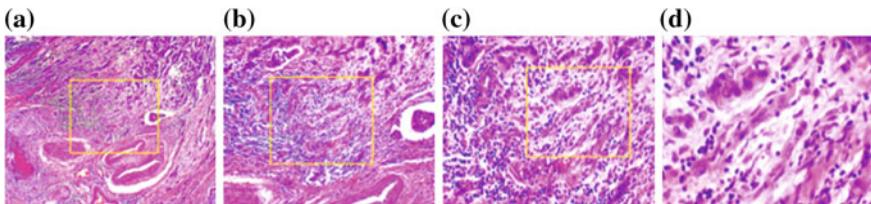


Fig. 5 Slide of breast malignant tumor (stained with HE) seen in different magnification factors: **a** 40 \times , **b** 100 \times , **c** 200 \times , and **d** 400 \times [7]

(40 \times , 100 \times , 200 \times , and 400 \times). To date, it contains 2480 benign and 5429 malignant samples (700×460 pixels, 3-channel RGB, 8-bit depth in each channel, PNG format). The dataset is divided into two main groups: benign and malignant tumors. Histologically, benign is a term referring to a lesion that does not match any criteria of malignancy. Normally, benign tumors are relatively “innocents”; they are slow growing and remain localized. Malignant tumor is a synonym for cancer: Lesion can invade and destroy adjacent structures (locally invasive) and spread to metastasize to cause death. Since the objective of this study is just to determine whether a histopathological image is benign or malignant, we will not be factoring the images based on the magnification levels (Fig. 5).

4.2 Transfer Learning

The Inception v3 [16] and NASNet [17] architectures are ginormous deep neural networks with numerous layers and a huge amount of parameters. Retraining such architectures all over again would not have been computationally feasible for us. That is why we decided on using the concept of transfer learning [11]. Transfer learning is

Table 1 Training-validation split

	Benign	Malignant	Total
Training	1984	4343	6327
Validation	496	1086	1582
Total	2480	5429	7909

a technique that uses a previously trained neural network for datasets like ImageNet [13], CIFAR-10 [18], etc., and trains it again from the weights obtained after training on such datasets. Since Inception v3 and NASNet both have been trained on the ImageNet dataset, transfer learning will help minimize the workload substantially. We can just use the weights of all the layers that were trained on ImageNet as they are. We have introduced a custom final layer on both the architectures. It is this final layer that we will train our BreakHis [7] dataset on. The advantage of this mechanism is that one can define their custom softmax layers to cater to the need of the use case that needs to be dealt with. For this study, we have introduced three layers on top of the existing architectures. We have added an average pooling layer, a dropout layer [21] with drop probability 0.4 to avoid overfitting and a softmax layer which will evaluate whether an image is malignant or benign.

4.3 Implementation

We have used the powerful Keras library for our implementation. Keras is a high-level deep learning library, written on top of Tensorflow [22]. 80–20 split was taken for this dataset. The split is shown in Table 1. The image size for the Inception v3 model is 224×224 whereas image size used for NASNet model is 331×331 . A batch size of 16 was used, and the models were trained for 150 epochs.

5 Results

The graphs of accuracy and loss portray the trends for training and validation after 150 epochs. For training, the graphs show that the accuracy and cross-entropy are more or less the same after the first few epochs itself. The validation accuracy is almost close to 75% for NASNet and over 70% for Inception v3 which is respectable and in sync with the loss components. The training accuracy and loss quickly generalized to their respective values over a few epochs itself. The NASNet training accuracy came around 86% with a loss of around 34%, whereas the Inception v3 accuracy came to be around 81% with a training loss of around 43%.

The confusion matrix shows the precision and recall metrics for this evaluation across 150 epochs. The NASNet model has a better precision, recall and F1-score overall than the Inception v3 model. Figures 6 and 7 show the classification metrics for both the architectures.

Classification Report		precision	recall	f1-score	support
	Benign	0.31	0.09	0.14	2480
	Malignant	0.69	0.91	0.78	5429
	accuracy			0.65	7909
	macro avg	0.50	0.50	0.46	7909
	weighted avg	0.57	0.65	0.58	7909

Fig. 6 NASNet classification metrics

Classification Report		precision	recall	f1-score	support
	Benign	0.29	0.16	0.21	2480
	Malignant	0.68	0.82	0.74	5429
	accuracy			0.61	7909
	macro avg	0.49	0.49	0.48	7909
	weighted avg	0.56	0.61	0.58	7909

Fig. 7 Inception v3 classification metrics

The precision value is higher than the recall value for the “Benign” class which signifies that the models have more false negatives than false positives. One of the reasons for this failure can be the fewer number of labeled images available for that class. This has also lead to drop in the overall accuracy. NASNet has shown better precision and recall values than Inception v3 for both the classes.

6 Conclusion

This study was an attempt to tackle the breast cancer classification problem by comparing two of the most famous and highly accurate deep learning models of recent times. We trained the Inception v3 network [16] and the NASNet [17] on the BreakHis [7] dataset using transfer learning [11] to add our own custom final layer and train just the custom layer while keeping the weights obtained from training the models on the ImageNet data. The objective was to determine which model could best classify whether a tumor was benign or malignant from a histopathological image. The result table and the accuracy graphs show that NASNet is a better classifier on this dataset. With an accuracy of 86.3% and a loss of 33.8%, the NASNet model outdid the Inception v3 network which posted an accuracy of 81% with a loss of 43.2%. The confusion matrix further backs the fact that the NASNet model had better scores for precision and recall than the Inception v3 model. Overall, keeping

in mind the scope of this study with respect to the dataset and the restrictions on our computational capabilities, NASNet architecture has proved to be a better classifier than the Inception v3 network.

References

1. Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A (2018) Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin 68(6):394–424
2. Malvia S, Bagadi S, Dubey U, Saxena S (2017) Epidemiology of breast cancer in Indian women: breast cancer epidemiology. Asia-Pacific J Clin Oncol 13. <https://doi.org/10.1111/ajco.12661>
3. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: CVPR
4. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: CVPR
5. He B, Guan Y, Dai R (2019) Classifying medical relations in clinical text via convolutional neural networks. Artif Intell Med 93:43–49
6. Bernal J, Kushibar K, Asfaw DS, Valverde S, Oliver A, Martí R, Lladó X (2017) Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. CoRR. [arXiv:abs/1712.03747](https://arxiv.org/abs/1712.03747)
7. Spanhol F, Oliveira LS, Petitjean C, Heutte L (2016) A dataset for breast cancer histopathological image classification. IEEE Trans Biomed Eng TBME 63(7):1455–1462
8. Sudharshan PJ, Petitjean C, Spanhol F, Oliveira LS, Heutte L, Honeine P (2019) Multiple instance learning for histopathological breast cancer image classification. Expert Syst Appl 117:103–111
9. Dietterich TG, Lathrop RH, Lozano-Perez T (1998) Solving the multiple instance problem with axis-parallel rectangles. In: Artificial intelligence
10. de Matos J, de Britto AS, Oliveira LES, Koerich AL (2019) Double transfer learning for breast cancer histopathologic image classification. In: International joint conference on neural networks (IJCNN), IEEE, pp 1–6
11. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2013) Decaf: a deep convolutional activation feature for generic visual recognition. CoRR. [arXiv:abs/1310.1531](https://arxiv.org/abs/1310.1531)
12. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. In: Neural computation
13. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: IEEE Conference on computer vision and pattern recognition 2009, CVPR 2009, pp 248–255, June 2009
14. Spanhol FA, Oliveira LS, Petitjean C, Heutte L (2016) Breast cancer histopathological image classification using convolutional neural networks 2016. In: International joint conference on neural networks (IJCNN), pp 2560–2567. <https://doi.org/10.1109/ijcnn.2016.7727519>
15. Spanhol FA, Oliveira LS, Cavalin PR, Petitjean C, Heutte L (2017) Deep features for breast cancer histopathological image classification. In: Proceedings of IEEE international conference on systems, man, and cybernetics (SMC), Oct 2017
16. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the inception architecture for computer vision. [arXiv:1512.00567](https://arxiv.org/abs/1512.00567)
17. Zoph B, Vasudevan V, Shlens J, Le QV (2017) Learning transferable architectures for scalable image recognition. [arXiv:1707.07012](https://arxiv.org/abs/1707.07012)
18. Krizhevsky A (2009) Learning multiple layers of features from tiny images. In: Tech report
19. Zoph B, Le QV (2017) Neural architecture search with reinforcement learning. In: International conference on learning representations

20. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
21. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
22. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z et al (2016) TensorFlow: large-scale machine learning on heterogeneous distributed systems. [arXiv:1603.04467](https://arxiv.org/abs/1603.04467)

Chapter 65

A Novel Design for Voice-Enabled Home Automation and Personalized Recommendation System



Harsh Parmar, Narendra Shekokar and Pranjali Thakre

1 Introduction

Over the past decades, technology has changed the way we communicate with our devices. The interaction mode that started with pressing buttons, in course of time, changes to touch screen and now voice-enabled request–response is becoming the most preferred mode of interfacing with our gadgets and devices. When the concept of home automation first came into limelight, no one predicted that in a few years the usage of Internet and extremely portable devices would grow at such an exponential rate. Hence, it would only be appropriate to not limit a home assistant to just controlling electrical appliances. The aim of this paper is to present a device that would control home appliances, help the user to perform other day-to-day activities and give the recommendation to the user. The user can activate the device by using a keyword VARAH. Once activated, the user can give a voice command to switch off/on different applications at home. Users can also ask the device to search for anything on the Web, and the device will return the results as a voice response. Users can also ask the device to give recommendations for something; the device then recommends items to the users based on the user's search history.

H. Parmar

Bachelor of Computer Engineering, D. J. Sanghvi College of Engineering, Mumbai University, Mumbai, Maharashtra 400056, India
e-mail: parmarharsh4296@gmail.com

N. Shekokar (✉)

Professor, Computers Department, D. J. Sanghvi College of Engineering, Mumbai University, Mumbai, Maharashtra 400056, India
e-mail: narendra.shekokar@djsce.ac.in

P. Thakre

Assistant Professor, Computers Department, SIES Graduate School of Technology, Mumbai University, Navi Mumbai, Maharashtra 400706, India
e-mail: Pranjali.Thakre@siesgst.ac.in

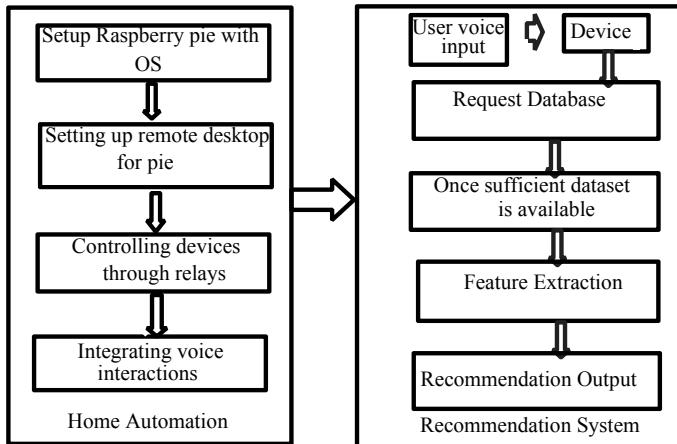


Fig. 1 System design

2 Proposed System Architecture

As shown in Fig. 1, the system includes two modules based on the flow of implementation. The home automation module involves the use of embedded systems that involves Raspberry Pie and Relays and program which allows voice requests–responses for the user. The other module is a personalized recommendation. There are primarily two approaches that can be followed for building a recommendation system: 1. collaborative filtering 2. content-based recommendation. Collaborative filtering analyzes current user's past preferences and based on that finds other similar users to perform a comprehensive prediction [1, 2]. A content-based recommendation algorithm performs analysis in only the current user's past preferences to give a new suggestion [1, 3]. Our proposed system aims to provide the user personalized recommendations, so a content-based filtering approach will be suitable for our problem statement.

3 Module I—Voice Controlled Home Automation

One of the important features of this device is that it is a standalone device. It means that all the user has to do is to switch on the device, and he can start using the device. There is no ‘app’ that the user has to run to use the device capabilities. For this purpose, a program is running in the background, as a thread, so that when the system boots, the program will automatically start running, and it will start listening for user’s commands. It will continue listening until the user explicitly asks it to stop running. This makes it a full-fledged voice-controlled device. The first phase of the

project is to develop a program that will run on the Raspberry Pi for processing voice requests from users.

3.1 Raspberry Pi and Relays Circuits

The Raspberry Pi is a series of small single-board computers. Raspberry Pi-3 Model B has on-boarded Wi-Fi and Bluetooth capabilities, ARM-compatible CPU which ranges from 700 MHz to 1.2 GHz. Also, it consists of on-board memory which ranges from 256 MB to 1 GB RAM. Raspbian is the operating system used in Pi. It consists of over 35,000 programs and is easy to install in Raspberry Pie [4]. This feature enables the device to connect to the Internet for sending e-mails, web searches and processing user's voice requests to control home appliances. Raspberry Pi consists of 40 pins, general input and output pins (GPIOs). These pins can accept I/O commands and further used to control relays [5, 6]. In our proposed system, Relays basically work as electrical switches capable to drive appliances requiring high voltage supply. Relay circuits in our proposed system will be to connect the appliances to the general input and output pins of the Raspberry Pi. Raspberry Pi supplies a max voltage of 5 V. This voltage is not enough to power most of our electrical appliances such as lights and fans. Hence, to interface such high voltage devices to the Raspberry Pi, Relays are used. Suitable current needs to be passed through the Relay's coils for it to operate. Their design is such that it can operate from 5 to 12 V [7]. A relay consists of an iron core wrapped around by coil of wire and a switch. Without any current, the switch is opened and the circuit is in the disconnected state [5] (Fig. 2).

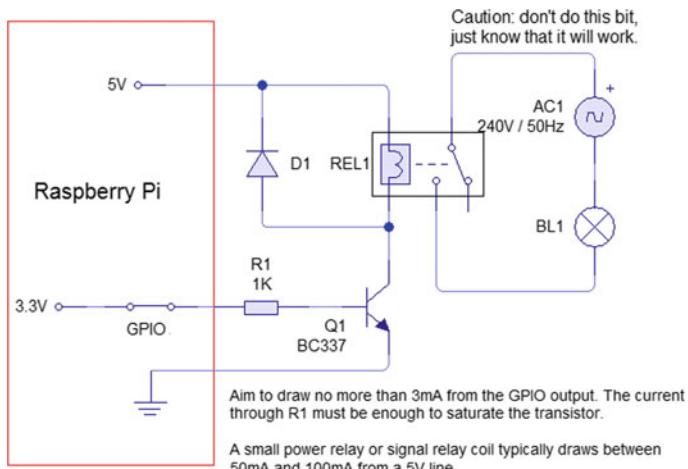


Fig. 2 Relay circuit

3.2 Voice Integration and Web Search

Voice input to the device is given using the microphone. Microphone takes voice command and sends it to speech-to-text conversion process, which converts the voice command into text, and the text is further parsed and search for keywords. The device has a system of keywords against which it tries to match the parsed text. Once the match is found, the device performs that particular task and gives the relevant output. For example, if the user voice request is “Switch On the Fan”—based on the keywords “Switch On” and “Fan”—device will interpret it as a task to turn on fan and generate a text response like “Fan switched on.” This is a text output, and it is then given to text-to-speech converter to get a voice response. Raspberry Pi has an audio jack through which voice output is given to the user. Similarly, this device can perform a web search of any query the user has inputted. To activate the web search, the user command should have the keyword, Web Search. For example, if the user gives a command, “Perform web search for best thriller movies,” All keywords after we search will be treated as a part web search request. Since Raspberry Pi provides Wi-Fi connectivity, the device then searches on Google for best thriller movies. The device returns voice response for the top five results. In case, the device is not able to find any response on the Web for the given request, the failure voice response is “Device could not find any web result.” [8].

4 Module II—Recommendation System

The concept of content-based recommendation is mainly the comparison of the objects being recommended. For instance, consider the recommendation of similar kinds of objects like games or movies. If the past search history of a user shows a list of science fiction movies, then a similar science-based movie is recommended. In the proposed system, the content-based recommendation algorithm: term frequency and inverse document frequency (TF-IDF) is used. TF-IDF algorithm determines the relative frequency of a word in the complete document and compares it to the inversions proportion in the collection of documents. Term frequency is the measure of how many times a term is occurring in a document. For instance, the term “lion” can occur 1000 times in animal study documentation. To determine what frequency count is enough to be categorized that word as frequent. In the previous example, word lion occurring 1000 times in a document of 50,000 words will be termed as less frequent than it occurring the same number of times in a document of 2000 words. Hence, in order to determine the relative word count, we take the ratio of frequency of word to document’s total word count as below [9, 10].

$$\text{TF score} = \frac{\text{Word Frequency}}{\text{Total Word Count}}$$

We should also take into account the fact that words like propositions can also have a high frequency in a document. Hence, it is important to give relevance weight to the words. Like in our example, the word “Lion” will have more weight than the word “the,” Inverse document frequency score describes how relevant that word is in the set of documents. Below equation can be used to determine the IDF score:

$$\text{IDF score} = \left(\frac{\text{Total number of documents}}{\text{Number of documents where the words occur}} \right)$$

TF-IDF score is the product of the TF score and IDF score. After calculating TF-IDF scores, vector space model is used to determine the similarity between items. In this model, the query and the documents are considered vectors as Q and D , respectively. TF-IDF scores are calculated for all the terms in these vectors and represented as $W(d, j)$ where j is a term and d can represent a document or a query. The angles between these vectors show the similarity between the vectors [11, 12].

$$\cos \theta = \frac{Q \cdot D}{|Q| \times |D|}$$

$$\cos \theta = \frac{\sum_{j=\text{number of terms}} W(\text{query}, j) \times W(\text{document}, j)}{\sqrt{\sum_{j=\text{number of terms}} W(\text{query}, j)^2} \times \sqrt{\sum_{j=\text{number of terms}} W(\text{document}, j)^2}}$$

5 Experimentation and Results

For our project, we connected Raspberry Pi-3 Model B to relays to generate a circuit that enabled us to interface with applications like LEDs and table fan that we used for testing. For voice input, we used a microphone that was connected to Raspberry Pi through its 3.5 mm audio jack. Once the voice input is given, a python program running in background would convert it into text using speech recognition API. We used Putty for SSH into Raspberry Pi and VNC for remote desktop GUI. Based on the request, the program will execute the command and return a voice response to the user. The program identifies the types of requests through specific keywords. When we gave an input request as “Turn on fan,” then the system recognizes it as a request to switch on the fan and the fan was turned on, and we got a voice response “Fan turned on.” For output, we used the speakers that are connected to Raspberry Pi’s audio jack. In case of any web search request, the Raspberry Pi-3 Model B has on-boarded Wi-Fi which we connected to our mobile hotspot. So, on request to “web search for doctor strange,” a Google search was successfully performed, the python program summarized the Web response and gave that response as voice output through the speakers connected. Below are snapshots of the project result:

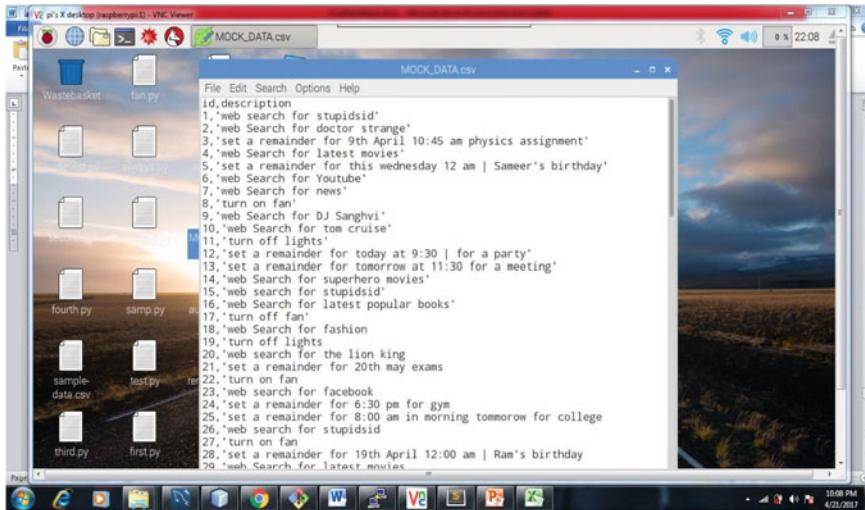


Fig. 3 History of user voice requests

- Figure 3 shows lists of user voice requests. For example, when the user gives input “web search for doctor strange.” The program will first convert the request into text and then do a web search to provide a summarized response.
- Figure 4 shows the summarized response for “doctor strange,” and the program then converts this response to voice output.

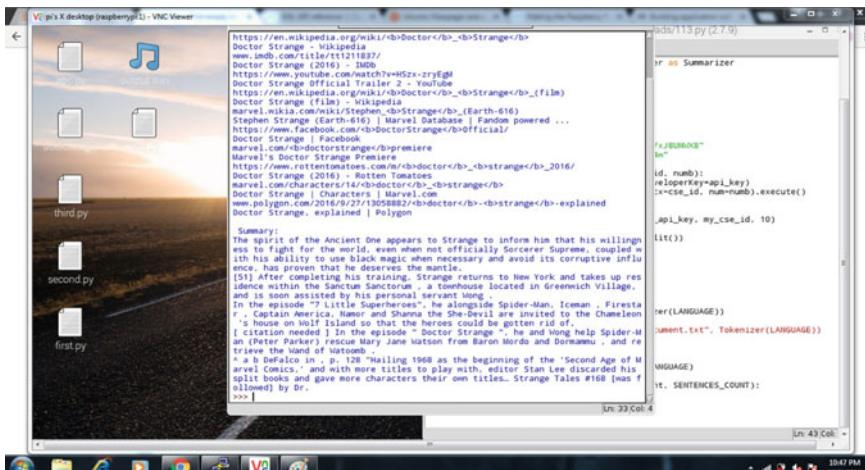


Fig. 4 Summary response for user’s web search request

6 Conclusion

The appliance control through Raspberry Pi has been achieved through the use of relays. A voice response is generated once the user's command is executed. This is a standalone device activated by a special user-defined keyword which can be used to control home appliances like bulbs or fans. The recommendation system based on the user's preferences has been implemented along with helpful modules such as web search. Further, a security mechanism can be added to authenticate the user while performing tasks linked to user accounts, such as sending e-mails. Devices can also be made more receptive by adding user interactions more natural by adding responses according to the person's emotional state; this can be identified by sentiment analysis. For example, Chatbot.

References

1. Lian Z, Hui C (2014) Research on recommendation algorithms in web of things system. In: Proceeding of 7th international conference on intelligent computation technology and automation, Changsha, pp 569–572
2. Chen R, Hua Q, Chang Y, Wang B, Zhang L, Kong X (2018) A survey of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks. Proc IEEE Access 6:64301–64320
3. Lops P, Jannach D, Musto C, Bogers T, Koolen M (2019) Trends in content-based recommendation. Proc User Model User-Adap Interact 29(2):239
4. Chandna M, Tyagi N, Singh R, Rehalia A (2015) Case study: RASPBERRIES Pi B+. Proc Int J Adv Res Comput Sci Softw Eng 5(8). ISSN: 2277 128X, Aug 2015
5. Nehete RO, Bhide AS (2015) Raspberry PI 3 based control and monitor remote machine automation. Proc Int J Eng Comput Sci 6(1): 20151–20155. ISSN: 2319-7242, Jan 2017; Index Copernicus Value 58(10). <https://doi.org/10.18535/ijecs/v6i1.54>
6. Shroff N, Kauthale P, Dhanapune A, Patil SN (2017) IOT based home automation using raspberry pi-3. Proc Int Res J Eng Technol 04(05). e-ISSN: 2395-0056
7. Yadav D, Singh Y, Gupta H (2018) Controlling of relay using raspberry pi via internet for home automation. Proc Int J Adv Res Eng Technol IJARET 9(1):1–11
8. Kaur1 S, Sharma S, Jain U, Raj A (2016) Voice command system using raspberry pi. Proc Adv Comput Intell Int JACII 3(3)
9. Qaiser S, Ali R Text mining: use of TF-IDF to examine the relevance of words to documents. Proc Int J Comput Appl 181. <https://doi.org/10.5120/ijca2018917395>
10. Dai W (2018) Improvement and implementation of feature weighting algorithm TF-IDF in Text classification. In: Proceeding of 2018 international conference on network, communication, computer engineering (NCCE 2018). <https://doi.org/10.2991/ncce-18.2018.94>
11. Singh JN, Dwivedi SK (2012) Analysis of vector space model in information retrieval. IJCA Proc Natl Conf Commun Technol Impact Next Gener Comput CTNGC 2:14–18
12. Gunawan D, Sembiring CA, Budiman MA (2017) The implementation of cosine similarity to calculate text relevance between two documents. In: Proceeding of 2nd international conference on computing and applied informatics 2017. <https://doi.org/10.1088/1742-6596/978/1/012120>

Author Index

A

- Agarwal, Anuja, 423
Agarwal, Yash, 303
Agrawal, Shreyyaa, 657
Ahuja, Bhavesh, 579
Ambarkar, Smita Sanjay, 199
Amin, Prithvi, 1
Arun Kumar, P., 11
Arya, Harshita, 11

B

- Baikerikar, Janhavi, 303
Bakar, Zuriana Abu, 311
Banginwar, Aniket Milind, 117
Barve, Amit, 31
Bhadane, Chetashri, 479, 573
Bhagat, Kalpita, 117
Bhangale, Chandan, 579
Bhat, Sanika, 303
Bhatt, Mittal, 107
Bhavsar, Prerna, 325
Bhoir, Amit, 1
Bhowmick, Kiran, 471
Bide, Pramod, 391, 441
Borhade, Vipul, 555

C

- Chatterjee, Madhumita, 453
Chaturvedi, Deepshikha, 117
Chauhan, Jay, 281
Chavan, Pallavi, 293
Chavan, Suyash, 441
Chitroda, Bhumi, 609
Coutinho, Austin, 579

D

- Dabre, Necil, 237
Dahiya, Vishal, 107
Dakhane, Dhananjay M., 489
Dakshayani, R., 555
Dalvi, Harshal, 465, 609
Dalvi, Prachi, 337
Dama, Sneha, 543
Dani, Apurva, 513
Deshmukh, Jyoti, 399
Deulkar, Khushali, 497
Diwan, Monil, 523
Dixit, Aditya, 609
D'Mello, Lynette, 325, 379
Doshi, Karan, 523
Doshi, Moxa, 523
D'Silva, Godson, 1

G

- Gandhi, Jimit, 411
Gangwal, Anushka, 75
Gawade, Aruna, 67, 97
Gawane, Sarang, 591
Gawde, Aruna, 75
George, Elizabeth L., 633
Ghai, Rishi, 543
Ghosh, Siddhartha, 399
Giri, Nupur, 579
Godbole, Milind, 423
Gokhale, Sanket, 85
Gupta, Ayush, 127
Gupte, Shreeshail, 399
Gurbani, Grishma, 85

H

Hasamnis, Akash, 85

I

Ingle, Shubham, 399

J

Jadhav, Ashish, 293

Jadhav, Vipul, 1

Jain, Aman, 609

Jain, Harsh, 657

Jain, Purvil, 411

Jangid, Mukul, 399

Jani, Megh, 1

Jeswani, Aditya, 379

Jethani, Vimla, 31

Jhunjhunwala, Parth, 325

Johnson, Janice, 379

Joseph, Richard, 85

Joshi, Abhijit, 621

Joshi, Abhijit R., 645, 663

Joshi, Ashwini M., 247

Juwatkar, Saloni, 303

Mehta, Aditya, 237

Mehta, Dhyey, 75

Mehta, Krisha, 57

Mehta, Krupalu, 533

Mehta, Viraj, 679

Metha, Arjav, 663

Mhapsekar, Mandar, 137

Mhapsekar, Prathamesh, 137

Mhatre, Aniket, 137

Mishra, Shubham, 211

Modi, Ashish, 633

Mohemad, Rosmayathi, 311

Mukherjee, Anirudh, 269

Mulchandani, Pratik, 269

N

Nagda, Keval, 269

Nalage, Pratik, 663

Nanaware, Shreyas, 117

Narawade, Vaibhav E., 489

Narvekar, Meera, 187

Natu, Nihit, 127

Nayak, Aparna, 555

Nisar, Purav, 177

Noor, Noor Maizura Mohamad, 311

K

Kadakia, Akshen, 679

Kalbande, D. R., 337

Kamble, Prajakta, 97

Kansara, Dhvani, 365, 621

Karani, Ruhina, 57

Karnade, Aarti M., 337

Kasbekar, Ameya, 645

Katre, Neha, 281

Kavathekar, Vaishali, 303

Kawade, Sarthak, 441

Khandelwal, Harshita, 657

Khan, Saiqa, 187

Khedekar, Soham, 75

Kirtikar, Rishil, 85

Kotadia, Yash, 57

Kurup, Lakshmi, 269, 411

Kyada, Minal, 255

P

Pandita, Anish, 45

Parekh, Jay, 663

Parekh, Mit, 45

Parekh, Niket, 391

Parmar, Harsh, 691

Parmar, Keyur, 345

Patel, Keval, 465

Patel, Naeem, 563

Patil, Bhaktij, 391

Patil, Sonali, 223

Pawade, Dipti, 533, 543

Polly, Christine, 303

Poly, Freddy, 563

Prabhune, Sameer S., 247

Prasad, Amandeep, 591

Puro, Jai, 441

M

Mahadik, Viraj, 127

Mahalle, Parikshit N., 21

Mamtora, Hemal, 391

Mandalya, Kunal, 391

Mangrulkar, Ramchandra, 45, 255

Manjrekar, Mihir, 57

R

Rana, Rashmi, 645

Rathod, Nilesh, 157

Rathore, Kishan, 165

Raut, Purva, 513, 523, 679

Rodrigues, Alban, 465

S

- Sachwani, Jitendra, 45
Sadiwala, Adit, 165
Sainger, Aastha, 31
Sakhapara, Avani, 533, 543
Sankhe, Chinmay, 579
Saraf, Vaibhav, 293
Save, Ashwini, 353
Sawant, Vinaya, 137, 365, 657
Shah, Aditya, 497
Shahane, Sneha, 497
Shah, Devansh, 177
Shah, Dhruvin, 497
Shah, Harsh, 165, 497
Shah, Maulik, 471
Shah, Milind, 269
Shah, Parth, 471
Shah, Raj, 471, 573
Shah, Romit, 45
Shah, Sambhav, 149
Shah, Vidhi, 645
Shah, Yash, 165
Shastry, Nishanth, 211
Shekokar, Narendra M., 199, 353, 691
Shettigar, Shaurya, 573
Shinde, Aditya, 621
Shinde, Gitanjali R., 21
Shinde, Shantanu, 563
Siddiqui, M. Umair, 479
Singh, Gaurav, 591
Singh, Riya, 255
Singh, Shruti, 11
Singh, Vijay Pratap, 479
Sisodia, Jignesh, 237
Sonawane, Pankaj, 177
Soni, Siddhant, 479
Srivastava, Kriti, 149, 165

Suba, Yashi, 621

- Sujith, Shruthi, 543
Surve, Vivek, 533
Swamy, Jalla Manikanta, 345

T

- Taware, Rutuja, 503
Teli, Mrunal, 255
Tendulkar, Soham, 465
Thakar, Mahima, 573
Thaker, Harshi, 573
Thakre, Pranjali, 691
Tipare, Pradnya, 31
Tirodkar, Vivek, 223
Tiwari, Tarun, 211
Tripathy, Amiya Kumar, 127
Trivedi, Kishan, 149

V

- Ved, Tejas, 281
Vidanage, Kaneeka, 311
Vinchhi, Rushabh, 67
Virkud, Sarvesh, 237
Vyawahare, Madhura, 453

W

- Wagaskar, Kalpita, 591
Wagharkar, Samidha, 31
Wankhade, Sunil B., 157

Y

- Yadav, Rishikesh, 31