

Algorithms for Intelligent Systems

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

Subhash Bhalla · Peter Kwan ·
Mangesh Bedekar · Rashmi Phalnikar ·
Sumedha Sirsikar *Editors*

Proceeding of International Conference on Computational Science and Applications

ICCSA 2019



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>

Subhash Bhalla · Peter Kwan · Mangesh Bedekar ·
Rashmi Phalnikar · Sumedha Sirsikar
Editors

Proceeding of International Conference on Computational Science and Applications

ICCSA 2019



Springer

Editors

Subhash Bhalla
University of Aizu
Aizu-Wakamatsu, Fukushima, Japan

Mangesh Bedekar
School of Computer Engineering
and Technology
MIT World Peace University
Pune, Maharashtra, India

Sumedha Sirsikar
School of Computer Engineering
and Technology
MIT World Peace University
Pune, Maharashtra, India

Peter Kwan
Hong Kong College of Engineering
Yau Ma Tei, Hong Kong

Rashmi Phalnikar
School of Computer Engineering
and Technology
MIT World Peace University
Pune, Maharashtra, India

ISSN 2524-7565

Algorithms for Intelligent Systems

ISBN 978-981-15-0789-2

<https://doi.org/10.1007/978-981-15-0790-8>

ISSN 2524-7573 (electronic)

ISBN 978-981-15-0790-8 (eBook)

© 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, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Organizing Committee

Chief Patrons

Prof. Dr. Vishwanath Karad
Prof. Dr. Rahul Karad

Patrons

Dr. Raghunath Mashelkar
Dr. Vijay Bhatkar
Dr. S. Parasuraman

Organizing Chairs

Dr. Shrihari Honwad
Dr. Lalitkumar Kshirsagar
Dr. Anil Hiwale
Dr. Prasad Khandekar

Organizing Co-chair

Dr. Mangesh Bedekar

International Advisory Committee

Dr. Subhash Bhalla, Japan
Ir. Dr. Peter Kwan, China
Dr. Murli Vishwanathan, AUS
Dr. Andrew Stranieri, AUS
Dr. Xin-Wen Wu, AUS
Dr. Jay Gore, USA
Dr. Suresh Borkar, USA
Dr. Maode Ma, Singapore

TPC Chair

Dr. Rashmi Phalnikar

TPC Co-chair

Prof. Sumedha Sirsikar

Publication Chair

Dr. Aninda Bose

Technical Review Committee

Dr. Prasad Kulkarni, Professor, Electrical Engineering and Computer Science, University of Kansas.
Dr. S. D. Joshi, Dean and Professor, Computer Department, BVCOE, Pune.
Dr. Sanjeev Wagh, Professor and Head, Information Technology Department, COE Karad.
Dr. B. B. Meshram, Professor, Computer Engineering and Information Technology, VJTI, Mumbai.
Dr. Anil Hiwale, Principal, MITCOE, Pune.
Dr. Kailas Patil, Professor, Vishwakarma University, Pune.
Dr. Shweta Dharmadhikari, PICT, Pune.
Dr. Sachin Sakhare, Professor and Head, VIIT, Pune.

Preface

The 2nd Springer International Conference on Computational Science and Applications (ICCSA 2019) was successfully organized by the School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, during August 7–9, 2019. The conference was supported by the **All India Council for Technical Education (AICTE)**. The objective of hosting ICCSA 2019 was to bring together experts for sharing knowledge, expertise and experience in the emerging trends of computer engineering and sciences.

The conference highlights the role of computational science in the increasingly interconnected world. The conference focused on recent developments in scalable scientific algorithms, advanced software tools, computational grids and novel application areas. These innovations will drive efficient application in allied areas. The conference discussed new issues and new methods to tackle complex problems and identified advanced solutions to shape new trends.

Research submissions in various advanced technology areas were received, and after a rigorous peer-review process with the help of program committee members and external reviewers, 44 papers were accepted. All the papers are published in Springer AIS series.

The conference featured eight special sessions on various cutting-edge technologies, which were conducted by eminent professors. Many distinguished personalities like Dr. Suresh Borkar, Illinois Institute of Technology, USA, Dr. Subhash Bhalla, University of Aizu, Japan, Dr. D. P. Chatterjee, Chittaranjan National Cancer Institute, Kolkata, and Dr. Chandrakant Pandav, WHO and UNICEF Consultant, Professor and Head of the Department, Centre for Community Medicine at the All India Institute of Medical Sciences (AIIMS), New Delhi, graced the conference.

Our sincere thanks to all special session chairs, distinguished guests and reviewers for their judicious technical support. Thanks to dynamic team members for organizing the event in a smooth manner. We are indebted to Dr. Vishwanath Karad MIT World Peace University for hosting the conference in their campus. Our entire organizing committee, faculty of MIT WPU and student volunteer deserve a mention for their tireless efforts to make the event a grand success.

Special thanks to our program chairs for carrying out an immaculate job. We would like to extend gratitude to our publication chairs who did a great job in making the conference widely visible.

Lastly, our heartfelt thanks to all authors without whom the conference would never have happened. Their technical contributions to make our proceedings rich are praiseworthy. We sincerely expect readers will find the chapters very useful and interesting.

Aizu-Wakamatsu, Japan
Yau Ma Tei, Hong Kong
Pune, India
Pune, India
Pune, India

Dr. Subhash Bhalla
Dr. Peter Kwan
Dr. Mangesh Bedekar
Dr. Rashmi Phalnikar
Dr. Sumedha Sirsikar

Contents

Part I Knowledge and Data Discovery

1 Empathic Diary Based on Emotion Recognition Using Convolutional Neural Network	3
Shreya Pendsey, Eshani Pendsey and Shweta Paranjape	
2 Detection of Ransomware Attack: A Review	15
Laxmi B. Bhagwat and Balaji M. Patil	
3 Room Service Robot	23
Elton Rodrigues, Pratik Sankhe, Swapnil Roge and Poonam Kadam	
4 Comparative Analysis for an Optimized Data-Driven System	33
Chinmay Pophale, Ankit Dani, Aditya Gutte, Brijesh Choudhary and Vandana Jagtap	
5 Fake Email and Spam Detection: User Feedback with Naives Bayesian Approach	41
Ayushi Gupta, Sushila Palwe and Devyani Keskar	
6 C-CASFT: Convolutional Neural Networks-Based Anti-spam Filtering Technique	49
Sunita Dhavale	
7 Cognitive Control of Robotic-Rehabilitation Device Using Emotiv EEG Headset	57
Neha Hooda, Ratan Das and Neelesh Kumar	
8 Non-stationary Data Stream Analysis: State-of-the-Art Challenges and Solutions	67
Varsha S. Khandekar and Pravin Srinath	

9	Parallel Job Execution to Minimize Overall Execution Time and Individual Schedule Time Using Modified Credit-Based Firefly Algorithm	81
	Hardeep Kaur and Anil Kumar	
10	A Novel Non-invasive Approach for Diagnosis of Medical Disorders Based on De Broglie's Matter Waves and Water Memory	91
	Vijay A. Kanade	
11	Tamper Detection in Cassandra and Redis Database—A Comparative Study	99
	Archana Golhar, Sakshi Janvir, Rupali Chopade and V. K. Pachghare	
12	Tamper Detection in MongoDB and CouchDB Database	109
	Rohit Kumbhare, Shivali Nimbalkar, Rupali Chopade and V. K. Pachghare	
13	Recommender System in eLearning: A Survey	119
	Pradnya V. Kulkarni, Sunil Rai and Rohini Kale	
14	A Realistic Mathematical Approach for Academic Feedback Analysis System	127
	Onkar Ekbote and Vandana Inamdar	
15	Fake News Classification on Twitter Using Flume, N-Gram Analysis, and Decision Tree Machine Learning Technique	139
	Devyani Keskar, Sushila Palwe and Ayushi Gupta	
16	Swarm Intelligence-Based Systems: A Review	149
	Vedant Bahel, Atharva Peshkar and Sugandha Singh	
17	Internet of Things: A Survey on Distributed Attack Detection Using Deep Learning Approach	157
	Saraswati Nagtilak, Sunil Rai and Rohini Kale	

Part II Image, Voice and Signal Processing

18	Precise Orbit and Clock Estimation of Navigational Satellite Using Extended Kalman Filter Applicable to IRNSS NavIC Receiver Data	169
	H. S. Varsha, Shreyanka B. Chougule, N. V. Vighnesam and K. L. Sudha	
19	Effects of Color on Visual Aesthetics Sense	181
	Shruti V. Asarkar and Madhura V. Phatak	
20	Performance Evaluation of Video Segmentation Metrics	195
	Shriya Patil and Krishna K. Warhade	

- 21 Suspicious Activity Detection Using Live Video Analysis** 203
Asmita Gorave, Srinibas Misra, Omkar Padir, Anirudha Patil
and Kshitij Ladole

- 22 A Review on Using Dental Images as a Screening Tool
for Osteoporosis** 215
Insha Majeed Wani and Sakshi Arora

- 23 An Expert Diagnosis System for Parkinson's Disease
Using Bagging-Based Ensemble of Polynomial Kernel SVMs
with Improved GA-SVM Features Selection** 227
Vinod J. Kadam, Atharv A. Kurdukar and Shivajirao M. Jadhav

Part III Communication and Networks

- 24 Case Study: Use of AWS Lambda for Building a Serverless
Chat Application** 237

Brijesh Choudhary, Chinmay Pophale, Aditya Gutte, Ankit Dani
and S. S. Sonawani

- 25 Detection and Classification of Diabetic Retinopathy Using
AlexNet Architecture of Convolutional Neural Networks** 245

Udayan Birajdar, Sanket Gadhave, Shreyas Chikodikar,
Shubham Dadhich and Shwetambari Chiwhane

- 26 Contextual Recommendation and Summary of Enterprise
Communication** 255

Anuja Watpade, Nikita Kokitkar, Parth Kulkarni, Vikas Kodag,
Mukta Takalikar and Harshad Saykhedkar

- 27 Cybersecurity and Communication Performance Improvement
of Industrial-IoT Network Toward Success of Machine Visioned
IR 4.0 Technology** 265

Sachin Sen and Chandimal Jayawardena

- 28 Dynamic Load Balancing in Software-Defined Networks
Using Machine Learning** 283

Kunal Rupani, Nikhil Punjabi, Mohnish Shamdasani
and Sheetal Chaudhari

Part IV Design and Application of Intelligent Computing and Communication

- 29 Analysis and Comparison of Timbral Audio Descriptors
with Traditional Audio Descriptors Used in Automatic Tabla Bol
Identification of North Indian Classical Music** 295
Shambhavi Shete and Saurabh Deshmukh

30	Sentiment Analysis on Aadhaar for Twitter Data—A Hybrid Classification Approach	309
	Priya Kumari and Md. Tanvir Uddin Haider	
31	Song Recommendation System Using Hybrid Approach	319
	Niket Doke and Deepali Joshi	
32	Arrhythmia Detection Using ECG Signal: A Survey	329
	Bhagyashri Bhirud and V. K. Pachghare	
33	Towards Designing the Best Model for Classification of Fish Species Using Deep Neural Networks	343
	Pranav Thorat, Raajas Tongaonkar and Vandana Jagtap	
34	A Study on Attribute-Based Predictive Modelling for Personal Systems and Components—A Machine Learning and Deep Learning-Based Predictive Framework	353
	Aswin Ramachandran Nair, M. Raj Mohan and Sudhansu Patra	
35	Text Categorization Using Sentiment Analysis	361
	Chaitanya Bhagat and Deepak Mane	
36	Automated Real-Time Email Classification System Based on Machine Learning	369
	Sudhir Deshmukh and Sunita Dhavale	
37	Smart Detection of Parking Rates and Determining the Occupancy	381
	Deepali Javale, Aatish Pahade, Rushikesh Singal, Akshay Potdar and Mohit Patil	
38	Deep Learning-Based Approach to Classify Praises or Complaints from Customer Reviews	391
	Sujata Khedkar and Subhash Shinde	
39	Psychological Behavioural Analysis of Defaulter Students	403
	Rucha Karanje, Sneha Jadhav, Divya Verma and Shilpa Lambor	
40	TNM Cancer Stage Detection from Unstructured Pathology Reports of Breast Cancer Patients	411
	Pratiksha R. Deshmukh and Rashmi Phalnikar	
41	Restructuring of Object-Oriented Software System Using Clustering Techniques	419
	Sarika Bobde and Rashmi Phalnikar	
42	Analysis of System Logs for Pattern Detection and Anomaly Prediction	427
	Juily Kulkarni, Shivani Joshi, Shriya Bapat and Ketaki Jambhali	

Contents	xiii
43 Phishing Detection: Malicious and Benign Websites Classification Using Machine Learning Techniques	437
Sumit Chavan, Aditya Inamdar, Avanti Dorle, Siddhivinayak Kulkarni and Xin-Wen Wu	
44 Automation of Paper Setting and Identification of Difficulty Level of Questions and Question Papers	447
Ayesha Pathan and Pravin Futane	
Author Index	459

About the Editors

Dr. Subhash Bhalla joined the faculty of Jawaharlal Nehru University (JNU), New Delhi in 1986, at the School of Computer and Systems Sciences. He was a post doctoral fellow at Sloan School of Management, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA (1987-88). He is a member of the Computer Society of IEEE and SIGMOD of ACM. He is with the Department of Computer Software at the University of Aizu. He has also toured and lectured at many industries for conducting feasibility studies and for adoption of modern techniques. He has received several grants for research projects. Prof. Bhalla currently participates in research activities on- New query languages, Big data repositories in science and astronomy, Standardized Electronic Health Records, Polystore Data Management, Edge Computing and Cloud based Databases. He is exploring database designs to support models for Information Interchange through the World Wide Web.

Dr. Peter Kwan graduated in 1984 at the University of Glasgow and started his career as a graduate engineer in 1985 in Scotland. Later he got an MBA degree in 1990 and a DBA in 2008. His expertise are in project & construction management, energy conservation audit, and M & Engineering. He has extensive experience in hotels, hospitals, shopping centers, club houses and various military establishments and marine structures. His last 10 years are devoted in data center engineering design and facilities management. He has a few research and publications in energy conservation, applications of Transwall and in innovation. He is a fellow member of the UK Chartered Institutions of Building Services Engineers and had served as a Subject Matter Expert in the HK Council of Academic Accreditation for 6 years and had been a Chartered Engineer interviewer before now devoting his key interests in training and teaching engineering students.

Dr. Mangesh Bedekar is currently the Professor and Head of the School of Computer Engineering and Technology at Dr. Vishanath Karad, MIT World Peace University based in Kothrud, Pune, Maharashtra, INDIA. His research interests are in the fields of User Modeling, Web Personalization, Web Data Mining, Browser

Customization, and User Interface Improvements. He has published 70+ papers in various Conferences and Journals. He is also associated with the Technology Business Incubator and advisor to start-up companies.

Dr. Rashmi Phalnikar has been working as an Associate Professor at School of Computer Engineering and Technology at Dr. Vishanath Karad, MIT World Peace University, Pune, India. She has published more than 50 papers in Journals of International repute and also presented papers in conferences around the world.

Her research has mainly revolved around the development of advanced methods in areas of Software Engineering in User Driven Software Application, Role of Non Functional Requirements, Aspect Oriented Software Development, and Application Areas of Machine Learning in healthcare.

Prof. Sumedha Sirsikar joined Maharashtra Institute of Technology, Pune, India in August 1995, at the Computer Engineering Department. She is with the Department of Computer Science and Engineering at the University MIT WPU, Pune, India. Prof. Sirsikar currently participates in providing solutions to design and create several modules in ERP of MIT WPU. She is also working on performance evaluation and reduction in energy consumption of mobile wireless sensor network using clustering algorithm. She had contributed in developing courses in Computer Networks and Security which was used in the University of Pune, India. Sumedha received her M.E. degree in Computer Engineering from University of Pune, Maharashtra, India, in 2001. Recently, she has completed Doctoral degree with Research Laboratory of Faculty of Engineering and Technology at Sant Gadge Baba Amravati University, Maharashtra, India. Her current research interests include wireless ad hoc networks and self-organization wireless sensor networks.

Part I

Knowledge and Data Discovery

Chapter 1

Empathic Diary Based on Emotion Recognition Using Convolutional Neural Network



Shreya Pendsey, Eshani Pendsey and Shweta Paranjape

1 Introduction

Mental health is extremely crucial to the overall well-being of a person and in turn affects society as a whole. These days many people are suffering from depression, anxiety, and feeling out of control. They feel a need to monitor their everyday activities and the way they feel about them. Initially it can be recovered by keeping track of daily situations and emotion associated with it. These can be recorded, retrieved, and monitored in the form of a diary. A web application called empathic diary is being implemented by us, which does the same with the provision of recognizing the users' emotions. User can upload a picture and provide a simple note about the scenario if needed, for better understanding of the situation and their response to that situation. The current emotion of the user will be detected and displayed on the screen. User can also check past records in the diary which will help them monitor their past behavior. The diary will help users record their immediate emotions to different stimuli throughout the day thus helping them take a better decision about the necessary action. It can be used by psychiatrists as a primary treatment for some patients or by users who wish to simply monitor their emotions or reactions. It can also be used by potential patients wishing to monitor their need to see a therapist in moments of extreme anxiety.

S. Pendsey · E. Pendsey · S. Paranjape (✉)

Pune Vidyarthi Gruha's College of Engineering and Technology, Pune, India

e-mail: paranjape.shweta1997@gmail.com

S. Pendsey

e-mail: shrependsey@gmail.com

E. Pendsey

e-mail: eshani.pendsey@gmail.com

2 Related Work

Reference [1] considers two methods for facial emotion detection, namely representational autoencoder units(RAUs) and convolutional neural network(CNN). Autoencoders are a class of neural networks capable of reconstructing their own input in a lower dimensional space. The other method used was a deep learning CNN with eight layers. A better accuracy was obtained for the CNN compared to the RAU.

Reference [2] considers deep and shallow architectures of CNNs for an image size of 48×48 pixels for facial expression recognition. Reference [3] considers a CNN to accomplish the tasks of emotion and gender classification simultaneously. Both these consider seven emotions, namely fear, anger, disgusted, surprised, glad, sad, and neutral.

In [4], a FER system is examined that uses a combination of CNN and a few specific preprocessing steps that are aimed to contribute to the accuracy of the system. Experiments showed a significant improvement in the method's accuracy with the combination of these procedures.

Use of the Emotient API is a great method for applications that wish to track attention and engagement from viewers. The RESTful API can be easily integrated into applications that demand for emotion detection.

A catalog of artificial intelligence APIs based on computer vision is the Microsoft's Project Oxford. It works with photos and detects faces. The response is in JSON that contains specific percentages for each face for the seven main emotions, as well as neutral.

Emotion API: The Emotion API demands an image containing a facial expression as an input and gives us as output the respective confidence across a set of emotions for every face in the image as output. A user having called already the face API can submit the rectangle of face as an input as well. The emotions detected are contempt, angry, fear, disgusted, glad, sad, neutral, and surprised.

3 System Overview

The empathetic diary is based on recognizing the emotion from the image uploaded by the user, using a convolutional neural network for the same. User may add a note along, for better monitoring his/her reaction to stimuli. The CNN is trained for recognition from amongst five emotions namely: happiness, sadness, anger, fear, and surprise. The architecture for the system can be seen in Fig. 1. The face present is first detected using a haar-cascade classifier, cropped, gray scaled, and extracted from the image taken from the user. The resultant image is then fed to the CNN which returns the confidence for each of the five emotions for that particular image, and the emotion with the highest score is returned as the detected emotion to the user. The records are stored and presented to the user at request. Each record consists of the detected emotion, the note uploaded by user along with it that could contain the

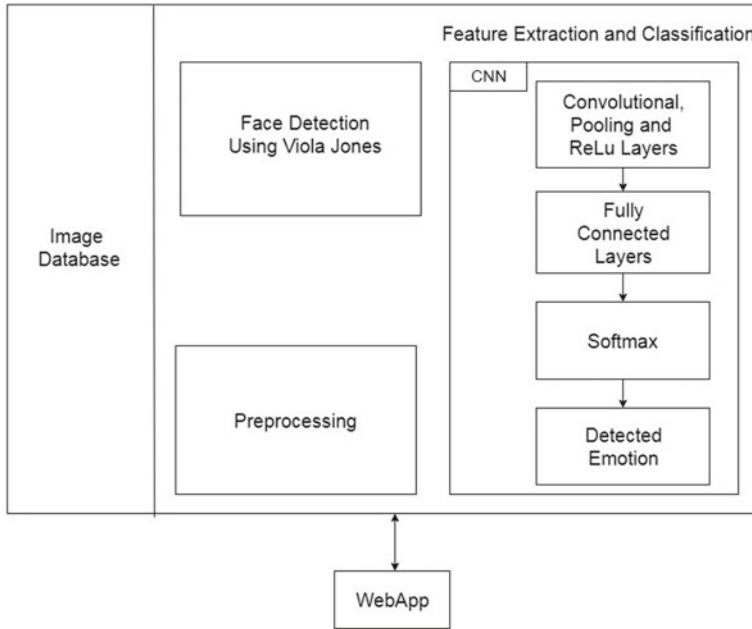


Fig. 1 System architecture

stimulus resulting in the particular emotion, and the date and time this record was made. Our model for empathic diary is trained on images from the Japanese Female Facial Expressions (JAFFE) [5] database as well as the Extended Cohn-Kanade database [6]. The Jaffe database contains a total of 213 images in seven distinct facial expressions out of which only the needed five were extracted for training our CNN. The CK+ consists of 593 different sequences of images. 327 of them have discrete image labels. A few images were taken for each subject from the available image sequences for the five required emotions that are anger, surprise, happiness, sadness and fear. A few sample images can be seen in Fig. 2.

Preprocessing is applied on each of these images so as to reduce their dimensionality and obtain better results as seen in [1, 4]. The image is cropped to show only the facial region so as to reduce the noise generated by background (as seen in [7]), and converted into a grayscale image so as to reduce the dimensionality aspect of the image. The collective dataset obtained so consisted of 827 images with discrete emotion labels for the required five emotions. All of these images were reshaped to the size 128×128 pixels to be fed to the CNN. A few sample images after preprocessing is performed on them can be seen in Fig. 3.



Fig. 2 Sample images from JAFFE dataset and [5] and Extended Cohn-Kanade dataset [6]

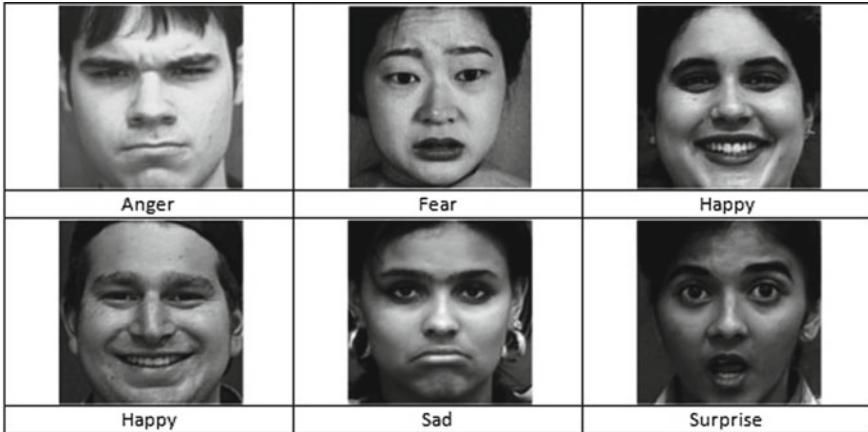


Fig. 3 Sample preprocessed images

4 Convolutional Neural Network

The Architecture of our CNN: The network receives as input a 128×128 image which is grayscale and outputs the confidence for each expression which one may visualize as a one dimensional matrix where each element represents an emotion. The class with the maximum value is used as the resultant expression in the image. The first layer of the CNN is a convolution layer that applies a convolution kernel of 4×4 . This layer is followed by a pooling layer that uses max-pooling (with kernel size 2×2) to reduce the image to half of its size. Subsequently, a new convolution layer performs convolutions with a 8×8 kernel to map of the previous

layer and is followed by another pooling, with a 4×4 kernel. Another convolution layer performs convolutions with a 4×4 kernel to map of the previous layer and is followed by another pooling, again with a 2×2 kernel. The outputs are given to a fully connected hidden layer that has 256 neurons. It considers a dropout of 0.4, in the view of avoiding overfitting. Finally, the network has five output nodes (one for each expression that outputs their confidence level) that are fully connected to the previous layer. The images that have been preprocessed are used to train the convolutional neural network. The architecture of this CNN can be seen in following Fig. 4.

5 System Screenshots

Following screenshots depict the action needed from users to use the empathic diary. These are the screenshots from our developed system (Figs. 5, 6 and 7).

6 Result

The total dataset consisted of 827 images belonging to the Extended Cohn-Kanade [6] and the JAFFE [5] datasets, grayscaled and cropped to include just the facial region, and reshaped into a dimension of 128×128 pixels. Out of these, 618 images were included for training set and the remaining 209 images were used for testing set for our CNN. The results obtained can be visualized as the following Tables 1 and 2 containing the accuracy and confusion scores for each emotion. It can be observed that the emotion happy, being the only positive emotion considered, can be easily differentiated from the other emotions, as is the ideal case. The other emotions comparatively have slightly more confusion amongst themselves, with the highest being that of fear being classified as surprise.

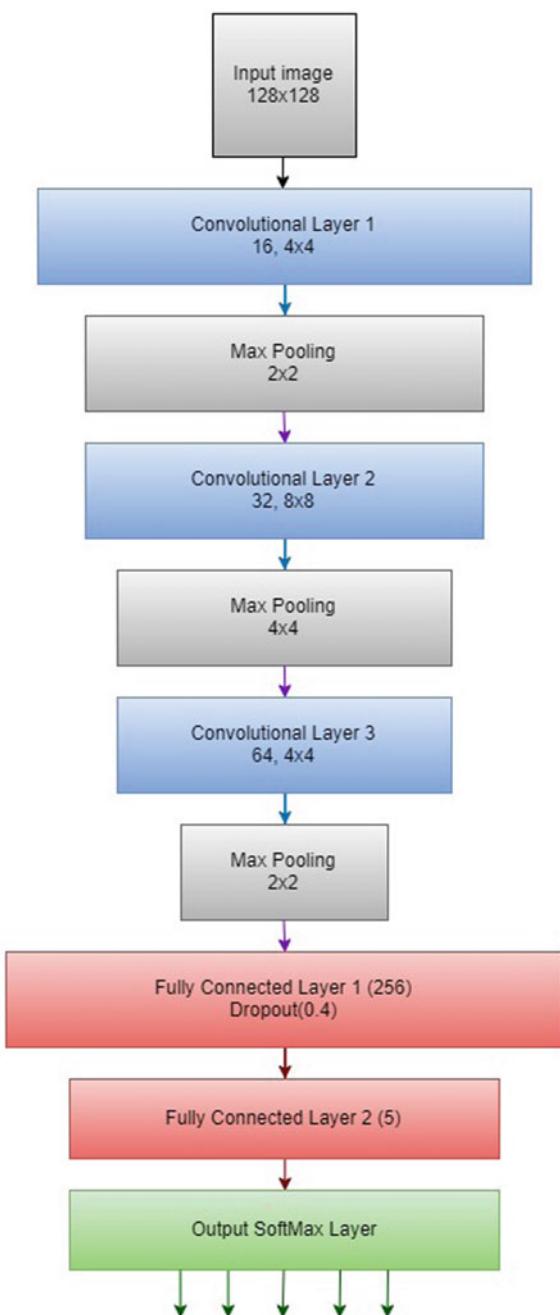
Let us consider a random test image from the wild and visualize with the help of it, the computations taking place in each layer. The images that will follow are a set of activation maps obtained for each layer presented as a horizontal grid of the same. These activation maps give us a clear idea of the feature maps in the CNN itself that are obtained through training.

The image is first preprocessed as shown in Figs. 8 and 9. The activation maps obtained though the corresponding convolutional and pooling layers can be seen through each image as shown in Figs. 10 and 11.

After the last pooling layer (Fig. 12).

As we go deeper through the layers, it can be seen that more abstract and specific features are seen through the activation maps (Fig. 13).

The image corresponding to the flattening after the final pooling layer shows 1600 different activations represented instead as a uniform image for better visualization or comparison, in contrast to the horizontal grids representing specific feature maps. The

Fig. 4 CNN architecture

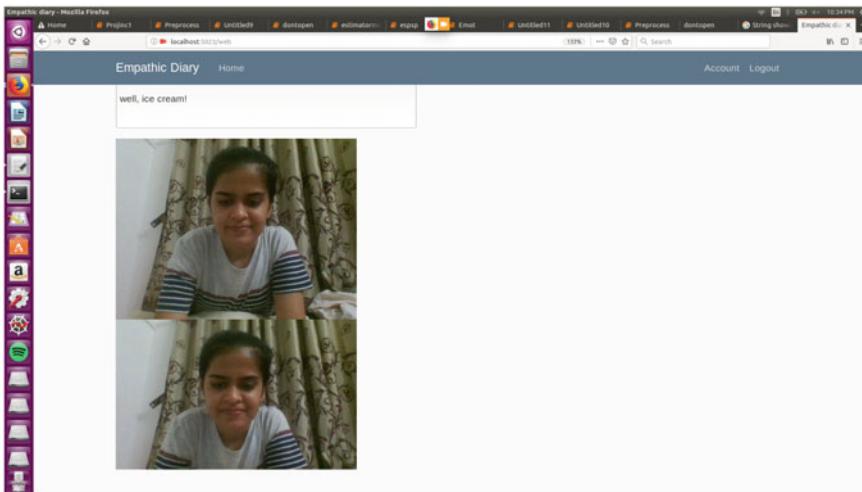


Fig. 5 Using webcam to capture an image

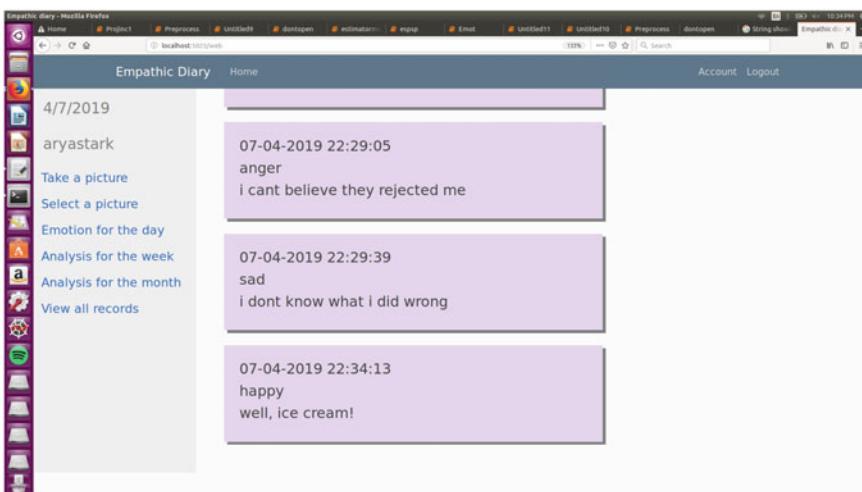


Fig. 6 New entry with identified emotion

flattened and corresponding layers are for the purpose of visualization of activations only (Fig. 14).

The dense layer shown is also represented here as a grid of different activation values whereas in reality one can visualize it as a simple sequence of 256 different individual activation values. The final layer is the output layer corresponding to required activations from the previous dense or fully connected layer and shows a

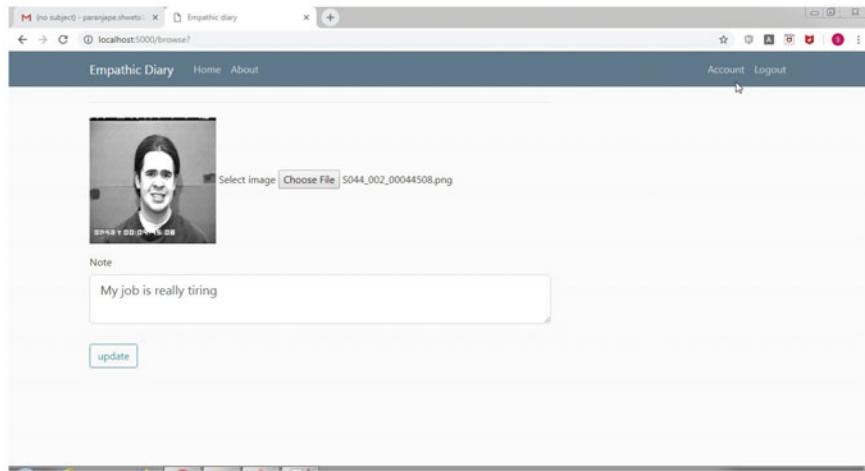


Fig. 7 Uploading an image previously captured

Table 1 Accuracy metrics

	Precision	Recall	F1-score
Surprise	0.82	0.89	0.85
Happiness	0.97	0.98	0.97
Anger	0.81	0.90	0.85
Fear	0.89	0.68	0.77
Sadness	0.87	0.89	0.88
Average	0.88	0.88	0.88

Table 2 Confusion matrix

	Anger	Happy	Sad	Fear	Surprise
Anger	26	0	3	0	0
Happy	1	58	0	0	0
Sad	3	0	34	1	0
Fear	0	2	1	25	9
Surprise	2	0	1	2	41

high activation (in this case, the second value is the highest which indeed was an indicator for the emotion ‘happy’) for respective emotion (Fig. 15).



Fig. 8 Sample image



Fig. 9 Sample image after preprocessing

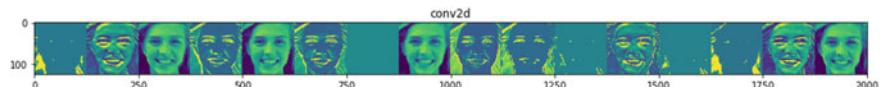


Fig. 10 Convolution layer 1



Fig. 11 Pooling layer 1

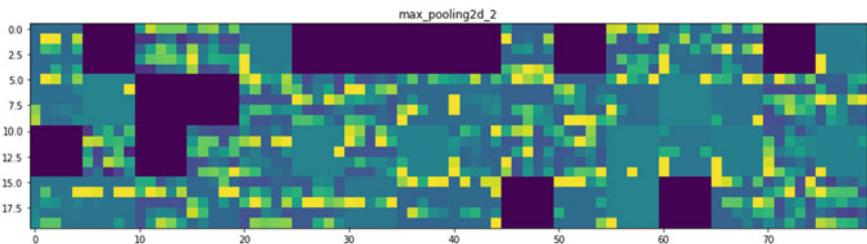
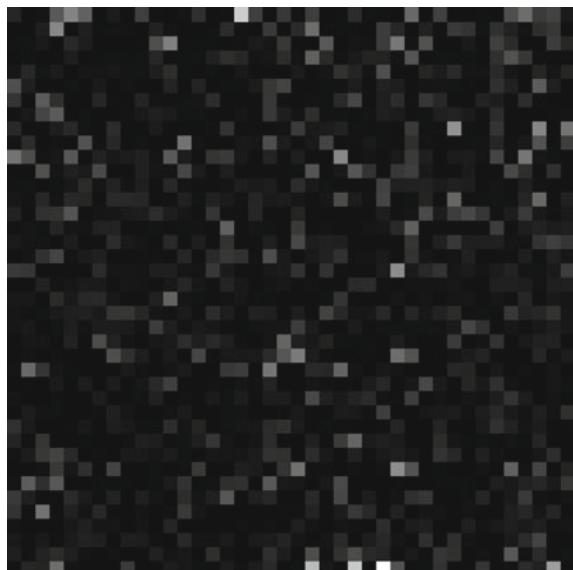


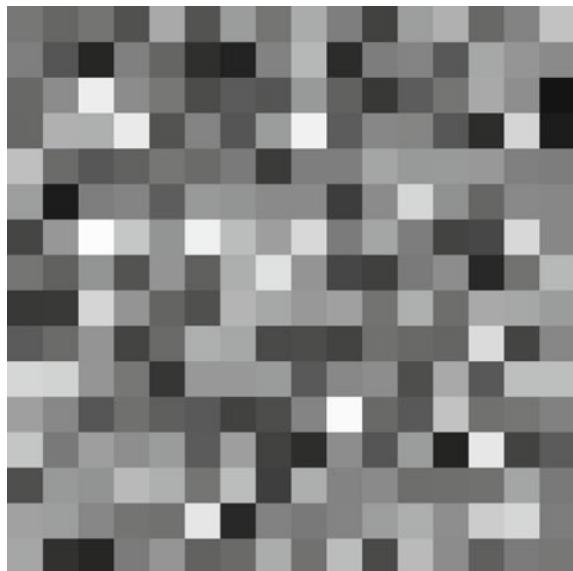
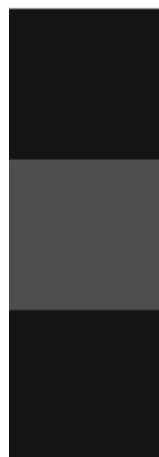
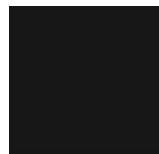
Fig. 12 Pooling layer 3

Fig. 13 Previous layer, flattened



7 Conclusion and Future Scope

We propose a facial emotion recognition system that uses a convolutional neural network preceded by a few image preprocessing techniques. The empathetic diary detects facial emotion from an image provided by the users themselves. There will be scope to increase accuracy of classifying emotion by adding more and more datasets. It will help to exactly detect right emotion. In future, we can add various responses from diary for respective emotion, and responses will vary every time even though the emotion is same. That way, diary will help users to make them happy if they are feeling low or suggest a nearby psychologist if needed. Additionally, text analysis over uploaded note generating keywords or summaries could help analyze associations between stimuli and emotions for any individual. Also, feature extraction is crucial for any recognition algorithm and system. A remarkable change will be noticed in the recognition rate using preprocessing and different feature extraction.

Fig. 14 Dense layer 1**Fig. 15** Output layer

Specially in cropping an image before it is run through a recognition system, there is still much work to be done in this area. It would be interesting to explore new techniques of preprocessing that would lead to the optimal recognition rates. Larger datasets involving inclusivity in terms of different age groups, ethnicity, and cultural diversity would enable any FER to produce ideal results for a large span of users.

References

1. Dachapally PR. Facial emotion detection using CNN and RAUs. Indiana University
2. Shima A, Fazel A. Convolutional neural networks for facial expression recognition. ArXiv2016
3. Real-time Convolutional Neural Networks for Emotion and Gender Classification: Octavio Arriaga, Paul G. Ploger, Matias Valdenegro, ArXiv2017
4. Lopesa AT, de Aguiarb E, De Souza AF, Oliveira-Santosa T (2017) Facial expression recognition with convolutional neural networks: coping with few data and the training sample order
5. Lyons MJ, Akemastu S, Kamachi M, Gyoba J (1998) Coding facial expressions with Gabor wavelets. In: 3rd IEEE international conference on automatic face and gesture recognition, pp 200–205
6. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z (2010) The extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 152131 Department of Psychology, University of Pittsburgh, Pittsburgh, PA, 152602
7. Kim S, An GH, Kang S-J. Facial expression recognition system using machine learning. In: ISOCC 2017. IEEE

Chapter 2

Detection of Ransomware Attack: A Review



Laxmi B. Bhagwat and Balaji M. Patil

1 Introduction

Ransomware is a new buzz word nowadays to attract the organizations to take necessary action against it. As many of us know, it is a malware that comes in the category of extortion as the intention behind it. It is a malware which acts by encrypting all the important files in the file system. Because of this there is a huge damage to the personal computers as well as big organizations. Also people has less awareness about this type of attack.

There are different ways with which this attack can be carried out by the attacker. To get install on the victim's computer the victim must download by any means the malicious code on the victim's machine. This can be done by the attacker by luring the victim to click on some link. Once it gets downloaded on the victim's machine it starts acting silently by exchanging some handshake control commands between the malicious code and control server and the malicious software on the victim's machine. After some control commands from the server which is away from the victim's machine, the malicious code starts taking actions according to the commands given by the control server.

There are different stages in which this execution of attack is carried out. With reference to Fig. 1 [1–3], the first stage is deployment, where it tries to get into the system. This is done by sending phishing e-mails which seems to come from some authentic person/friend or using through social forums. The deployment can also be done by exploiting the system vulnerabilities. The second stage is installation. Once the deployment is done without the knowledge of the end user, and

L. B. Bhagwat (✉) · B. M. Patil

School of Computer Engineering and Technology, MIT-WPU, Pune, Pune, India
e-mail: laxmi.bhagwat@mitwpu.edu.in

B. M. Patil
e-mail: balaji.patil@mitwpu.edu.in

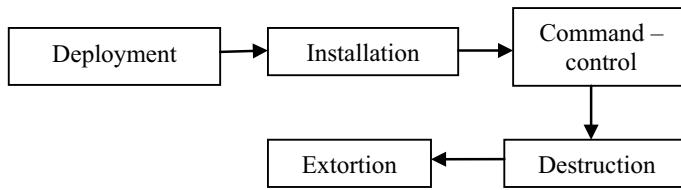


Fig. 1 Phases of ransomware

the installation and infection start silently. The first task is to evade the detection and have a communication between the attacker and the target machine. If it finds the required vulnerabilities, it does the MD5 of the machine name or MAC address so that the extortionist can know which machine has been compromised. After the commands that are received by the command and control server, the malicious code on the machine starts the destruction of the file system. Once the destruction is done the malicious code displays the message for the payment of the ransom for the data access. The main difference between a malware attack and the ransomware attack is the extortion in terms of money and no access to data, which is not in malware attacks.

2 Overview of Different Techniques for Detection of Ransomware Attack

There are different detection techniques used for the detection and analysis of the attack. Following are some of the recent techniques for the detection of ransomware attack.

Operating system-level detection where solution is provided for the access to file system.

Application code level detection is done using a data-flow graph for the cryptographic function calls or at system-level calls.

Machine learning-based detection techniques

- a. Static-based analysis and detection technique
- b. Dynamic-based analysis and detection technique
- a. Static-based analysis technique

In static-based analysis technique, the application's code is examined before it is executed to determine if it is a malicious code or not. In this technique, the signature of the code is compared with the already available repository of known malicious code patterns. If a malicious pattern is found that application is stopped from execution. The major advantage is that before the application gets executed it can be detected as a malicious application.

The major drawback of static-based detection technique is, repository of known malicious code should be frequently updated so that new type of malware attacks can be detected otherwise they will go unnoticed.

b. Dynamic-based analysis technique

In dynamic-based analysis technique, the live examination of the processes is done while the application is executing. If some malicious behavior of the process is detected, then it is signaled as a malicious application and its further execution is stopped. The major advantage of this technique is new malware can be detected by this technique.

The drawback of this technique is if the detection of the malware is not done early, there can be damage to the system.

These are some of the major detection techniques used till now for the detection and analysis of ransomware attack.

3 Literature Review

Han et al. [4] have contributed by alerting and making aware about ransomware attack and how to tackle them. The solution given by them is development of browser extension which provides assistance to users by alerting them of plausible dangers while surfing the Internet. It allows the users to surf the web safely by generating the required alert.

Ami et al. [5] have presented AntiBotics which is a new authentication-based file access control system. AntiBotics makes sure that a user applies file access control policy by presenting periodic identification/authorization challenges. An authentication-based access control mechanism is implemented by assigning access permissions to execution objects, which is based on policies specified by an administrator and also based on the responses to challenges that are presented upon attempts to modify or delete files. The major components of the system are

- (1) The Policy Enforcement Driver (PED)
- (2) The Policy Specification Interface (PSI), and
- (3) The Challenge-response Generator (CRG).

Che et al. [6] overcome the drawback of signature- and static-based ransomware detection approach using data mining techniques such as Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), and Simple Logistics (SL) algorithms for detecting known and unknown ransomware by proposing a dynamic ransomware detection system. The API call flow graph was generated by monitoring the behavior of the software. Then the feature selection and data normalization techniques were applied to select best feature attribute for building a data mining model by using different data mining algorithms, which will correctly detect whether a software is benign or ransomware. The main drawback is there is no standard data

set for comparison for dynamic analysis of ransomware so the authors could extract only 168 samples of the software for estimating the system.

Shukla et al. [7] worked on gaps found by them in the solutions provided by other researchers and have provided solution as a dynamic system which learns new behavior while the system might be under attack.

The first solution is related to system side, which is named as FileTracker and is implemented as a client–server architecture in a distributed manner. There are two major modules: One is client module which consists of a user mode analytics component and second is the kernel module for event monitoring. The kernel mode is further divided into two modules: (i) file system filter driver and process monitoring driver. The filter driver is responsible for monitoring I/O events and process monitor which is responsible for process-related events. (ii) The FileTracker acts as a aggregator for all the system events which are collected from each node and it builds a global model for abnormal and normal behavior of file access and modification.

Kharraz and Kirda [8] proposed Redemption, as a new defense mechanism that makes the operating system more strong enough to deal with ransomware attacks. The main component in this technique that is applied for the detection of ransomware attack is the abstract model for the current class of ransomware behavior that is constructed by minimal modification to OS.

Continella et al. [9] propose ShieldFS, which is an add-on driver that makes the Windows native file system unaffected to ransomware attacks. ShieldFS monitors the file system model at low level. For each of the processes running in the system, ShieldFS toggles a protection layer according to the detection component. Whenever there is a violation by a process in the model which seems to be malicious the side effect on the file system is rolled back, it then uses machine learning technique to select important features for the detection of ransomware attack.

Lestringant and Fouquey [10] have presented a novel approach to automatically identify symmetric cryptographic algorithms and their parameters inside binary code. They have used a static approach, based on data-flow graph (DFG) isomorphism. The data-flow graph isomorphism was accurately able to identify and locate cryptographic algorithms and their parameters inside binary executables. Authors have targeted only symmetric cryptographic algorithms.

Daniele Sgandurra et al. implemented EldeRan [11], which used machine learning to dynamically analyze and classifying ransomware. In first phase, while installation of the infection vector EldeRan examines a set of actions that are performed by different applications which does checking for the characteristics of ransomware. In the second phase, they have used Logistic Regression classifier. The main aim of this classifier is to model the log-posterior probability of the different classes when the data is given via linear function depending on the features.

Lee et al. [12] have proposed a method to make decoy files which are dummy folders and files for detecting ransomware efficiently. They did the source code level analysis for already existing ransomware. Different types of ransomware that have been already released were analyzed. The analysis was based on how the files are accessed and infected in Windows operating system.

Daku et al. [13] have done behavioral-based detection of ransomware attack. Many attackers use techniques such as polymorphic and metamorphic approach to avoid their detection. So, Hajredin Daku et al. have used machine learning techniques on their behavior to identify the modified variants of signature-based systems. Hajredin Daku et al. have studied behavior samples of 150 ransomware from 10 different ransomware families. To have the best possible results an iterative approach was used to identify the behavioral attributes for best classification accuracy.

To achieve best classification attribute the feature selection should be done correctly. So the study was divided into two parts, one was the feature selection and the other was to have best classification accuracy.

Min et al. [14] implemented hardware level detection of ransomware attack. They have proposed an SSD system that does automated backup which is called as Amoeba. Amoeba is a hardware accelerator that detects infection of pages by ransomware and a control mechanism to minimize the space overhead for the original data backup. They have simulated Microsoft SSD to implement Amoeba. It was evaluated on block-level traces.

Wang et al. [15] propose a system RansomTracer which combats the untargeted attacks such as Remote Desktop Protocol (RDP). This is achieved by creating a deception environment which monitors the attacker and collects clues about the attacker. After this, it automatically extracts and analyzes these clues for detection of ransomware attack.

4 Findings and Discussion on Detection of Ransomware Attack

Some of the current challenges for detection of ransomware attack. According to the above survey done, following challenges needs to be overcome by the researchers:

- (i) According to Han et al. [4] proposed browser extension to be developed which should provide assistance and warn users of plausible dangers while surfing on the Internet. This was just conceptual but implementation of such concept may prove to be useful for detection of ransomware.
- (ii) Ami et al. [5] have presented AntiBotics which is a new authentication-based file access control system. The major drawback of the current implementation is, when new permits are created they all are set according to the same global permit duration and permit scope configuration parameters, which are independent of the PID or program to which they are granted. Also, their granularity is determined globally according to the system-wide execution object type parameter.
- (iii) Che et al. [6] overcome the drawback of signature- and static-based ransomware detection approach using data mining techniques. But, the future work can be focusing on collection of more samples and data features to further optimize the system.

- (iv) Shukla et al. [7] worked on gaps found by them in the solutions provided by other researchers and have described a dynamic system which learns new behavior while the system might be under attack. In the future, the work will have to be in hardening identification logic and make the virtualization layer more robust.
- (v) Kharraz and Kirda [8] proposed Redemption, as a new defense mechanism that makes the operating system more strong enough to deal with ransomware attacks. Future work can be reducing the delay in accessing the files from the file system as it requires changes in the operating system.
- (vi) Continella et al. [9] propose ShieldFS, which is an add-on driver that makes the Windows native file system unaffected to ransomware attacks. ShieldFS monitors the file system model at low level. The major drawback is if the attacker could guess exactly the thresholds of the classifiers and value of the parameter T , it may attempt to perform a mimicry attack and encrypt few files so that it remains below the thresholds until T hours and act as a benign file. So, setting the T value correctly is very important.
- (vii) Lestringant and Fouquey [10] have presented a novel approach to automatically identify symmetric cryptographic algorithms and their parameters inside binary code.
- (viii) Daniele Sgandurra et al. EldeRan [11] used machine learning to dynamically analyze and classifying ransomware. It is difficult to detect and analyze the samples of ransomware that remain silent for some amount of time to avoid their detection. The second limitation is that when the analysis was done it had only the applications that were running in the VM machine that were available when the installation of Windows OS in VM. No other applications other than that were running and analyzed.
- (ix) Lee et al. [12] have proposed a method to make decoy files which are dummy folders and files for detecting ransomware efficiently. The main problem that is faced is the size order of decoy files, i.e., the value of n . If n is small the ransomware may encrypt the important file and if it is large, it will occupy the storage space. That is why value of n should be set properly.
- (x) Daku et al. [13] have done behavioral-based detection of ransomware attack. The future work can be the selection of the best behavioral attributes for classification and detection of ransomware.
- (xi) Min et al. [14] implemented hardware level detection of ransomware attack. They have proposed an SSD system that does automated backup which is called as Amoeba.

5 Conclusion

In this paper, we have given an elaborated overview of various techniques and methods for the detection and analysis of ransomware attack. According to our study, it is clear that there are different techniques and methods like OS/file system level, hardware level, application's binary code level or system call level that can be used for the detection of ransomware attack. Also the detection of the ransomware attack can be done using various data mining and machine learning algorithms. Hence from the study, we can conclude that more work is needed at the file system level as it directly deals with the file system.

References

1. Evolution of Ransomware, Review Article, Symantec Response, IET Networks, The Institution of Engineering and Technology 2018
2. Tuttle H. Ransomware attacks pose growing threat. <http://www.rmmagazine.com/2016/05/02/ransomware-attacks-pose-growing-threat/>
3. Liska A, Gallo T: Ransomware-defending against digital extortion. Blueprint for O'Reilly Media
4. Han JW, Hoe OJ, Wing JS, Brohi SN (2017) A conceptual security approach with awareness strategy and implementation policy to eliminate ransomware. In: CSAI 2017 proceedings of the 2017 international conference on computer science and artificial intelligence, Jakarta, Indonesia, pp 222–226. ACM
5. Ami O, Elovici Y, Hendler D (2018) Ransomware prevention using application authentication-based file access control. In: Proceedings of SAC2018: symposium on applied computing, Pau, France
6. Che Z-G, , Kang H-S, Kim S-R (2017) Automatic ransomware detection and analysis based on dynamic API calls flow graph. In: RACS '17 proceedings of the international conference on research in adaptive and convergent systems, Krakow, Poland. ACM, pp 196–201
7. Shukla M, Mondal S, Lodha S (2016) POSTER: locally virtualized environment for mitigating ransomware threat. In: CCS '16 Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, Vienna, Austria. ACM, pp 1784–1786
8. Kharraz A, Kirda E (2017) Redemption: real-time protection against ransomware at end-hosts. In: Dacier M, Bailey M, Polychronakis M, Antonakakis M (eds) Research in attacks, intrusions, and defenses. RAID 2017. LNCS, vol 10453. Springer, Cham, pp 98–119
9. Continella A, Guagnelli A, Zingaro G, De Pasquale G, Barenghi A, Zanero S, Maggi F (2016) ShieldFS: a self-healing, ransomware-aware filesystem. In: ACSAC '16 proceedings of the 32nd annual conference on computer security applications, Los Angeles, CA, USA. ACM, pp. 336–347
10. Lestringant P, Fouquey P-A (2015) Automated identification of cryptographic primitives in binary code with data flow graph isomorphism. In: ASIA CCS '15 proceedings of the 10th ACM symposium on information, computer and communications security, Singapore, Republic of Singapore. ACM, pp 203–214
11. Sgandurra D, Muñoz-González L, Mohsen R, Lupu EC (2016) Automated dynamic analysis of ransomware: benefits, limitations and use for detection. ArXiv 2016
12. Lee J, Lee J, Hong J (2017) How to make efficient decoy files for ransomware detection? In: RACS'17, proceedings of the international conference on research in adaptive and convergent systems, Krakow, Poland. ACM, pp 208–212

13. Daku H, Zavarsky P, Malik Y (2018) Behavioral-based classification and identification of ransomware variants using machine learning. In: 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering, New York, NY, USA. IEEE
14. Min D, Park D, Ah J, Walker R, Lee J, Park S, Kim Y (2018) Amoeba: an autonomous backup and recovery SSD for ransomware attack defense. *IEEE Comput Archit Lett* 17:245–248 (IEEE)
15. Wang ZH, Wu X, Liu CG, Liu QX, Zhang JL (2018) RansomTracer: exploiting cyber deception for ransomware tracing. In: 2018 IEEE third international conference on data science in cyberspace, Guangzhou, China. IEEE

Chapter 3

Room Service Robot



Elton Rodrigues, Pratik Sankhe, Swapnil Roge and Poonam Kadam

1 Introduction

Hotel industry is booming and new technologies are implemented every year. The heart of hospitality is about customer satisfaction. Automation is taking place in every industry. The processes which are manually done presently and require less or no human intelligence are replaced by robots. Progress in artificial intelligence is making it possible to make a complicated system autonomous [1]. For this purpose, a versatile robot can be designed for room service in hotels. If we compare robots and automated machines, robots have the advantage of being mobile and being human-like. Thus, they can serve us in better ways. Using these robots will serve as a great help to those entrepreneurs who desire to open hotels, but have a tight budget to handle it. That is because such robots are one-time investments. This robot will only act as a helping hand to the hotel staff. This project is aimed at low-finance hotels in order to gain more or less the same improvement and experience as perceived by the high-end hoteliers around the world. It is observed that if room service is done by robot instead of human beings, the room service order increases by 5% [2, 3]. We have used Raspberry Pi 3 Model B as a microcontroller which controls the working of the robot. In this project, we have implemented a four wheel-two motor system for the purpose of traversing around the hotel floor.

E. Rodrigues · P. Sankhe · S. Roge (✉) · P. Kadam
Electronics and Telecommunications, D. J. Sanghvi College of Engineering,
Vile Parle, Mumbai 400056, India
e-mail: rogeswapnil@gmail.com

2 Literature Survey

The trends in artificial intelligence and robotics are dominating the modern technology era and greatly impact the services sector. Human–robot interaction (HRI) is more focused by researchers of current generation. Nowadays, there is a huge trend of integrating AI and robots into tourism and hospitality industry, so there is a serious need for these robots, and thus, our project is proposed to lie on the same line. The famous M Social Hotel in Singapore has kept a robot named ‘AURA the relay robot’ on trial basis. Their motive is to please the guests apart from acting as a helping hand to the staff so that they can focus more on the other complex tasks like improving the guest experiences, etc. In the hospitality management sector, most of the human staff do very simple jobs, which can be easily replaced by robots so that the human staff can do more complex jobs. To solve this problem, Alibaba AI Labs had decided to develop a robot for hospitality management sector for performing simple tasks like making deliveries to guests, and they have launched these types of robots on September 2018. So, introduction of robot in this sector is at a developing stage, and it is open to further developments.

3 Description

3.1 Face Recognition System

Face recognition system will be used as a security feature to control the robot and opening it for accessing its contents. This feature can be used by both the hotel staff and the desired customer. Facial recognition based on OpenCV Python can be implemented. A digital camera will be used in this case. The user’s face is captured, and different characteristics of his/her face are extracted. These characteristics include all the parts of the user’s face which are measured, and then these characteristics are stored in a database. So, when that user wants to access anything, his/her face is captured by the camera, and then the characteristics are extracted and compared with the one stored in the database for recognizing him/her (Fig. 1).

We have used Raspberry Pi 5 megapixel Camera Board Module for capturing the user’s face and a servo motor for opening and closing the vault. Once the customer registers his/her name in database when he/she books a hotel room, the face samples of that customer will be updated in the robot database. So, when he/she orders a room service, the robot will reach the destination (outside the customer’s room), but its vault will open only when face is recognized by it using its digital video camera. Then the customer can take the items and close the vault.

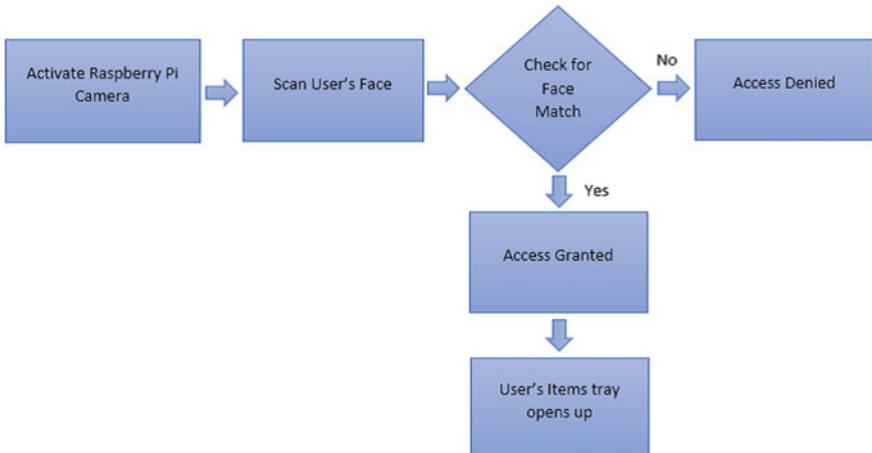


Fig. 1 Flowchart of the face recognition system

3.2 *Servo-Based Locking System*

Servo motor is the basic mechanism for items safety in this project. According to the requirement, other motors can be used for locking system. It works on voltage range of 4.8–6 V.

3.3 *Robot Commanding System*

A Matrix 4×3 Keypad has 12 buttons that are arranged in a telephone-line 4×3 grid type matrix. It requires only 7 microcontroller pin connections. It is flexible and can be attached on any surface. This keypad is used to command the robot to travel to a specific room (Fig. 2).

3.4 *Travelling System*

The motor used is a 30 RPM motor. It works on 12 V DC, its stall value is 41 kg/cm and its rated torque is 10 kg/cm. In this system, four wheels, each having diameter of 100 mm are used. The motors are connected only to the front wheels. This system is controlled by the Raspberry Pi (Figs. 3 and 4).

Fig. 2 Matrix 4×3 Keypad



Fig. 3 DC motor used in the robot



Fig. 4 Wheel used in the robot



4 Block Diagram and Working Principle

4.1 Face Recognition System

Facial Recognition is a technology which is used for unique identification of any person based on his/her facial characteristics. It comes under the field of biometric artificial intelligence.

In facial recognition, the user's face is captured and different characteristics of his/her face are extracted. These characteristics include all the parts of the user's face which are measured and then stored in a database. So, when that user wants to access anything, his/her face is captured by the camera, and the characteristics are compared with the one stored in the database for recognizing him/her.

There are other methods of identification, like fingerprints, which have more accuracy as compared to face recognition, but face recognition is getting a lot of limelight in researches that is because it does not require any physical contact (like fingerprint scans) and this is the way humans identify each other, which makes robots more human-like. In our case, we are using Open Source Computer Vision (OpenCV) software with help of Python language for face recognition. Using the Haar cascade classifiers in OpenCV, we can distil out the facial features of a person and store it as a NumPy array. OpenCV uses a Local Binary Patterns Histogram face recognizer to compare the trained images of a person with the live picture captured from the camera to predict the face of a person [4]. Facial recognition involves three steps:

- Collecting facial data: The person's face is captured in this case for identification. All the facial characteristics are also extracted.
- Training the facial recognition system: Facial characteristics of different people along with their respective names are fed to the facial recognition system for training it.
- Identification: In this case, new faces of the person whose face is trained is fed to the system to find out if his/her face is getting recognized [5] (Fig. 5).



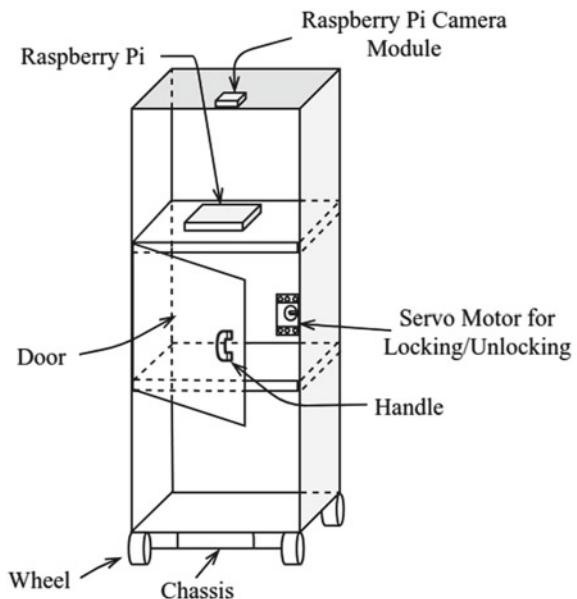
Fig. 5 Block diagram of the face recognition system

4.2 Servo-Based Locking System

Servo motor is used for locking the vault door for security. It is a kind of motor that allows us to precisely control its angular/linear position, velocity and acceleration. The working principle is as follows:

- When robot starts its journey from initial position moving towards specific hotel room, throughout the journey the servo motor is in one position such that the vault door is closed.
- Once the robot reaches its desired position and the customer's face is recognized, the servo motor moves and its blades position changes in such a way that the locked door is open [6] (Fig. 6).

Fig. 6 Pictorial representation of the robot



4.3 Robot Commanding System

In order to service the different rooms present in the hotel, the keypad will act as a commanding component used to instruct the robot which path to take to the concerned room. The numbers on the keypad can be used to program the room service robot based on the room numbers of the hotel rooms. For each and every room number, a separate path is defined to travel to the room according to the room code or room number inserted into the keypad. Once the microcontroller receives this room number as an input, it will travel towards the designated customer's room.

Using the keypad, we can make various combinations of the room numbers and hence can be used for multiple rooms in a hotel. This commanding system is only used for the hotel staff person, and upon giving the command, the robot will initiate into its travelling system.

4.4 Travelling System

Distance from initial position to every room will be a fixed value, so accordingly, for each room the path will be defined. The distance travelled by the robot will be an integral multiple of the circumference of the wheel. This algorithm is called distance calculation algorithm [7, 8]. In future, we can also use A* algorithm for effectively traversing in more than one room in shortest time possible. In A* algorithm, the shortest distance is calculated at each instance [9, 10]. We are using Johnson 30 RPM motor for this system (Fig. 7).

Two DC motors of the type as shown in Fig. 3 are attached to the two rear wheels of the robot. The wheels are used for moving the robot in forward and reverse directions. They can also be used for turning the robot. For example, if we want to turn the robot in the right direction, stop the left motor and only rotate the right motor. Vice versa is done for turning the robot in the left direction. For reversing the motor direction, opposite supply is given. L293D Motor IC is used for controlling the motors.

Four wheels of the type as shown in Fig. 4 are used in the robot. The two wheels at the front are dummy wheels, which are not connected to the motor. The two wheels at the back are connected to the motor.

The wheel size and strength are enough to bear the weight of the complete robot structure.



Fig. 7 Block diagram of the robot travelling system

Fig. 8 Frontal view of the actual robot



5 Actual Robot Structure

See Figs. 8 and 9.

6 Conclusion

There can be numerous applications of these types of robots, right from hotels to companion bots that can be used in day-to-day life. Applications of room service robots can extend in rural areas where there is less manpower, and these robots can help in the hospitality services for the hotels. This service of robot can also be extended to field like medical, retail after certain modifications. This robot can become a popular product in hotel industry if more user-friendly features are added in it. If new age technologies like virtual reality, artificial intelligence, etc. are integrated into this robot, it can become a popular product [11].

Fig. 9 Top view of the actual robot



References

1. Arkin R, Balch T (1970) Aura: principles and practice in review. *J Exp Theor Artif Intell* 9:02
2. Abdel-Aleem M, Shawky R, Aziz WM (2015) Evaluating the attendants' performance of guest room service department: applied to four-star hotels in Cairo. *J Fac Tourism Hotels* 9:69–76
3. Tussyadiah I, Park S (2018) Consumer evaluation of hotel service robots, pp 308–320
4. Faruqui N (2017) Open source computer vision for beginners, dated 26 Aug 2017
5. Wilson PI, Fernandez J (2006) Facial feature detection using Haar classifiers. *J Comput Sci Coll* 21(4):127–133
6. Koselka H, Wallach B, Gollaher D (2007) Autonomous personal service robot. US Patent 7,228,203, 5 June
7. Sankhe P, Rodrigues E (2018) Smart backpack. In: 3rd international conference for convergence in technology (I2CT), Pune, pp 1–4
8. Kumar GS, Shravan BVS, Gole H, Barve P, Ravikumar L (2011) Path planning algorithms: a comparative study (December)
9. Khemani D (2013) A first course in artificial intelligence
10. Deepa SN, Sivanandam SN (2011) Principles of soft computing. Wiley, India
11. Boatwright P, Cagan JM, Vogel CM (2013) The design of things to come: how ordinary people create extraordinary products. Tim Moore

Chapter 4

Comparative Analysis for an Optimized Data-Driven System



Chinmay Pophale, Ankit Dani, Aditya Gutte, Brijesh Choudhary and Vandana Jagtap

1 Introduction

This paper presents a deep insight into the development and management of large data-based systems through the use of various technologies in the market and also suggests the implementation of methodologies through a case study.

The performance metrics of any data-driven system with an interface depend on the selection of the database to be used which further is affected by the type of data to be handled by the system, i.e. structured or unstructured. The capacity and features of the web frameworks currently in market including Django, Laravel, and Flask amongst others play an important role and so does their use alongside communication protocols like REST and SOAP.

A real-life application based on above-mentioned techniques is discussed. Comparative analysis with considerable improvement over existing systems is depicted.

C. Pophale (✉) · A. Dani · A. Gutte · B. Choudhary · V. Jagtap
Department of Computer Engineering, Maharashtra Institute of Technology, Pune, India
e-mail: chinmay2997@gmail.com

A. Dani
e-mail: ankitdani1997@gmail.com

A. Gutte
e-mail: adityagutte@gmail.com

B. Choudhary
e-mail: brijesh.choudhary7@gmail.com

V. Jagtap
e-mail: vandana.jagtap@mitpune.edu.in

2 Literature Survey

The analysis of the various relevant frameworks and technologies leads to determine that each tool handles files of diverse settings; in the same way, it could be observed that each of the tools offers different types of useful features to facilitate the programmer, in his work environment.

The criteria for the selection of a database were studied thoroughly pointing out the features of various available databases [1]. Django, Flask, and Laravel were the identified frameworks suitable for handling web applications of large data systems [2]. A critical review of most widely used web communication protocols like REST and SOAP was performed [3]. For providing requests and response functionality, latest networking services like Google Volley and Retrofit were compared [4].

3 Comparative Analysis

3.1 Database

For any system architecture, the database is its backbone. The database performance is the most important metrics to be considered. So selection of the database which suits your system is crucial for its efficient working. Various parameters to be considered while choosing a database are schemas, structuring, data access time, etc. The detailed study of many databases has led to the following conclusions about the same.

SQL and No SQL

Structured query language (SQL) is the basis of relational database management system (RDBMS). Few databases based on SQL are MySQL, Oracle, MS SQL Server, IBM DB2, Microsoft Access, etc. SQL has an upperhand when high speeds are required for data retrieval. Large amount of data which is in a standard format is best suited for SQL databases [5]. Also not much of coding is required while handling the data to and fro from the database. This makes database management and indexing process way simpler.

NoSQL databases are unstructured databases with no fixed schemas attached to them. These are more scalable, more flexible and have a great extent of agility [6]. These provide unrestricted management of structured, semi-structured, and unstructured data [7].

3.2 Web Frameworks

In any large-scale data-driven project, a web framework plays a pivotal role in the development of different web applications like web resources, web APIs, and web

services. A web framework in simple terms facilitates a reliable method to build and deploy these applications on the World Wide Web.

Django, Flask, and Laravel

Django is a high-level web framework based on Python that encourages rapid development of websites. It handles much of the perks of web development, so there is more focus on writing the application without any need to reinvent the wheel [8].

On the other side, Flask is a lightweight and minimalist web framework. It lacks some of the predefined features provided by Django. But it helps developers to keep the core of a web application simple and extensible. Unlike Django, Flask is more difficult for users to handle administrative tasks as the former gives a ready to use admin functionality. Django also contains a bootstrapping tool called Django-admin. Django-admin enables users to start applications without any external requirements. Django also provides a built-in object-relational mapping (ORM) system to facilitate multiple kinds of database connections which is lacking in Flask [9].

Laravel is a PHP developed web framework based on model view controller (MVC) compared to Django's model view template (MVT) design [10]. Laravel functions on significantly slower speeds than Django since Python is a fast language compared to PHP. Django also aids developers avoid the mistakes of web development and implement efficient security measures. While Laravel covers security yet it does not come close to Django's measures. That is the reason why, for example, NASA uses Django for their web portals and applications.

3.3 Web API Services: REST and SOAP

Representational state transfer (REST) and simple object access protocol (SOAP) are primarily used communication protocols [11]. REST functions through a solitary interface to access resources while SOAP exposes components of application as services instead of data. REST allows more variety of data formats and SOAP only works with XML [12]. Compared to SOAP, REST is significantly faster and uses less bandwidth. It also offers better support for browser clients [13].

3.4 Networking Services: Volley and Retrofit

Volley is a networking library which inculcates helpful features like synchronous and asynchronous requests, priority handling, multiple and ordered requests, JSON parsing, and of course caching advantages [14]. Retrofit is a REST client library for Android, through which a user can make easy to handle interfaces. Retrofit performs async and sync requests with automatic JSON parsing. Unlike Volley, it does not support caching. Also, Volley allows the retrying of requests along with modified timeouts automatically which Retrofit does not support. Volley supports inbuilt image loading features while Retrofit needs third-party libraries [4]. Retrofit, as an advantage, supports more varied output formats than Volley.

4 Case Study

The following case study highlights the development of an optimized architecture for ASTROSAT data quality reports (DQRs). ASTROSAT is India's first dedicated multi-wavelength space observatory which was launched on a PSLV-XL on 28th Sep 2015. The architecture works as a backend for an Android application which was created to fetch and display the above-mentioned DQRs.

4.1 MySQL Database

The project requires handling of large sets of structured astronomical data for which an SQL database (MySQL) proved to be the best for use when it comes providing effective data insights for data analysis, minimal latency in response time and facilitates simpler database management.

4.2 Django Framework

The architecture described in the case study makes use of the distributed system functionality of Django web framework which involves the creation of a separate “app” for each feature of the app. Use of Django significantly enhances the scalability, ease of use, robustness, and security characteristics of the proposed architecture (Fig. 1).

4.3 Rest API

In the implemented RESTful API, endpoints (URLs) define the structure of the API and how end-users access data from the app. REST framework is used as a base for serialization which allows complex data such as query sets and model instances to be converted to native Python data types that are then easily rendered into JSON.

4.4 Google Volley

Google Volley adds some powerful options and is a ton quicker than other alternatives like AsyncTask and Retrofit. In this architecture, instead of creating a new instance of RequestQueue every time, the singleton pattern of creating a single instance of RequestQueue throughout the application is followed. This results in faster JSON retrieval speed. Caching capabilities of the Volley framework helps the app to pre-fetch data for faster loading.

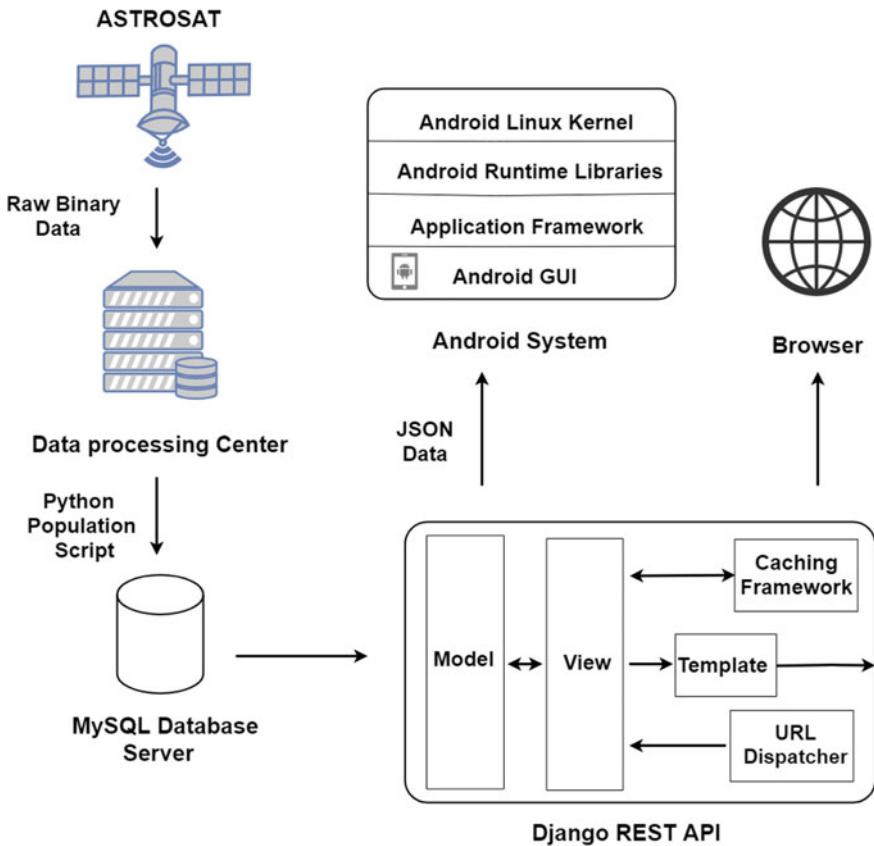


Fig. 1 Proposed system for ASTROSAT DQRs featuring a database server with an efficient Django REST API

4.5 Results

The following representation proves that the correct combination of technologies compared and analysed above is beneficial towards the effective development of any large-scale data-driven system. This case study implementation was compared with an existing system which lacked the use of an application server, web framework, services, and data caching. The new proposed system achieves 97.2% efficiency over the existing system. The above graph clearly demonstrates the improvement over the existing architecture shows exponential increase in response time whereas the proposed architecture gives us a linear minuscule increase. The proposed system tries to minimize the scaling time factor of the directly proportional relation.

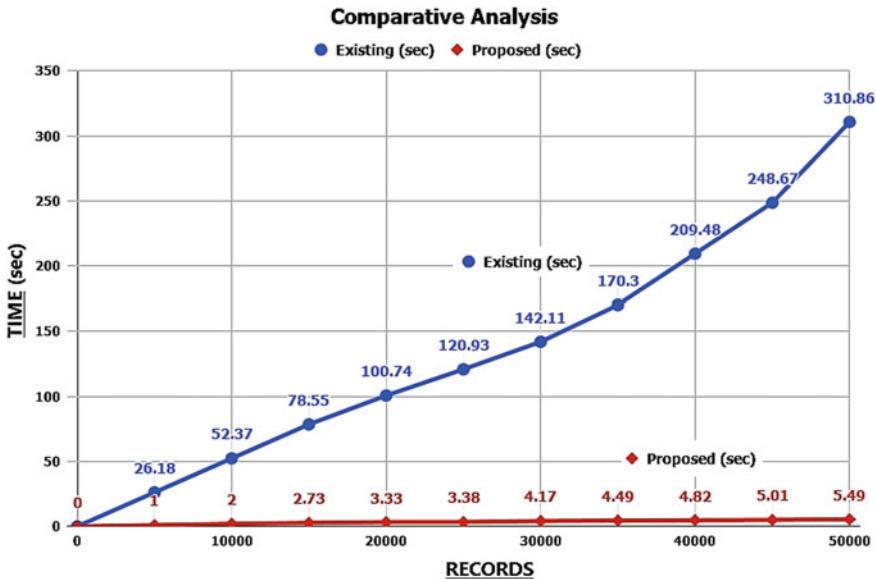


Fig. 2 Comparative analysis of existing system and proposed system

5 Conclusion and Future Scope

In this paper, an architecture for the ASTROSAT DQRs employing a web service based on Django web framework and RESTful API; application server consisting of a relational SQL database; a client-side user interface comprising of a mobile interface is propositioned that optimizes the storage and fetching mechanism of the DQRs. It enhances the speed of the system significantly. The load balancing and dynamic data fetching process employed in the Django and RESTful API allows the mobile application to maintain a lightweight architecture. It also makes the data considerably ubiquitous compared to the existing system owing to the mobile user interface it provides to its users. These implementations have resulted in establishment of a highly scalable and robust system architecture (Fig. 2).

References

- French JC, Powell AL (1999) Metrics for evaluating database selection techniques. In: Proceedings. Tenth international workshop on database and expert systems applications. DEXA 99, Florence, Italy, pp 726–730
- Valarezo R, Guarda T (2018) Comparative analysis of the laravel and codeigniter frameworks: for the implementation of the management system of merit and opposition competitions in the State University Península de Santa Elena. In: 2018 13th Iberian conference on information systems and technologies (CISTI), Caceres, pp 1–6

3. Su H, Cheng B, Wu T, Li X (2011) Mashup service release based on SOAP and REST. In: Proceedings of 2011 international conference on computer science and network technology, Harbin, pp 1091–1095
4. Shulin Y, Jieping H (2014) Research and implementation of Web Services in Android network communication framework Volley. In: 2014 11th international conference on service systems and service management (ICSSSM), Beijing, pp 1–3
5. Patil MM, Hanni A, Tejeshwar CH, Patil P (2017) A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retrieval operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages. In: 2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC), Palladam, pp 325–330
6. Győrődi C, Győrődi R, Pecherle G, Olah A (2015) A comparative study: MongoDB vs. MySQL. In: 2015 13th international conference on engineering of modern electric systems (EMES), Oradea, pp 1–6
7. Rautmare S, Bhalerao DM (2016) MySQL and NoSQL database comparison for IoT application. In: 2016 IEEE international conference on advances in computer applications (ICACA), Coimbatore, pp 235–238
8. Plekhanova J (2009) Evaluating web development frameworks: Django, Ruby on Rails and CakePHP. Institute for Business and Information Technology
9. Forcier J, Bissex P, Chun WJ (2008) Python web development with Django. Addison-Wesley Professional
10. Chou J, Chen L, Ding H, Tu J, Xu B (2013) A method of optimizing Django based on greedy strategy. In: 2013 10th web information system and application conference
11. Rubio D (2017) REST services with Django. In: Beginning Django. Apress, Berkeley, CA
12. Li L, Chou W (2015) Designing large scale REST APIs based on REST chart. In: 2015 IEEE international conference on web services, New York, NY, pp 631–638
13. Li L, Chou W, Zhou W, Luo M (2016) Design patterns and extensibility of REST API for networking applications. IEEE Trans Netw Serv Manage 13(1):154–167
14. Lachgar M, Benouda H, Elfirdoussi S (2018) Android REST APIs: Volley vs Retrofit. In: 2018 international symposium on advanced electrical and communication technologies (ISAECT), Rabat, Morocco, pp 1–6

Chapter 5

Fake Email and Spam Detection: User Feedback with Naives Bayesian Approach



Ayushi Gupta, Sushila Palwe and Devyani Keskar

1 Introduction

In neoteric times, amongst the quickest and majorly reasonable methods of intercommunication in the society for interaction and conversing with parents and friends and for exchanging files, data, etc., emails are being used. They are segregated into distinct categories like ham (solicited) or spam (unsolicited). The classification which deals with legal, authorized and verified emails falls under ham mails, whereas on the other hand, the category which accounts to fake, unwanted, useless and pointless mails comes under spam mails. Thus, these unwanted are causing major problems and dire consequences. Amidst these, the rooted spam emails are crucial as they embezzle storage space, generate time wastage, induce harmful malware and crucially affect phishing. Issues like resource consumption, transmission bandwidth costs, user's time wastage, etc., cost billions of dollars. Segregation of spam mails at real time is deployed and performed by Naive Bayes method. The major work carried out by a classifier is to recognize the unwanted, fake or harmful mails which are sent to the user and characterize it as unsolicited (spam) mail.

A. Gupta (✉) · D. Keskar

Department of Computer Engineering, MITCOE, Pune, India

e-mail: ayushimg9@gmail.com

D. Keskar

e-mail: devyani.keskar@gmail.com

S. Palwe

School of CET, MITWPU, Pune, India

e-mail: sushila.palwe@mitwpu.edu.in

2 Literature Survey

Email spam [1] is an issue which is being constantly addressed and researched to improve the user experience and utilize the resources to user benefit. Segregating the emails into spam and ham is achieved by various machine learning algorithms like Naïves Bayesian, Swarm Particle Optimization, Support Vector Machine, etc.

In Naïves Bayesian approach, the classification was done on the basis of taking into account the frequency of particular word corresponding to the event like ham or spam in which they occur [2]. The basic approach limited only to one word and one event notation, which was further overcome by the modified approach taking into consideration of multiple events and reducing the computing power and gives faster results. Bayesian method is a very simple and highly scalable approach which helps to get optimized results. This method gave accurate results but had a drawback of not considering the user feedback into consideration of event. User feedback includes the choices made by the user to filter out spam or ham from their inboxes based on their preferences.

3 Methodology

3.1 *Introduction*

Amongst the category of uncomplicated “probabilistic classifiers,” Naïves Bayes classifier is the one which is based upon naïve assumptions that features demonstrate individualistic nature amid each other.

Naïve Bayes is a comprehensible approach for constructing classification models which allots class labels to sample problem that exhibits a feature value vector. Class labels or characteristics are extracted from a delimited set. Simple Naïve Bayesian approach takes into consideration single event and single feature instance which is contrary to modified Naïve Bayesian approach. Training even with a small dataset is achievable by applying Bayes classifier. Computation of parameters through “method of maximum likelihood” is maneuvered for Naïve Bayes models in diverse practical applications.

3.2 *Probabilistic Model*

Naïve Bayes is based upon conditional probability framework which is described for a given problem instance which is to be classified based on certain characteristics conveyed by a vector m where m is described as

$$\text{Vec } m = (m_1, \dots, m_n)$$

which denotes and depicts n features that are presumed to be independent variables, Assignment of instance probabilities is accomplished as

$$p(C_k|m_1, \dots, m_n)$$

where K denotes each possible outcomes or classes C_k .

Conditional probability can be quoted using Bayes's Theorem by

$$p[C_k|m] = \frac{p(C_k) * p[m|C_k]}{p(m)}$$

The following equation can be depicted as

$$\text{posterior} = \left(\frac{\text{prior} * \text{likelihood}}{\text{evidence}} \right)$$

The joint probability model is analogous to numerator which is “prior x likelihood”

$$p(C_k, m_1, \dots, m_n)$$

where m is the events taken into account for calculating probability.

This expression can be recapitulated by taking the help of chain rule which is used for repeated applications of the statement of conditional probability:

$$\begin{aligned} p(C_k, m_1, \dots, m_n) &= p(m_1, \dots, m_n, C_k) \\ &= p(m_1|m_2, \dots, m_n, C_k) * p(m_2, \dots, m_n, C_k) \\ &= p(m_1|m_2, \dots, m_n, C_k) * p(m_2|m_3, \dots, m_n, C_k) * p(m_3, \dots, m_n, C_k) \\ &= \dots \\ &= p(m_1|m_2, \dots, m_n, C_k) * p(m_2|m_3, \dots, m_n, C_k) \dots * p(m_{n-1}|m_n, C_k) \\ &\quad * p(m_n|C_k) * p(C_k) \end{aligned}$$

4 Database Collection/Preparation

In this project, we have taken an email dataset consisting of 5730 emails containing the body and subject of various emails which are bifurcated into two classes—0 and 1 for ham (solicited email) and spam (unsolicited email), respectively.

From the 5730 emails, 4358 emails are ham, and 1372 emails are spam. Table 1 shows the calculation of frequencies of spam words like—discount, flash, #, www, etc., and finds the probability of each word.

We can deduce from Table 1, the information about particular words present in solicited emails (ham mails) and unsolicited emails (spam mails), w_{ti} denotes the favorable occurrence of event in spam (S) or ham (H).

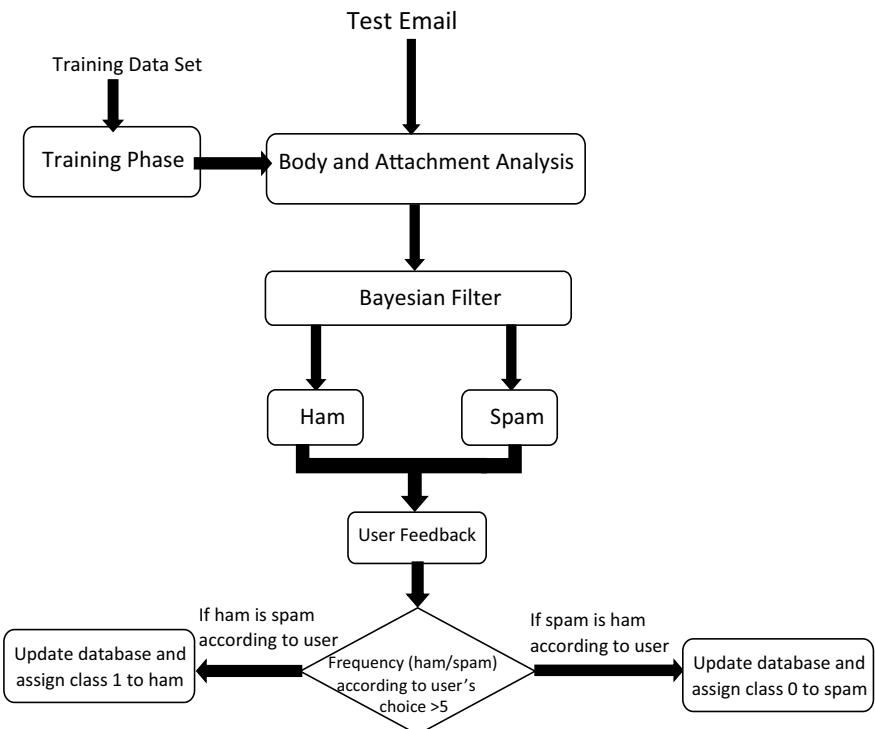
Table 1 Probability table for occurrence of words in email

Word	Present in solicited email	Present in unsolicited email	$P(wt_i S)$	$P(wt_i H)$
Discount	52	70	0.0379	0.016
Flash	13	18	0.419	0.58
#	108	160	0.402	0.597
www	242	482	0.334	0.665

In our experimentation, 1139 emails are selected randomly for the training dataset and 4591 mails for testing purpose. Training dataset is analyzed to generate a confusion matrix to compute the number of true positives, false positive, true negatives and false negatives for given dataset.

In the user feedback methodology after a frequency of five spam marked as ham or vice versa, we update the class of the corresponding dataset with 1 or 0 according to the user's response. This helps us in segregating the emails with better accuracy.

Following Fig. 1 is the flow diagram for the method adopted.

**Fig. 1** Flow diagram for user feedback with Naïves Bayesian approach

5 Result Discussion

After analyzing the testing data, Fig. 2 confusion matrix is generated. Following results are drawn from the matrix in Figs. 2 and 3.

The accuracy for the following classifier is $[(TP + TN)/Total]$ which is 0.9903 (99.03%).

The misclassification/error rate is $[(FP + FN)/Total]$ which is 0.00965 (0.965%).

The true positive rate is $[TP/\text{actual spam}]$ which is 0.9927 (99.27%).

The false positive [3] rate is $[TN/\text{actual spam}]$ which is 0.0104 (1.04%).

The true negative rate is $[TN/\text{actual ham}]$ which is 0.9895 (98.95%).

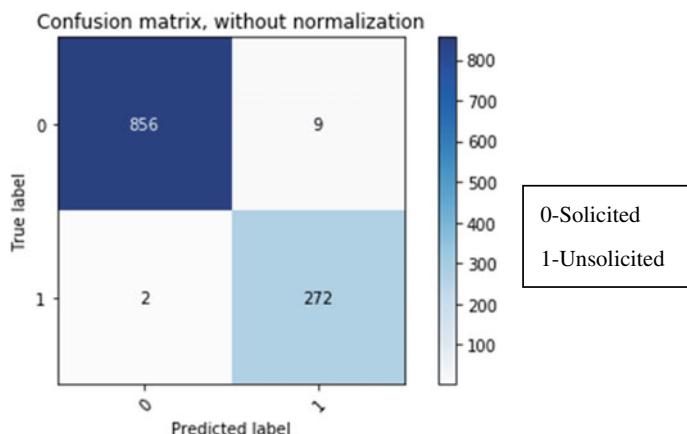


Fig. 2 Confusion matrix for ham and spam training email dataset

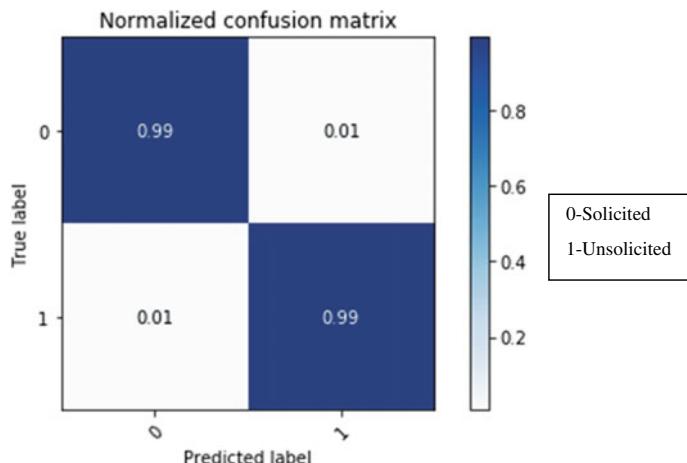


Fig. 3 Normalized confusion matrix for ham and spam training email dataset

Precision is [TP/predicted spam] which is 0.9679 (96.79%).

The precision without user feedback on the testing dataset is 96.79%, and after inclusion of user feedback, the precision increased to 98.7% which is an improved result by 2%.

The false positive rate ideally should be as low as possible. Nowadays, spammers are using this approach to increase Bayesian Poisoning which reduces the accuracy of the filter. Our aim is to increase the true positive rate which can be done through the approach suggested.

User feedback plays a major role in driving the filter accuracy to give optimum result. After the user marks a spam email from a particular sender to be ham for himself/herself, after a counter frequency of five and above, the email from the particular sender is updated in the database and is not filtered as spam further for the user.

6 Future Scope

In this approach, we have discussed the classification on a dataset of 5730 emails divided into classes 0 and 1 corresponding to ham and spam, respectively, and user feedback and preferences were taken into consideration to deduct those mails from spam category. For extending the accuracy, we can also maintain a lookup table for the classifications done, so that the next time any such case arises, the computation is faster and more optimistic. Sender's email addresses and URL links from the body of the email can also be analyzed to reduce the efforts on the earlier levels of filtering of emails.

7 Conclusions

Fake email and spam detection are an important topic of research and network security. Various machine learning techniques have been applied for solving it, out of which Naives Bayesian classifier approach is a simple and probabilistic approach. It can be trained using a small dataset. The precision of filtering and classifying emails through this classifier is 96.79% by experimentation on the dataset carried out. Inclusion of user feedback has increased the optimization of filtering out the emails by 2% which is an advantage for the user. User feedback improves the accuracy of filtering of spam emails by Bayesian Classification on given dataset by taking into consideration user preferences and generated the confusion matrix and results related to it.

Acknowledgements I would like to thank Prof Sushila Palwe for her constant guidance and support during my research.

References

1. Vijayasekaran G, Rosi S (2018) Spam and email detection in big data platform using naives bayesian classifier. IJCSMC 7(4):53–58
2. Sun T. Spam filtering based on Naive Bayes classification
3. Cosoi AC (2008) A false positive safe neural network; the followers of the Anstrim Waves. In: Proceedings of MIT spam conference

Chapter 6

C-ASFT: Convolutional Neural Networks-Based Anti-spam Filtering Technique



Sunita Dhavale

1 Introduction

E-mail system has been one of the most widespread and essential form of communication in our day-to-day life for business and personal purposes. E-mails containing unwanted/unsolicited messages sent by unknown sender in massive manner are called spam e-mails. Today spam mails have become a dominant problem for any organization due to its voluminous, offensive and annoying nature. Besides commercial purposes to advertise various products, they are also used to send bogus offers or fraud services that could lead to cybercrimes. Organizational employees will waste their valuable time in analyzing spam mails manually from their mail inbox [1].

Spam filters serve as one of the technological solutions against spam which has the capability of differentiating the mails into spam and non-spam (ham). Analyzing spams at server-side reduces network load and computational complexity at client-side. However, incorrect classifications may end up in labeling legitimate e-mails as spam [1]. Analyzing spams at client-side may offer more personalized low-cost spam management [1]. Many machine learning classifiers like Naïve Bayesian (NB), k-Nearest Neighbor (k-NN), Neural Network (NN), C4.5 Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM) have been explored in case of spam detection (see, e.g., [2–6]). In these examples, spam classifier is trained using training dataset containing examples belonging to both spam and legitimate classes. The trained classifier is tested against labeled test dataset. The performance of classifier is evaluated against the parameters that measure the degree of misclassification [1].

This paper proposes a Convolutional Neural Networks (CNN)-based server-side solution called C-ASFT (CNN-based Anti-Spam Filtering Technique). Instead of

S. Dhavale (✉)

Department of Computer Science and Engineering, Defence Institute of Advanced Technology,
Girinagar, Pune 411038, India

e-mail: sunitadhavale75@rediffmail.com

utilizing handcrafted image features [16], C-ASFT employs CNN, a deep learning approach for learning text features automatically for spam filtering. The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 provides the details of proposed system. Section 4 contains the experimental results and analysis. Conclusion of the work is stated in Sect. 5.

2 Related Work

In [2], authors have analyzed different versions of Naive Bayes on Enron-Spam dataset. They made this new dataset publicly available. Compared to other existing datasets like Ling-Spam [7] or SpamAssassin [4], Enron-Spam datasets maintain the temporal order of the messages. After series of experimentations, the authors found that both Flexible Bayes and Multinomial NB with Boolean attributes provide good performance for spam filtering tasks [2].

In [8], the authors investigate the performance of spam filtering based on adaptive statistical data compression models. These fast incrementally updateable probabilistic text classifiers modeled messages as sequences. Peng et al. [9] proposed NB classifier along with statistical language models for capturing word dependencies. The authors found that using characters instead of words for training the language models will enhance the spam filter detection accuracy. Authors in [10] explored semi-supervised learning methods for spam filtering and found that detection accuracy enhances.

Recently, with the advent of GPUs and server-side hardware, deep learning techniques have been successfully applied in the field of text/image/video classification domain [11]. Features learnt by deep neural net structure have shown their great potential in various classification tasks instead of exploring different handcrafted features on trial and error basis [11]. The authors in [12] utilized adaptive feature vectors according to different lengths of e-mails using recurrent neural networks (RNNs) for spam detection. In [13], spam detection on Twitter having short text and high variability in the language used in social media using CNN and Long Short-Term Neural Network (LSTM) is focused. The authors used the knowledge-bases such as WordNet and ConceptNet is used to extract the semantic information contained in the words to improve the performance of detection.

3 Proposed Model

We are interested in finding a CNN-based model $f(\cdot)$, that maps messages represented as m-dimensional vectors $x \in R^m$ from some distribution D to predicted class labels $f(x) \in \{+1, -1\}$ for spam and ham (not spam), respectively, such that the predicted class label agrees with y , the true class of the message. The proposed scheme consists of following modules as shown in Fig. 1.

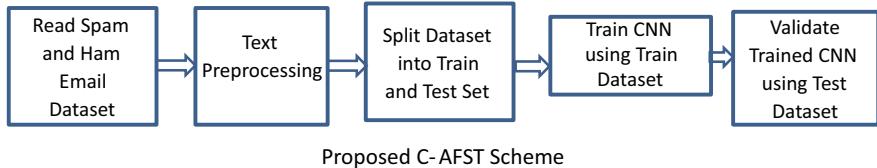


Fig. 1 Proposed C-ASFT scheme

3.1 Text Data Preprocessing

In this phase, both spam and ham text messages containing total 32,766 files from standard Enron dataset [2] are read. During preprocessing, (1) remove numeric texts/empty texts; (2) remove punctuation/stop words from texts; (3) convert words to lower case; and (4) apply stemming.

Raw text data must be encoded as numbers before giving to any deep learning models as input. Keras, a deep learning library, Tokenizer Class is constructed and fit on raw text data to get 155,776 unique tokens. Unique sequence numbers are assigned to each unique word in the input text data and the whole text data is replaced by the sequence numbers. For example, if the spam text is “prize for click,” then it will be replaced by code = [1058, 7, 215] where 1058 sequence ID refers to the word “prize.” Next, padding is used to ensure that all encoded text sequences in a list have the same length.

Create embedding matrix for all unique tokens using GloVe (Global Vectors for Word Representation) [14], standard dictionary based on factorizing a matrix of word co-occurrence statistics. GloVe maps semantic meaning of word into a geometric/embedding space. GloVe associates a numeric vector to every word in a dictionary in such a way that the distance between any two vectors denotes the semantic relationship between the two associated words. For example, “apple” and “dinner” are related words, hence they will be embedded close to each other; however, “office” and “tiger” are words that are semantically quite different, so they will be embedded too far from each other in geometric space.

If word is not part of GloVe, then set the matrix values to zero. GloVe consists of 400,000 well-known words with each word represented or encoded in a number vector format. For each word in dataset’s vocabulary, we check if it is on GloVe’s vocabulary. If so then, use its pre-trained word vector. The GloVe has embedding vector sizes: 50, 100, 200, and 300 dimensions. We have chosen the 100-dimensional vector size. These pre-trained word embeddings are used to set the weights of embedding layer of CNN. Embedding layer provides dictionary mapping integer indices reflecting dense vectors. For a given input integer, it looks up for associated vectors. The dataset is then split into train and test dataset for further training and testing efficacy of CNN model.

Table 1 CNN architecture

Layer (Type)	Output shape	Parameters
input_1 (Input Layer)	1000	0
Embedding	1000, 100	15,577,600
Conv1D	996, 128	64,128
max_pooling1D	199, 128	0
Conv1D_1	195, 128	82,048
max_pooling1D_1	39,128	0
Conv1D_2	35,128	82,048
Global_max_pooling1D	128	0
dense	128	16,512
dense_1	(None, 2)	258
Total		15,822,594
Trainable		244,994
Non-trainable		15,577,600

3.2 CNN Structure

Table 1 gives a summary of CNN architecture used for training.

Here, the input layer is followed by embedding layer and three convolutional layers with RELU activations. Each convolutional layer consists of 128 filters with kernel size of 5×5 . 1 Dimensional CNN layers learn the spatial structure or invariant features present in the sequence of words in the input mail text data. Each convolutional layer is followed by max pooling stage over a 5×5 window. The final convolutional layer is followed by global max pooling layer and two fully connected layers one containing 128 nodes followed by RELU activation and second containing 2 nodes followed by Softmax function at the end. A dropout a regularization technique with value of 0.5 used to reduce over-fitting [11]. Final fully connected layer classifies the text messages into spam and ham classes. The CNN model is compiled by setting loss parameter to categorical cross entropy and selecting RMSprop optimizer. The model is evaluated against final dataset containing original training and unseen test dataset.

4 Experimental Results and Analysis

4.1 Experimental Setup

For experimentation, we used both TREC 2007 spam corpus [15] and Enron-Spam [2] datasets. The text e-mail messages are preprocessed before giving to the CNN model for training. Batch size of 16 is used for training each epoch and model is

trained for total 10 epochs only. Total messages are split in ratio of 80 percent training data and 20 percent testing data. The trained CNN model is evaluated against the testing dataset on laptop with containing Core-i7 CPU processor, 16 GB RAM, and 6 GB NVIDIA GPU card. The algorithm is implemented in spyder anaconda python environment using Keras, Sklearn, Opencv, and Tensorflow deep learning libraries.

TREC 2007 spam corpus [15]: This corpus contains 75,419 messages: 25,220 ham and 50,199 spam. It can be downloaded from the link, <http://plg.uwaterloo.ca/~gvcormac/spam/>.

Enron-Spam datasets [2]: This corpus contains total 88,792 preprocessed messages: 20,170 spam messages and 16,545 ham.

4.2 Results and Analysis

Both training and testing scores along with accuracy are used for evaluation of proposed C-ASFT model with respect to true data labels. As we feed the model a batch of inputs, it will return the mean loss. If model has lower loss at test time, means it exhibits lower prediction error, and hence, accuracy will be higher. The performance of the model with filter size of 128 is assessed. Applying activation function after each convolution layer selects the invariant spam e-mail text features. The non-linear activation function RELU helps in limiting the vector values to specified range [0, 1]. RMSprop optimization method is used to adapt learning rate.

Table 2 shows the comparison of proposed C-ASFT with existing CNN-based text spam detection work [12]. Proposed C-ASFT shows better results compared to the previous one. Both testing and training score matches. The average accuracy is 98%. Figure 2 shows how the accuracy increases and loss decreases per epoch [13].

Table 3 shows the values of normalized confusion matrix.

False positive rate (FPR) is fraction of ham messages which are misclassified as spam messages. False negative rate (FNR) is fraction of spam messages that are misclassified as genuine messages by the spam filter. Both true positive rate (TPR) and true negative rates (TNR) are high while low FPR and low FNR. These values are essential, as no user will like that any legitimate e-mail is classified as spam and goes to spam folder. This may cause great loss for user if any important mail, e.g., related to his job offers, etc., goes undetected. Experimental simulations during training phases

Table 2 Comparison

Parameters	Proposed C-ASFT	[15] on PU dataset	[16] on Twitter dataset
Model	CNN	RNN	CNN + LSTM
Train score	0.002	Not given	Not given
Train accuracy	0.999	Not given	Not given
Test score	0.109	Not given	Not given
Test accuracy	0.980	0.969	0.950 (With RELU)

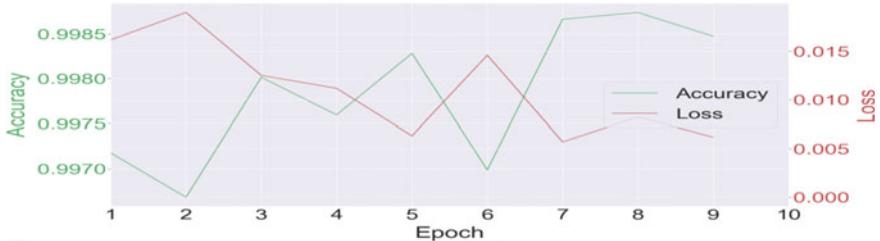


Fig. 2 Accuracy and loss per epoch

Table 3 Confusion matrix

True label	Predicted label	
	Ham	Spam
Ham	0.981	0.018
Spam	0.015	0.984

also show that using pre-trained word embeddings GloVe for initializing embedding layer weights achieves more than 90% accuracy in less number of epochs compared to that of using our model with embedding layer without initializing weight vectors.

5 Conclusions and Future Work

In this work, we proposed the deep learning-based server-side solution called C-ASFT (CNN-based Anti-Spam Filtering Technique). C-ASFT employs three 1D CNN layer model for effective text-based spam detection and filtering. The deep learning approach not only eliminates time-consuming extraction of handcrafted features but also provides accurate robust real-time server-side text-based spam filtering detection. Experimental simulations during training phases also show that using pre-trained word embeddings GloVe for initializing embedding layer weights achieves more than 96% accuracy in less number of epochs. This will save the time required for training the model. Once the e-mail tagged with spam/ham category, user input can also be learnt in case of misclassifications by applying reinforcement learning techniques along with maintaining white and black lists. Transfer learning can be applied at regular interval to make trained classifier more robust by accommodating newly unseen ham/spam mails. Further research can be carried out by exploring other deep learning algorithms in this text classification context and tuning CNN architecture parameters to enhance the accuracy further.

Acknowledgements We would like to thank NVIDIA Corporation for granting us TITAN V GPU for carrying out deep learning-based research work.

References

1. Dhavale S (2017) Advanced image-based spam detection and filtering techniques, 1st edn. IGI Global, USA
2. Metris V, Androutsopoulos I, Paliouras G (2006) Spam filtering with Naive Bayes—which Naive Bayes? In: Proceedings of the 3rd conference on email and anti-spam (CEAS 2006), Mountain View, CA, USA
3. Carreras X, Marquez L (2001) Boosting trees for anti-spam email filtering. In: 4th International conference on recent advances in natural language processing, Tzigov Chark, Bulgaria, pp 58–64 (2001)
4. Drucker HD, Wu D, Vapnik V (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10(5):1048–1054
5. Hidalgo JG (2002) Evaluating cost-sensitive unsolicited bulk email categorization. In: 17th ACM symposium on applied computing, pp 615–620
6. Kolcz A, Alspector J (2001) SVM-based filtering of e-mail spam with content-specific misclassification costs. In: Workshop on text mining, IEEE international conference on data mining, San Jose, California (2001)
7. Sakkis G, Androutsopoulos I, Paliouras G, Karkaletsis V, Spyropoulos C, Stamatopoulos P (2003) A memory-based approach to anti-spam filtering for mailing lists. Inf Retrieval 6(1):49–73
8. Bratko A, Cormack G, Bogdan F, Lynam T, Zupan B (2006) Spam filtering using statistical data compression models. J Mach Learn Res 7:2673–2698
9. Peng DS, Wang S (2004) Augmenting naive bayes classifiers with statistical language models. Inf Retrieval 7(3–4):317–345
10. Mojdeh M, Cormack GV (2008) Semi-supervised spam filtering: does it work? In: SIGIR’08, Singapore, 20–24 July 2008
11. Goodfellow I, Bengio Y, Courville A (2015) Deep learning. MIT Press, Cambridge
12. Gao Y, Mi G, Tan Y (2015) Variable length concentration based feature construction method for spam detection. In: International joint conference on neural networks (IJCNN). IEEE, pp 1–7
13. Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. Ann Math Artif Intell Springer Nature Switzerland AG 2019(85):21–44
14. Jeffrey P, Richard S, Christopher DM (2014) GloVe: global vectors for word representation, empirical methods in natural language processing (EMNLP), pp 1532–1543
15. Cormack GV (2007) TREC spam track overview. In: Sixteenth text retrieval conference (TREC-2007), Gaithersburg, MD, NIST (2007)
16. Francesco G, Carlo S (2008) Combining visual and textual features for filtering spam emails. In: 19th International conference on pattern recognition (ICPR), IEEE, 8–11 Dec 2008

Chapter 7

Cognitive Control of Robotic-Rehabilitation Device Using Emotiv EEG Headset



Neha Hooda, Ratan Das and Neelesh Kumar

1 Introduction

With the advent of technology, it has become possible to process brain signals in real time. Electroencephalography (EEG) is the non-invasive method of brain signal acquisition. Brain–computer interfacing (BCI) defines the technique of transferring and interpreting the neurological (brain) signals through non-neurological channels. Although it is a considerably old concept, the recent technological development has enabled the researchers to validate. The translation of thoughts to system control has been achieved in a plethora of ways in the past decade itself [1]. BCI is a combination of hardware and software unit that can be employed to work in two different ways. The first is communication via brain signals to control external interfaces, for example, light switch, fan, robots, etc. Based on this, second method uses brain-controlled operations to enhance neuroplasticity. The technique has been used for motor re-learning using focused attention and task repetition paradigms. It has enabled injured or paralyzed or amputated patients with the ability to operate a computer display (letters or virtual avatar), wheelchair, prosthetic arm or leg, robotic assistance and many more [2].

While working with a BCI device, the ease of use and cost plays an important factor [3]. Various BCI-based devices have been developed but the end-user applications

N. Hooda (✉) · R. Das · N. Kumar

Biomedical Instrumentation Unit, CSIR-Central Scientific Instruments Organisation, Chandigarh 160030, India

e-mail: getneha.hooda@gmail.com

R. Das

e-mail: ratans16@gmail.com

N. Kumar

e-mail: neel5278@gmail.com

Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India

are quite rare due to cost as well other constraints. Emotiv neuroheadset, a commercially available headset, has been found to be effective to use in real and virtual environments [4]. It has been used in variety of applications due to its ease of use and accessibility. The device effectively translated the brain signals to perform 3D object rotation in virtual space [5]. Various gaming platforms have incorporated Emotiv-based control for entertainment as well as medical purposes. Researchers have used different parameters evaluated from the EEG signals including task engagement [6–8], cognitive thoughts [5, 9] as well as motor imageries [10–12]. The technique of task engagement used the attention/focus or relaxation of the user as performance measure. Based on this, theta to beta ratio [7] or visual engagement index [6, 8] has been used as classification parameters. Notably, in order to obtain significant result in motor imagery control and classification, the positioning of headset has been changed by [11, 12]. It suggested tilting around the reference axis compared to its default placement, so as to cover the sensory motor areas defined by international 10–20 system. In addition, the neuroheadset has been used to control and navigate robotic systems [13–15]. The method has been extended for remote control of humanoid robots citing application in military, medicine or disaster management [16]. Also, [17] presented a hybrid BCI technique, combining P300 potential, steady state visually evoked (SSVEP) potential and event-related desynchronization (ERD) to solve multitask problem of object recognition, robot navigation and obstacle avoidance. In addition, the headset has been found to be effective for control of an external robotic arm [18, 19], wheelchair [20, 21], mobile phone [22], etc. Even though the application area is vast, the signals for detection mostly used facial expressions [15, 16, 18, 22], attention/engagement level [13] or sometimes inbuilt gyroscope [14, 20] for classification. When using cognitive imageries, the classification provided by Emotiv’s “black box” has been used [21, 22]. While some researchers have opened avenue for independent signal recognition and classification [11, 12, 17], efforts are still needed to incorporate different control strategies and real-time applications.

Current study aims to present a solution in this direction by directly accessing the Emotiv raw EEG data, as viable with other high precision neuroheadsets. A system has been developed for real-time control of a lower limb robotic-rehabilitation device through brain EEG signals. The brain state of attention/relaxation has been studied for driving the rehabilitation unit. An algorithmic approach has been developed for EEG signal processing along with classification to generate and send command signals for external device control.

2 System Architecture

The proposed work aimed to develop a BCI system for EEG-based control of an ankle rehabilitation device, for facilitation of ankle joint recovery from sprain or injury. The system employed Emotiv EPOC+ (Emotiv Systems Inc., USA) headset for data acquisition. The acquired data was read and processed using program developed in MATLAB platform (MATLAB 2015a). The control signal generated after

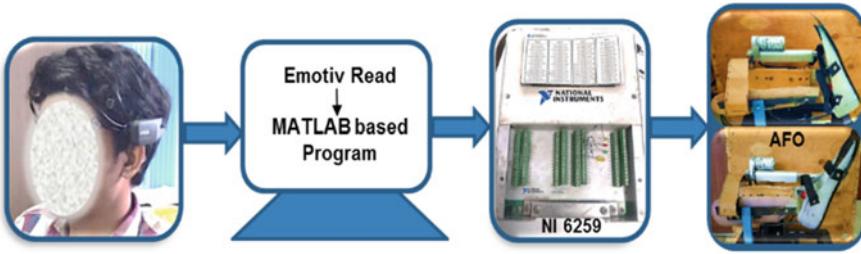


Fig. 1 Architecture of the proposed system

classification was transferred to the NI USB-6259 data acquisition (DAQ) unit for driving the external robotic-rehabilitation interface. The architecture of the proposed system is shown in Fig. 1. Further, each sub-system has been detailed.

2.1 Data Acquisition

Present paper acquired neural data using commercially available wireless EEG headset, Emotiv EPOC+. The headset has 14-bipotential sensors with gold plated electrodes. The electrodes are fixed at Af3, Af4, F3, F4, F7, F8, Fc5, Fc6, P7, P8, T7, T8, O1 and O2 positions, as per international 10-20 system of electrode placement. Two additional channels CMS (Common Mode Sense) and DRL (Driven Right Leg) served as reference channels for left and right hemisphere of the head, respectively. The data is recorded at a sampling frequency of 128 Hz and 16-bit analog to digital resolution.

Further the Emotiv system's software development kit (SDK) has three main featuristic suites:

1. “Expressiv Suite” analyzes the signal captured by 14-channels of Emotiv EPOC+ headset in terms of facial muscle twitches or eye movements. As a whole, 12 separate expressions, look left/right, blink, left/right wink, left/right smirk, clench, smile, laugh, furrow/raise brow can be identified.
2. “Affectiv Suite” represents EEG signal in terms of level of engagement or excitement or meditation or frustration of the headset wearer.
3. “Cognitiv Suite” interprets the changes in terms of thoughts or visual imageries to detect virtual movement of a cube. It includes commands like cube pull/push, lift/drop, rotate, disappear, etc.

2.2 Data Recording and Analysis

The Emotiv headset transmits the EEG data to host PC via Bluetooth, recorded directly with Emotiv Testbench software. However, the authors scripted a program here for direct acquisition and recording of the raw EEG data into its workspace. The data has been processed and classified within the script for sending output command signals to data acquisition hardware (NI USB-6259) supported by Data Acquisition Toolbox of MATLAB. The control command has been sent through wired USB cable.

2.3 Electronics and Control Hardware

National Instruments' USB-6259 DAQ hardware unit is used for receiving control commands from the host PC. Based on the algorithm, the control logic generates digital output signals. This signal controls an H-Bridge motor drive circuit that further controls the motion of the linear actuator of rehabilitation device. The driver circuit is powered using a 12 V/5 A SMPS. Ankle angle variation input from the electrogoniometer of the device is also measured using an analog input channel of DAQ-6259. The analog signal is processed using a third-order low pass Butterworth filter with 40 Hz cut off frequency.

2.4 Ankle-Foot Orthosis (AFO)

The main objective of the present work is to drive a robotic device, controlled via brain potential, for ankle therapy and rehabilitation. For this, the authors designed and developed a simple Ankle-Foot Orthosis (AFO) with linear actuation. It is an electromechanical actuated device with one degree of freedom, capable of providing powered actuation in the desired direction with a precision of one degree. The device enables the movement of foot in two actions: Dorsiflexion-moving the top of the foot toward the chin (only at the ankle) and Plantarflexion-moving the sole of the foot downward (pointing the toes). The device supports a movement of 25° in dorsiflexion and 40° toward plantarflexion action. Figure 2 shows the basic setup of the integrated device, with a participant's foot strapped inside.

3 Experimental Procedure

For the implementation of the proposed approach, experimental trials have been conducted over two healthy individuals with no known neurological disorder. The participants were naive to the BCI structure and methodology.



Fig. 2 Developed AFO for ankle rehabilitation

The trials were conducted over separate sessions with only one participant present at a time. Both were briefed about the cognitive process and the expected outcome of the procedure. The algorithm of the developed code is given in Table 1. Further, the procedure has been detailed in the following steps.

- Step 1. The Emotiv EPOC+ neuroheadset was placed at the participant's head by carefully considering the electrode positions. Each electrode was supported by a saline dipped felt pad in order to maintain conductivity for about an hour.
- Step 2. Each participant was given a training time of 10–15 min, so as to familiarize them with the cognitive process. Emotive SDK's "Cognitiv Suite" was used during this time, to enhance the attention and imagination level of the participants. Simultaneously data was recorded by the scripted program so as to generate prediction model from this training data. The data was rejected

Table 1 Algorithm of the scripted program for the proposed system

Algorithm
a. Acquire EEG data from Emotiv EPOC+ headset for training. (15 min)
b. Perform data normalization and smoothing after band pass filter between 0.1 and 40 Hz using fourth-order Butterworth filter
c. Extract features using ratio of theta to beta power spectral density value (θ)
d. Label the signal based on calculated threshold before classification
e. Train the SVM classifier and generate prediction model
f. Acquire EEG data for testing the generated model (0.5 s). Repeat b to d before calculating test label
g. Send control command to external DAQ unit (NI USB-6259)
h. Repeat f in case gyrosensor amplitude changes
i. Terminate program if prompted externally or "blink" command from headset

and a retraining would be needed in case of presence of excessive head or bodily movements during recording.

- Step 3. After initial training, subject was asked to imagine the most suitable cognitive command, i.e., the one easily controlled by the wearer's attention and level of imagination. The complementary control was achieved by a relaxed or neutral state. The wearer was specifically asked to avoid excessive movements at this stage. However, the developed algorithm has been equipped to counter this problem by prompting the user with a warning signal. For this, the authors utilized the data available from inbuilt gyroscope sensor of the Emotiv EPOC+. Also no data was recorded or tested unless the wearer is stationary and hence, no external hardware communication.
- Step 4. The data is processed by the scripted program wherein output predictions are made every half of a seconds. This corresponds to 64 new samples of the acquired data. Using overlapped windowing technique, the data processing has been performed over 128 (64 new +64 old) samples for sending decision signals to the output hardware. Data processing includes data normalization and smoothing along with feature extraction and classification. Feature extraction ensures the extraction of relevant features for successful classification. Proposed work calculated the ratio of theta to beta power spectral density value (θ), as detailed in [23], to set a threshold for classification between attention and relaxation state. Further, the supervised learning technique of Support Vector Machine (SVM) with linear kernel scale has been employed for feature classification.
- Step 5. The control command generated from the algorithm is sent as control input of the H-Bridge drive circuit using NI USB-6259 digital input/output pins.
- Step 6. The drive circuit then actuates the AFO device in response to the control command. The attention state was used for dorsiflexion of the robotic foot. Alternatively, relaxed state was used for plantarflexion. The range and speed of ankle flexion can be adjusted manually from the control algorithm, as per the patient/wearer's requirement. Efforts are underway to make this process autocontrolled as well.
- Step 7. An external control has also been provided for manual or "blink" controlled restriction of the robotic foot movement. This prevents the patient/user from false detections as well as during connection or power failure.

4 Results and Discussion

The proposed system resulted in successful control of a robotic ankle rehabilitation device (AFO) using brain potentials. Both the participant does not encounter much difficulty while achieving this feat. The total time taken by initial successful control of AFO by first participant took 25 min (after fifteen minutes of initial training) while second participant took about 21 min, hence an average of 23 min. Also after first

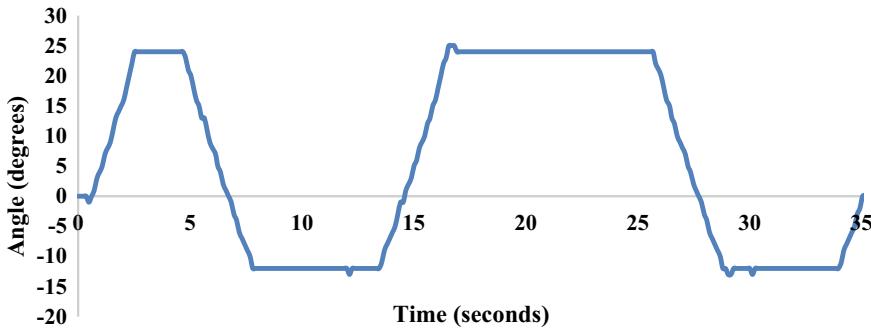


Fig. 3 Ankle angle trajectory for a trial of participant 1

thirty minutes, the participants were asked to relax for a period of ten minutes in order to remove any bias of the obtained results. After ten minutes, participants were able to repeat their performance without much efforts. Figure 3 represents the ankle trajectory plotted against the cognitive commands of relaxation and attention for first participant. The resultant plantarflexion and dorsiflexion are represented with angle rotation of $+24^\circ$ and -12° , respectively. Initial zero angle represents the idle or no command state. As subject was initially at rest, a plantarflexion was observed during start of all testing procedures. The AFO foot was held at the respective flexion position until the next or alternative command is received. As subject is perceived to have a relaxed state easily compared to attention, the duration for plantarflexion was generally observed more than dorsiflexion.

The whole process of translation of cognitive commands to robotic flexion took approximately less than a second. Even though this time is greater than the neurosensory response of human body, it is a viable option for training brain to induce motor-based learning. The repetitive cognitive efforts could prove to be a good mental exercise for a paralytic patient. Therefore, the proposed approach has the potential to successfully control a robotic-rehabilitation device using only cognitive signals.

Few limitations of the proposed approach are also acknowledged. The control has been exercised for classification of only two states. More classes should be included to implement a holistic system for robotic-rehabilitation. Also, efforts are needed to reduce the existing delay in control command execution. Emotiv headset also has few inherent limitations, therefore, care should be taken during data acquisition [9]. Overall, results are promising and important for the development of future rehabilitation applications.

5 Conclusion

This work presented the design and implementation details of a robotic-rehabilitation device controlled solely by brain originated cognitive signals. The aim was to induce ankle rehabilitation therapy based on the analysis of cognitive EEG. Using a combination of commercially available Emotiv EPOC+ and in-house developed AFO device along with developed algorithm, the authors were successful in real-time control of the rehabilitation device. This success encourages authors to explore new possibilities in the direction. The proposed system can be used in rehabilitation of patients with ankle sprains or injury. Other problems like foot drop and ankle spasticity can also be targeted based on the recommendation of the clinician.

Future work involves the extension of the approach to include more than two classification states. Other applications related to rehabilitation such as communication, robot control can also be explored. Also patient-based trials need to be conducted to ensure the efficacy of the proposed approach. Some other areas, such as entertainment, household also hold potential application of the system.

Acknowledgements The authors are grateful to Director, CSIR-CSIO, for providing all necessary resources required for the above work.

References

1. Abdulkader SN, Atia A, Mostafa M-SM (2015) Brain computer interfacing: applications and challenges. *Egypt Inform J* 16(2):213–230
2. Daly JJ, Wolpaw JR (2008) Brain–computer interfaces in neurological rehabilitation. *Lancet Neurol* 7(11):1032–1043
3. Ekandem JI et al (2012) Evaluating the ergonomics of BCI devices for research and experimentation. *Ergonomics* 55(5):592–598
4. Vourvopoulos A, Liarokapis F (2014) Evaluation of commercial brain–computer interfaces in real and virtual world environment: a pilot study. *Comput Electr Eng* 40(2):714–729
5. Poor GM et al (2011) Thought cubes: exploring the use of an inexpensive brain-computer interface on a mental rotation task. In: The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility. ACM
6. Wang Q, Sourina O, Nguyen MK (2010) Eeg-based “serious” games design for medical applications. In: International conference on Cyberworlds (CW). IEEE
7. Shenjie S, Thomas KP, Vinod A (2014) Two player EEG-based neurofeedback ball game for attention enhancement. In: International conference on systems, man and cybernetics (SMC). IEEE
8. Vi CT et al (2014) Quantifying EEG measured task engagement for use in gaming applications
9. Lievesley R, Wozencroft M, Ewins D (2011) The Emotiv EPOC neuroheadset: an inexpensive method of controlling assistive technologies using facial expressions and thoughts? *J Assist Technol* 5(2):67–82
10. Kawala-Janik A et al (2014) Use of a cost-effective neuroheadset emotiv EPOC for pattern recognition purposes. *Int J Comput* 13(1):25–33
11. Dharmasena S et al (2013) Online classification of imagined hand movement using a consumer grade EEG device. In: 8th IEEE international conference on industrial and information systems (ICIIS). IEEE

12. Hurtado-Rincon J et al (2014) Motor imagery classification using feature relevance analysis: an Emotiv-based BCI system. In: XIX symposium on image, signal processing and artificial vision (STSIVA). IEEE
13. Vourvopoulos A, Liarokapis F (2011) Brain-controlled NXT Robot: Tele-operating a robot through brain electrical activity. In: Third international conference on games and virtual worlds for serious applications (VS-GAMES). IEEE
14. Tripathy D, Raheja JL (2015) Design and implementation of brain computer interface based robot motion control. In: Proceedings of the 3rd international conference on frontiers of intelligent computing: theory and applications (FICTA). Springer
15. Lin J-S, Jiang Z-Y (2015) Implementing remote presence using quadcopter control by a non-invasive BCI device. *Comput Sci Inf Technol* 3(4):122–126
16. Thobbi A, Kadam R, Sheng W (2010) Achieving remote presence using a humanoid robot controlled by a non-invasive BCI device. *Int J Artif Intell Mach Learn* 10:41–45
17. Choi B, Jo S (2013) A low-cost EEG system-based hybrid brain-computer interface for humanoid robot navigation and recognition. *PLoS ONE* 8(9):e74583
18. Ranky G, Adamovich S (2010) Analysis of a commercial EEG device for the control of a robot arm. In: Proceedings of the IEEE 36th annual northeast bioengineering conference. IEEE
19. Fok S et al (2011) An EEG-based brain computer interface for rehabilitation and restoration of hand control following stroke using ipsilateral cortical physiology. In: 2011 annual international conference of the IEEE engineering in medicine and biology society. IEEE
20. Rechy-Ramirez EJ, Hu H, McDonald-Maier K (2012) Head movements based control of an intelligent wheelchair in an indoor environment. In: IEEE international conference on robotics and biomimetics (ROBIO). IEEE
21. Fattouh A, Horn O, Bourhis G (2013) Emotional BCI control of a smart wheelchair. *Int J Comput Sci Issues (IJCSI)* 10(3):32
22. Campbell A et al (2010) NeuroPhone: brain-mobile phone interface using a wireless EEG headset. In: Proceedings of the second ACM SIGCOMM workshop on networking, systems, and applications on mobile handhelds. ACM
23. Hooda N, Kumar N (2019) Cognitive imagery classification of EEG signals using CSP-based feature selection method. *IETE Tech Rev* 1–12

Chapter 8

Non-stationary Data Stream Analysis: State-of-the-Art Challenges and Solutions



Varsha S. Khandekar and Pravin Srinath

1 Introduction

In real world, to process the data for making it more useful, there are huge applications of machine learning which are using data typically in static form. But today's advancements in digital technology giving rise to emergence in the data and which is arriving continuously with high speed. This data is called as data stream. There is need of methods which are capable of processing this non-stationary, dynamic data. This includes network traffic analysis, credit card fraud detection, sensor data analysis, social network data analysis for sentimental analysis or product recommendations, medical diagnosis, and many more [1]. In data stream analysis stream data classification and prediction is one of the prolific research. Data stream classification distressed with challenges like limited usage of memory and restricted amount of time for classifying the new instance as data is arriving high speed and volume. In addition to this, there are other challenges in data stream classification which are raised due to data complexities, and first challenge in this is due to dynamic or non-stationary nature of data stream where underlying distribution of data is changing over the time awfully called as concept drift. While another important challenge faced by data stream classification is class imbalance where one of the classes is highly underrepresented than others classes which creates class imbalance situation [2]. All aforementioned challenges are overall affecting the performance of classifiers most of the time if these are not handled they are deteriorating it. There are many algorithms and solutions suggested in the literature to handle these issues separately but when both are occurring simultaneously becomes more inspiring and less

V. S. Khandekar (✉) · P. Srinath

Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering, Mumbai, India

e-mail: varsha.khandekar@gmail.com

P. Srinath

e-mail: Pravin.Srinath@nmims.edu

© Springer Nature Singapore Pte Ltd. 2020

S. Bhalla et al. (eds.), *Proceeding of International Conference on Computational Science and Applications, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-15-0790-8_8*

attentive issue of research. Many existing surveys on data stream analysis [3, 4] or classification are with focus on concept drift [5], data imbalance when they occur separately or simultaneously, and some are specifically on ensemble approach for stream classification [6, 7]. There is still lack of systematic survey where incremental, online, and ensemble approaches are considered for handling different challenges in data stream classification under one roof.

This survey has organized in different sections where Sect. 2 presents concept drift in data stream classification, Sect. 3 is on state-of-the-art techniques to handle them, Sect. 4 discusses imbalance issue in the data classification and incremental, online, and ensemble learning approach used to handle these issues independently, Sect. 5 discusses the techniques used to handle concept evolution and new class detection, and the last section concludes with future research opportunities in this field.

2 Concept Drift in Data Stream Classification

This section discusses basic concepts, notations used for defining concept drift. It gives an overview of on different approaches used to tackle these issues.

Data stream is continuously, sequentially arriving over time at very high speed, unbounded set of items. Here, supervised learning is assumed where every item is defined in the form of items. (x_t, y_t) that is every item is labeled data item. Where x_t is set of attribute or feature values and y_t is class label or target value at time t . This target variable y_t is either discrete or nominal. Supervised learning usually gains the knowledge from past data by finding the relationship between the attribute values and class label. This knowledge is utilized by classifier for predicting the class label for new unseen instance which is unlabeled.

Stationary data streams: The relationship between x_t and y_t is unknown but fixed which is not changing over the time.

Non-stationary data stream: The relationship between input attribute values and target variable typically called as a concept is changing over the time. If the joint probability at time t is $p_t(x, y)$ then at time $t + k$ this joint probability may not be same, i.e., $p_t(x, y) \neq P_{t+k}(x, y)$ at $t + k$. This is an indication of concept drift.

Due to concept drift, there is change in the three variables of Bayes' theorem

$$p(y|x) = \frac{p(y)p(x|y)}{p(x)} \quad (1)$$

Based on Eq. (1), concept drifts are defined as:

1. Change in posterior probability $p(y|x)$ called as class drift.
2. Change in prior probabilities $p(y)$.
3. Change in class conditional probability $p(x|y)$.

Change in posterior probability distribution is called as real concept drift while rest of two drifts are virtual drifts [5, 8].

Fig. 1 Sudden or abrupt drift

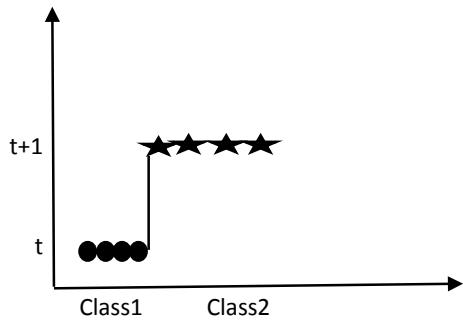
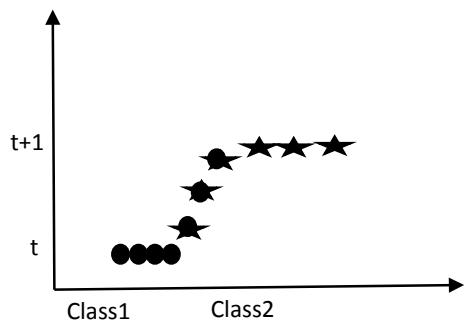


Fig. 2 Gradual drift

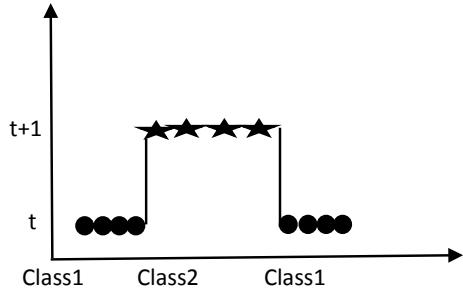


Speed of Concept Drifts: The change in data distribution by considering duration of time steps decides the speed of occurrence of concept drift. Following are the different types of concept drifts based on speed.

1. Sudden: At certain time instance, there is sudden or abrupt change in the class assignment as shown in Fig. 1.
2. Gradual: This drift occurs when there is change in class distribution after certain time steps that is previously occurred concept may reappear for certain time steps again depicted in Fig. 2.
3. Recurring: Instead of new concept, certain concept reappears again at irregular or regular intervals. Figure 3 illustrates this recurring drift.

3 Different Approaches to Handle Concept Drift

There are two main approaches have been suggested in the literature to handle the concept drift in non-stationary environment. The first approach is active approach where explicit concept drift detection technique is used and based on change the classifier is adapted. These change detection mechanisms involve various statistical tests.

Fig. 3 Recurring drift

3.1 Change Detection Mechanisms Using Statistical Analysis

Authors have developed various mechanisms to detect the underlying change in data distribution by extracting statistical features like sample mean, variance, etc. from incoming data streams

- a. Statistical parametric and non-parametric tests:

In [9], non-parametric univariate statistical tests for discerning virtual drift in the multidimensional data stream are used. Authors have applied Kolmogorov–Smirnov, t-test, and Wilcoxon rank-sum tests on all the features of dataset. Based on a change in the variation of statistical value inspected for predetermined window size, change is detected. Conjunctive normal form (CNF) density function along with Mann–Whitney test is explored in [10] to measure the significance of difference in the sequences which is based on binary representation of data. A computational intelligence extension of the adaptive CUSUM test to inspect variations in sample statistical moment features as well as internal core variables coming from other statistical tests is presented in [11]. The intersection of confidence intervals (ICI) CDT and its variants have been demonstrated for change detection in [12].

3.2 Threshold-Based Mechanisms

Threshold-based mechanisms are simple and straightforward which are based on pre-defined thresholds set by considering classification error or accuracy. Use Bernoulli exponentially weighted moving average values of error observations is investigated in [13] for concept drift detection. For detection of gradual drifts [14] notices the changes in exponentially weighted moving average of misclassification error rate. Statistical hypothesis testing with Bernstein bound as a threshold has been used for concept change detection by assessing variations in the classification errors in [15]. A reservoir sampling mechanism presented in [16] used Bernstein bound as a threshold value for change detection. In a similar way, [17] proposed two approaches for change detection, using moving average technique for abrupt change while weighted

moving average for gradual change detection is done where McDiarmid's inequality bounds are used as a threshold. Feature-based drift detection method has been introduced in [18] which is based on Hellinger distance to measure data change in data distribution combined with t-statistics threshold.

The second step of active approach is adaption of classifier if change detection module triggers the change. Following are the major mechanisms used under active approach.

3.3 Just-in-Time Classifiers

These are using windowing techniques (where window size for considering subset of instances from incoming data stream is predefined) and competent to work in non-stationary environment using adaptive approach. For exploiting change detection in both supervised and unsupervised data [19] used probability density function and distribution in input data. To deal with recurrent concepts, novel just-in-time (JIT) adaptive classifiers have been suggested by [20] which is based on two change detection tests. In [21], ICI-based change detection test that combines the ICI rule with a local polynomial approximation (LPA) estimator has explored with improved classification accuracy with abrupt change detection.

3.4 Weight-Based Classifiers

Appropriate weights are assigned using different decay functions like polynomial or exponential to the instances in a dataset and are decreased with weight. This approach mostly coupled with collection of classifier.

Second approach known as passive approach where instead of using explicit trigger module detecting the concept drift continuous adaption of classifier is done for incoming data stream. Following are the different techniques used under passive approach.

3.5 Decision Tree-Based Classifier

Decision trees are one of the prevalent and widely used classifiers for handling the concept drift in data streams because of its overlooking features like its ability to work in presence of outliers or irrelevant attributes, robustness. Decision trees are based on divide and conquer strategy where complex problems can be solved by dividing it into simpler subproblems. Many of them have used Hoeffding inequality bound while splitting the node. In [3] have implemented Very Fast Decision Tree (VFDT) which supports incremental and anytime approach along with one time scan. VFDT

are the rising decision trees to handle the stream data. Extended version of VFDT has been presented in [22] where Naive Bayes classifier is implemented at leaf level, and also it was able to deal with numerical feature values named as VFDTc or CVFDT. Similar to this, an adaptive Hoeffding window tree (HWT) with sliding window approach where, instead of using fixed size sliding window to detect changes, HWT employs an adaptive window at each internal node has been proposed in [23] which is an improvement over CVFDT. Another attempt for improving CVFDT has been explored for solving the underlying problem by incorporating ambiguities in CVFDT. At the time of node splitting, various options are explored by considering multiple attributes which can reflect multiple concepts at every node [24]. Although Hoeffding bound tree algorithm is common in literature, [25] presented McDiarmid's bound for analyzing data streams and endeavored to show the limitations of Hoeffding bound for computing evaluation parameters like information gain or Gini index.

3.6 Multi-classifier or Ensemble Approach

Multi-classifier or ensemble approach is encouraging and popular research direction in data stream analysis especially for concept drift handling because of its distinctive features. Following are important advantages of ensemble approach identified from the literature [26–28]:

To get the enhanced result, ensemble approach combines multiple weak classifiers using appropriate majority voting rule or using some other techniques.

Ensemble classifiers are able to handle the scarcity of data as well as huge amount of data. It is flexible to handle new data instances coming continuously simply by incorporating new classifier in the existing one.

It is possible to use the historical knowledge use instead of building new model from scratch for every new instance [29].

Ensemble approach is either batch-based or online learning-based. In batch-based ensemble approach at particular time instant, new classifier is added in the family of ensemble classifiers for a batch of data which essentially provides innate way to deal with concept drift. In this approach, forgetting mechanisms are also used to remove the poor performance members or by exploiting historical learnt knowledge recurring concepts are handled. These ensemble approaches are adopted either using passive or active method. Ensemble approach is further categorized depending upon the methodology to build the ensemble model to achieve the final performance result as follows [6].

3.6.1 Majority and Weighted Majority Voting

Majority voting is straightforward and simple ensemble classifier where prediction of a class label for a new instance is done using majority rule that is the same class is assigned new instance which is predicted by most of the members of class by

assuming same weight to all classifiers while in weighted majority voting weights are assigned depending upon certain condition like accuracy or overall error rate of prediction. Authors of [30] presented leveraging bagging to impose randomization on input with assigning weights on input instances in combination with majority voting and output correcting code. With the inspiration of accuracy weighted ensemble, adaptive accuracy ensemble using weighting mechanism and updating the classifiers according to current distribution has been proposed by [31]. For reacting to different types of drifts [32], designed new hybrid accuracy updates ensemble model by combining different weighting mechanisms with incremental learning. In this approach, authors' requirement of cross-validation of candidate classifier and maintaining classifier buffer is removed and also pruning of base learners is done along with updating of component classifier which has improved accuracy and memory usage. Probabilistic framework for comparing majority voting, weighted majority voting, recall combiner, and Naïve Bayes classifier has been proposed in [33] which gives minimum classification error. In addition to this, there are Learn++ family ensemble algorithms which are using batch incremental learning. Rather than using fixed size window or adaptive window, another attempt is done by [34] where prediction is done by combining multiple window sizes that is hybrid technique of windowing method and adaptive ensemble.

For learning new concept Learn++NC [35] investigated a dynamically weighted consult-and-vote mechanism to determine which classifiers should or should not vote for any given instance based on the (dis)agreement among classifiers trained on different classifiers. In [36], Learn++NF uses random subspace selection using vertical partitioning for training the components in ensemble and to deal with missing features. Both Learn++NC and Learn++NF algorithms are based on stationary environment. To handle non-stationary data streams, Learn++NSE [37] uses novel approach for voting the weights using current and past learnt parameters and each member's time-dependent accuracy in the ensemble for sudden, gradual, or recurrent concepts.

Diversity-based ensemble classifiers: Recently, diversity is considered as one of the important factors in ensemble models, but there is no generalized theoretical guarantee which shows the correlation between accuracy and diversity in classifier components. The role of diversity in accuracy of ensemble classifier and size of ensemble in presence of concept drift has been focused in [38], where analysis shows that diversity needs to be addressed to reduce initial error due to drift. To acquire more generalization on new concept, different diversity levels should be considered before and after the drift. Based on this reveal, new approach diversity for dealing with drifts (DDD) is presented by [8]. To exploit historical knowledge of component classifiers, ensemble diversity, and transfer learning approach has been proposed to resolve the issue of preserving the historical models for drift handling in [29].

3.6.2 Online Learning-Based Ensemble Classifiers

In online learning, only one instance is considered at a time using single pass which makes it faster and limited usage of memory in comparison with batch-based ensembles. Incremental streaming decision tree and random forests are combined which is also able to handle unlabeled data introduced in [39] which is fast and achieved higher classification accuracy. In this approach, trees are added in forest by considering threshold value of classification error. Authors of [40] introduced strategy to transform block-based learning to online ensemble learning to process single example at a time using incremental learner and weighting mechanisms with efficient time and memory.

3.7 Alternative Approaches

There are some alternative approaches for dealing with concept drifts like active learning, transfer learning, and semi-supervised learning. Most of the concept drift handling work have not taken account the concept latency which is practically not true all the time. To knob label latency active in non-stationary environment, some researchers joined active learning with ensemble approach. In [41], active learning is presented for drifting streams to control the adaption of ensemble. Optimal weight assignment to the components in ensemble and update of weight after the availability of label is done through active learning using minimum variance principle. New strategies for active learning to explicitly handle concept drift have been proposed by [42], which are based on uncertainty, dynamic allocation of labels by using balanced labeling over a time, and randomization of search space by querying over all instance space.

Following Table 1 summarized key findings, advantages, and disadvantages of these approaches.

4 Class Imbalance

Class imbalance arises when the number of instances of one of the classes (majority class) is more than number of instances of another class (minority class) that is generally called skewed distribution of data instances. Dis-proportionality in distribution of classes leads to highest classification error as most of machine learning classification algorithms are biased toward majority classes and at the same time misclassifying minority classes are more expensive. For static dataset, this issue has been widely studied using machine learning techniques and its negative effect on classifier's performance is shown in neural networks [43], SVM [44], and k-nearest neighbor [45, 46]. Handling the imbalance problem with concept drift becomes more

Table 1 Approaches to handle concept drift

Approach	Key findings	Advantages	Disadvantages
Statistical test	Based on continuous sequential analysis	Reduced complexity	Cannot control false-positive rate and cannot handle gradual drift
Threshold-based mechanisms	Moving average and weighted Moving average are suitable for sudden and gradual concept drifts, respectively	Are classifier independent	Selecting threshold value is crucial which may result in false or wrong concept detection
Just-in-time classifiers	Able to handle gradual drifts and can observe input feature distribution as well as classification error	JIT classifiers based on windowing techniques are able to adapt the window size dynamically	JIT based on weighting mechanism requires whole data in memory
Decision tree classifier	Very fast decision trees based on Hoeffding bound and MCDIARMID'S inequality bounds are used for classification	Adaptive to time changing circumstances	Prone to overfitting
Ensemble classifiers	Along with bagging, boosting, majority weighting, and weighted majority weighting have shown improved accuracy in stream classification	New concept can be easily incorporated by adapting new knowledge	Storage overhead due to multiple models

challenging. Following are few state-of-the-art solutions to handle the concept drift and imbalance problem.

Sampling methods: Sampling methods typically modify the underlying data to make its distribution more balanced one to improve the classifier's accuracy. Over-sampling is one of the sampling methods where instances in minority classes are added whereas in undersampling instances from majority classes are removed to achieve certain degree of balance. Rather than considering all minority examples from whole data set at a time in SERA [47] and REA, few numbers of most similar instances to minority class are considered from current block based on Mahalanobis which are combined with majority class instances from current block and maintaining single hypothesis. Similar approach is extended with ensemble approach in [48]. Authors of [49] extended Learn++NSE for non-stationary and imbalanced environments using a bag of multiple classifiers and applying undersampling on majority class.

Nguyen et al. [50] used random undersampling to deal imbalanced data streams and proposed an online algorithm by selecting lower probability majority class examples for training. But it is based on assumption that imbalance rate does not change over time and information about the minority and majority classes is known prior. Gao et al. [51] utilized ensemble model for handling concept drift and oversampling for dealing with class imbalance. Same approach used in [52] with some improvements in both parts. By extending SMOTE, oversampling technique and some alternative smart oversampling and undersampling techniques were developed like ADASYN [53], MWMOTE [54], and one-sided selection [45], GLMBoost with SMOTE [55].

Cost-sensitive learning methods: This approach is based on assigning different costs for learning the classifiers. Adel Ghazikhani et al. [56] proposed online two layer ensemble of neural network where in first layer cost-sensitive learning is used which uses cost-sensitive feature and in second layer adapted winnow weighting strategy is used to deal with imbalance class situation and concept drift. Wang et al. [57] proposed cost-sensitive online classification algorithm where concept drift has not been considered explicitly.

Additional methods: In addition to above methods, there are other methods which are using some different approaches. In [58], Hellinger distance and information gain applied with weighted ensemble where instead of considering classification performance distributional, information-theoretic properties of data are taken into account which has improved classification accuracy [59] (Table 2).

Table 2 Approaches to handle class imbalance

Approach	Key findings	Advantages	Disadvantages
Sampling methods	Synthetic minority or majority oversampling or undersampling data-level approaches can be combined with non-stationary environment to handle imbalanced dataset	Improve the classification accuracy by making data balanced without changing underlying feature and concept space	For large dataset possibility of loss of information or possibility of overfitting of the model
Cost-sensitive methods	Cost-sensitive methods can be applied to imbalanced data streams. Learning from these algorithms can be connected with learning from imbalanced datasets	Cost-sensitive algorithms are superior alternatives to sampling methods	Degree of imbalance has an affect on complexity of cost-sensitive algorithms

Table 3 Approaches to handle concept evolution or novel class detection

Approach	Key findings	Advantages	Disadvantages
Parametric	Parametric approaches are statistically based on parameters of data like mean and covariance	Simple to design and straightforward	Based on prior assumption
Non-parametric	Non-parametric novel class detection can be implemented as a single class classifier or as ensemble	Unrestricted to underlying data distributions. More flexible and general	Complex to implement

5 Concept Evolution or Novel Class Detection

Concept evolution is a special case of concept drift where classes appear or fade gradually. As in concept drift, there is change in probability distribution of $p(x, y)$ where x is feature vector and y is corresponding class label. In concept evolution or novel class detection, there is change prior probability $p(y)$ [60]. Novel class detection methods are either parametric or non-parametric. In [61], parametric approach is used based on estimating parameters of distribution of data and novel class is identified as one which is not following that distribution. Authors of [59, 62, 63] used kernel based and learning rule based non-parametric approaches without assuming prior probability distribution (Table 3).

6 Conclusion

Recently to process dynamic or non-stationary data streams, many models have been adopted out of which many are intended to focus on concept drift, while there are some other challenges like new concept evolution and imbalance problem which are rarely explored. Furthermore, it is difficult to design an approach which deals with all types of concept drifts which are based on the speed of occurrence. So it is required to find certain measure for concept drift by which we can deal with all types of concept drifts like sudden, gradual, or recurring drifts. In addition to this, it is required to find an approach which not only deals with concept drift but also is able to handle class imbalance, concept evolution, or new concept detection. By the study of existing approaches for data stream classification, we found above various challenges and still some of the following research questions need to be addressed. (1) Can we make data stream classification or analysis parallel or scalable for proper utilization of computing power when it is ample? (2) Can we make data stream analysis online to give quick response when computing resources are limited? (3) Can system able to handle more realistic, big, and continuous data?

References

1. Zliobaite I, Pechenizkiy M, Gama J (2016) An overview of concept drift applications. In: Big data analysis: new algorithms for a new society. Springer, pp 91–111
2. Brzezinski D, Stefanowski J (2018) Ensemble classifiers for imbalanced and evolving data streams. *Int J Data Min Time Ser Streaming Databases* 44–68
3. Domingos P, Hulten G (2000) Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, pp 71–80
4. Ditzler G, Roveri M, Alippi C, Polikar R (2015) Learning in nonstationary environments: survey. *IEEE Comput Int Mag* 10(4):12–25
5. Ao JO, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. *ACM Comput Surv (CSUR)* 46(4):44
6. Gomes HM, Barddal JP, Enembreck F, Bifet A (2017) A survey on ensemble learning for data stream classification. *ACM Comput Surv (CSUR)* 50(2):23
7. Krawczyk B, Minku LL, Stefanowski J (2017) Ensemble learning for data stream analysis : a survey. *Inf Fusion* 37:132–156
8. Minku LL, Yao X (2012) DDD: a new ensemble approach for dealing with concept drift. *IEEE Trans Knowl Data Eng* 24(4):619–633
9. Sobolewski P, Wozniak M (2013) Comparable study of statistical tests for virtual concept drift detection. In: Proceedings of the 8th international conference on CORES. Springer, pp 329–337
10. Dries A, Ulrich R (2009) Adaptive concept drift detection. *J Stat Anal Data Min* 2(5–6):235–246
11. Alippi C, Roveri M (2008) Just-in-time adaptive classifiers—part I : detecting nonstationary changes. *IEEE Trans Neural Netw* 19(7):1145–1153
12. Alippi C, Boracchi G, Roveri M (2010) Change detection tests using the ICI rule. In: Proceedings of the international joint conference on neural networks, pp 1–7
13. Nishida K, Yamauchi K (2009) Learning detecting understanding and predicting concept changes. In: Proceedings of the international joint conference on neural networks, pp 2280–2287
14. Ross G, Adams N, Tasoulis D, Hand D (2012) Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognit Letters* 33(2):191–198
15. Sakthithasan S, Pears R, Koh YS (2013) One pass concept change detection for data streams. In: Lecture notes in computer science: vol 7819. Advances in knowledge discovery and data mining. Springer, Berlin, pp 461–472
16. Pears R, Sakthithasan S, Koh YS (2014) Detecting concept change in dynamic data streams. *Mach Learn* 97(3):259–293
17. Frías-Blanco I, del Campo-Ávila J (2014) Online and non-parametric drift detection methods based on Hoeffding's bounds. *IEEE Trans Knowl Data Eng* 27(3):810–823
18. Ditzler G, Polikar R (2011) Hellinger distance based drift detection for nonstationary environments. In: Proceedings IEEE symposium on computational intelligence dynamic uncertain environments, pp 41–48
19. Alippi C, Boracchi G, Roveri M (2012) Just-in-time ensemble of classifiers. In: Proceedings of the international joint conference on neural networks, pp 1–8
20. Alippi C, Boracchi G, Roveri M (2013) Just-in-time classifiers for recurrent concepts. *IEEE Trans Neural Netw Learn Syst* 24(4):620–634
21. Alippi C, Boracchi G, Roveri M (2011) A just-in-time adaptive classification system based on the intersection of confidence intervals rule. *Neural Netw* 24(8):791–800
22. Medas P. Accurate decision trees for mining high-speed data streams. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining, pp 523–528
23. Bifet A, Gavald R (2009) Adaptive parameter-free learning from evolving data streams (August)
24. Liu J, Li X, Zhong W (2009) Ambiguous decision trees for mining concept-drifting data streams. *Pattern Recognit Lett* 30(15):1347–1355
25. Rutkowski L, Pietruczuk L, Duda P, Jaworski M (2013) Decision trees for mining data streams based on the McDiarmid's bound. *IEEE Trans Knowl Data Eng* 25(6):1272–1279

26. Hoens TR, Chawla NV (2012) Learning in non-stationary environments with class imbalance. In: Proceedings of the 18th ACM SIGKDD conference on KDD, pp 168–176
27. Wozniak M, Corchado E (2014) A survey of multiple classifier systems as hybrid systems. *Inf Fusion* 16:3–17
28. Tsymbal A, Pechenizkiy M, Cunningham P (2008) Dynamic integration of classifiers for handling concept drift. *Inf Fusion* 9(1):56–68
29. Sun Y, Tang K, Zhu Z, Yao X (2018) Concept drift adaptation by exploiting historical knowledge. *IEEE Trans Neural Netw Learn Syst* 1–11 (2018)
30. Bifet A, Holmes G, Pfahringer B (2010) Leveraging bagging for evolving data streams. In: Proceedings of European conference on machine learning/PKDD I, pp 135–150
31. Brzezi D (2011) Accuracy updated ensemble for data streams with concept drift. In: Proceedings of 6th HAIS international conference hybrid artificial intelligence system II, pp 155–163
32. Brzezinski D, Stefanowski J (2014) Reacting to different types of concept drift. *IEEE Trans Neural Netw Learn Syst* 25(1):81–94
33. Kuncheva LI, Rodríguez JJ (2014) A weighted voting framework for classifiers ensembles. *Knowl Inf Syst* 38(2):259–275
34. Yoshida S, Hatano K, Takimoto E (2011) Adaptive online prediction using weighted windows. *IEICE Trans* 94(10):1917–1923
35. Muhlbaier MD, Topalis A, Polikar R (2014) Learn++. NC: combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes (May)
36. Polikar R, Depasquale J, Syed H, Brown G, Kuncheva LI (2010) A random subspace approach for the missing feature problem. *Pattern Recognit* 43(11):3817–3832
37. Elwell R, Polikar R, Member S (2011) Incremental learning of concept drift in nonstationary environments. *IEEE Trans Neural Netw* 22(10):1517–1531
38. Minku LL, Member S, White AP (2010) The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Trans Knowl Data Eng* 22(5):730–742
39. Abdulsalam H, Skillicorn DB, Martin P, Society IC (2011) Classification using streaming random forests. *IEEE Trans Knowl Data Eng* 23(1):22–36
40. Brzezinski D, Stefanowski J (2014) Combining block-based and online methods in learning ensembles from concept drifting data streams. *Inf Sci (NY)* 265:50–67
41. Zhu X, Zhang P, Lin X, Shi Y (2010) Active learning from stream data using optimal weight classifier ensemble. *IEEE Trans Syst Man Cybern Part B* 40(6):1607–1621
42. Bifet A, Pfahringer B, Holmes G (2014) Active learning with drifting streaming data. *IEEE Trans Neural Netw Learn Syst* 25(1):27–39
43. Visa S (2005) Issues in mining imbalanced data sets—a review paper. In: Proceedings of the 16th, MAICS, pp 67–73
44. Yan R, Liu Y, Jin R, Hauptmann A (2003) On predicting rare class with SVM ensemble in scene classification. In: Proceedings of IEEE international conference on acoustics speech signal processing, vol 3, pp 21–24
45. Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor Newslett* 6(1):20–29
46. Zhang J, Mani I (2003) kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of the ICML
47. Chen S, He H (2009) SERA selectively recursive approach towards nonstationary imbalanced stream data mining. In: Proceedings of international joint conference on neural networks, pp 552–529
48. Chen S, He H, Li K, Desai S (2010) MuSeRA: multiple selectively recursive approach towards imbalanced stream data mining. In: Proceedings of international joint conference on neural networks, pp 2857–2864
49. Ditzler G, Polikar R (2010) An ensemble based incremental learning framework for concept drift and class imbalance. In: Proceedings of international joint conference on neural networks, pp 1–8

50. Nguyen HM, Cooper EW, Kamei K (2011) Online learning from imbalanced data streams. In: Proceedings of international conference on soft computing pattern recognition, pp 347–352
51. Fan W, Yu PS (2007) A general framework for mining concept-drifting data streams with skewed distributions. In: Proceedings of SIAM international conference on data mining, vol 7, pp 3–14
52. Chen S, He H (2011) Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach. *Evolving Syst* 2(1):35–50
53. He H, Bai Y et al (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of international joint conference on neural networks, pp 1322–1328
54. Barua S, Islam M, Yao X, Murase K (2014) MWMOTE majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans Knowl Data Eng* 26(2):405–425
55. Hao M, Wang Y, Bryant SH (2014) An efficient algorithm coupled with synthetic minority over-sampling technique to classify imbalanced PubChem BioAssay data. *Anal Chim Acta* 806:117–127
56. Ghazikhani A, Monsefi R, Yazdi HS (2013) Ensemble of online neural networks for non-stationary and imbalanced data streams. *Neurocomputing* 122:535–544
57. Wang J, Zhao P, Hoi SCH (2014) Cost-sensitive online classification. *IEEE Trans Knowl Data Eng* 26(10):2425–2438
58. Lichtenwalter RN, Chawla NV (2009) Adaptive methods for classification in arbitrarily imbalanced and drifting data streams. In: New frontiers in applied data mining, PAKDD, international workshops, Bangkok, pp 53–75
59. Masud MM, Gao J, Member S (2011) Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Trans Knowl Data Eng* 23(6):859–874
60. Sun Y, Tang K (2016) Online ensemble learning of data streams with gradually evolved classes. *IEEE Trans Knowl Data Eng* 28(6):1532–1545
61. Roberts SJ (2000) Extreme value statistics for novelty detection in biomedical signal. In: Proceedings of the IEEE conference on science, measurement and technology, vol 147. IET pp 363–367
62. Ahmed T, Coates M (2007) Multivariate online anomaly detection using kernel recursive least squares. In: Proceedings of IEEE INFOCOM, pp 625–633
63. Yeung D, Bay CW, Kong H (2002) Parzen-window network intrusion detectors. In: Proceedings of the 16th international conference on pattern recognition, vol 4. IEEE

Chapter 9

Parallel Job Execution to Minimize Overall Execution Time and Individual Schedule Time Using Modified Credit-Based Firefly Algorithm



Hardeep Kaur and Anil Kumar

1 Introduction

Parallel computing used extensively nowadays to provide resources to the clients on pay per use basis. Parallel services generally include service-level agreement (SLA). A resource that parallel service provider (CSP) provides to the user is according to the SLA. Theoretically, parallel contains infinite resources but user requirements are enhancing day by day causing deficiency of resources within the parallel [12]. To tackle the issue of deficiency of resources, resource sharing was suggested. Resource sharing enhances performance of parallel in the short run as resource utilization improves. Resource sharing however degrades the performance in the long run as the load on individual resource enhanced [9].

One of the methodologies to handle the over-burdening is job scheduling. Job scheduling utilizing multi-heuristic systems like hereditary methodology is normal. The issue with this methodology is poor combination rate [7]. Alteration to GA prompts molecule swarm enhancement. It mimics the aggregate conduct of social animals [10]. This calculation searches for ideal arrangement by spreading the molecule along every one of the headings. Swarm that finds the nourishment at the lead position is trailed by the various swarms. Combination rate is poor on the off chance that recurrence of jobs turns out to be amazingly high causing overall execution time and individual schedule time to increment past indicated extents [8].

Job waiting time and fitness function evaluation are also investigated using the proposed system. The waiting time indicates the amount of time required by the

H. Kaur (✉) · A. Kumar

Department of Computer Science and Technology, Guru Nanak Dev University, Amritsar, India
e-mail: badhanhardeep7@gmail.com

A. Kumar

e-mail: anil.dcese@gmail.com

job before resources are allotted to the job. The waiting time for the job is reduced considerably by the use of proposed system. The fitness function is obtained using the overhead encountered; hence, this function value must be minimized. The proposed system considerably reduces this objective function.

To defeat the issues brought about by existing writing, proposed writing utilizes another meta-heuristic firefly calculation with job choice stage altered with min cost and max benefit target work. Rest of the paper is composed as under: Segment 2 gives the writing review of existing job scheduling components utilized for accomplishing ideal outcomes, segment 3 gives the proposed framework with the technique to be pursued, area 4 gives outcome and execution examination, segment 5 gives end and future extension and last segment gives the references.

2 Literature Survey

This section puts the light on various multi-heuristic approaches used to optimize process of job execution in multi-cluster environment.

Javanmardi et al. [5] proposed job requesting instrument by the utilization of hereditary calculation. Hereditary calculation utilizes populace that is number of jobs displayed to the framework. Arbitrary choice method was pursued to produce posterity. Execution of jobs records the outcome as far as overall execution time and individual schedule time. Chromosomes haphazardly rearranged to be introduced to the main stage once more [3]. Procedure repeats until wanted target was met or ages lapses. This technique in spite of the fact that outcomes in most ideal job requesting yet poor combination rate chase the general strategy.

Gopalan and Suresh [4] proposed subterranean insect state streamlining for burden adjusting in parallel framework. Insect province streamlining instrument pursued accomplishes load offsetting with the cost of combination rate.

Keskinturk et al. [6] proposed a multi-objective ACO for job scheduling. The essential parameter utilized for enhancement incorporates cost. This writing utilized two limitations: client spending plan and overall execution time. Client spending plan utilized in this writing progresses toward becoming edge past which cost experienced makes plan fall flat. Overall execution time related to the calendar was recorded and checked for streamlining.

Wang and Zeng [11] proposed firefly improvement system where vitality proficiency ends up basic in the relocation procedure. This methodology relocates intensely stacked VM to least-stacked VM without influencing vitality proficiency related to server farm.

Cui et al. [1] proposed proficient firefly calculation to take care of issue of job shop scheduling. Congruity seek calculation was utilized as a team with firefly calculation in this methodology. Congruity look gives optimality in resource seeking and firefly allots the jobs to the resources for execution. Intermingling rate anyway ends up poor as mind-boggling and bigger number of jobs progresses toward becoming member of the framework. Proposed job requesting component by the utilization

of hereditary calculation. Hereditary calculation utilizes populace that is number of jobs displayed to the framework. Arbitrary choice method was pursued to produce posterity. Execution of jobs records the outcome as far as overall execution time and individual schedule time. Chromosomes haphazardly rearranged to be exhibited to the principal stage once more. Procedure emphasizes until wanted target was met or ages terminate. This technique despite the fact that outcomes in most ideal job requesting yet poor intermingling rate chase the general method.

Fidelis and Arroyo [2] proposed subterranean insect settlement advancement for burden adjusting in parallel framework. Insect province improvement instrument pursued accomplishes load offsetting with the cost of assembly rate. Overall execution time related to the timetable was recorded and checked for streamlining.

To overcome the problem, proposed literature combines firefly algorithm with the highest credited job first algorithm. Description of the proposed system is given in the next section.

3 Proposed System

This section gives the in-depth study of the proposed system followed to reduce overall execution time and individual schedule time associated with the system. The phases associated with highest credited job first firefly approach (CJFFA) are described as under.

3.1 Credit-Based Approach

In this approach, highest credited job is being identified. Highest credited job is identified by determining cost associated with the job. Higher the cost, minimum will be the profit. Job cost is assumed to be directly proportional to burst time of the job. Sorting of job on the basis of cost associated with the job is made and then presented to the next phase for execution. The algorithm associated with highest credited job first mechanism is given as under

- Input job list and obtain cost associated with each job.
- For Loop=1:N
- For Loop1=1:N-i
- If(Job_cost_{Loop1}>Job_cost_{Loop1+1})
- Perform swapping of jobs
- T=Job_cost_{Loop1}
- Job_cost_{Loop1}=Job_cost_{Loop1+1}
- Job_cost_{Loop1+1}=Temp
- End of if
- End of for

- End of for
- Store Jobs in Job_List_{Loop1}

This job list is transferred to the next phase for execution by firefly algorithm. Firefly algorithm executes the jobs by allotting the jobs to the resource cluster.

3.2 *Firefly Algorithm for Job Execution*

The job list so obtained from previous phase is presented to the firefly algorithm to determine the optimal solution in terms of overall execution time, individual schedule time, fitness function and waiting time. Job list is arranged in terms of maximum profit or low-cost basis. The algorithm used to achieve the result is given as under.

Firefly algorithm

- Receive job list from highest credited job first approach described in section 3.1
 - Generate fireflies corresponding to jobs in the job list
 - Input number of generations(G)
 - Repeat while generations exhausted
 - Obtain local solution(Overall execution time_i and Individual schedule time_i)
 - Assign fireflies to the resource clusters
If(available(Resource_i))
Resource_i=Resource_i-Fireflies _Req_i
Highlight the firefly for path following by predecessors
Else
I=i+1
End of if
Repeat this step until job list becomes empty
 - Check the result in terms of Overall execution time_i and Individual schedule time_i
 - Initialize job list
 - End of generation loop
 - Output min(Overall execution time) and min(Individual schedule time) along with job sequencing
 - Output: Waiting time and fitness function
-

The proposed system gives the best possible solution in terms of overall execution time and individual schedule time. Overall execution time is observed to be time taken to complete schedule of jobs and individual schedule time is the time taken to complete one job. Result obtained through the proposed system is better as compared to individual firefly algorithm as described in the next section.

4 Performance Evaluation

Simulation of the proposed system is conducted in MATLAB 2017a. The simulation environment consists of five clusters containing 30, 30, 26, 26, 26 machines each. Job list is randomly initialized. Sorting operation is applied in order to sort the jobs within the queue on the basis of minimum cost first. For this purpose, burst time of the job is analyzed. Result is obtained in terms of overall execution time and individual schedule time. Simulation is conducted by varying the number of jobs within the queue. Simulation results are given as under.

Results in terms of Overall execution time

Comparison indicates that proposed system with optimal job first algorithm collaborated with firefly algorithm produces the better result as compared to existing literature without optimal job first approach (Figs. 1, 2, 3 and 4).

Job Scheduling—It is a process in which jobs are scheduled for processing with the help of scheduler so that all the jobs will get processor within minimum time and accomplish its task without more delay otherwise resource starvation or time delay problem can occur within the system.

The below plot indicates the job allocation process, i.e., the number of jobs that are coming in a given time to accomplish its task (Fig. 5).

The below plot indicates the firefly convergence which indicating the jobs meeting the resources (Fig. 6).

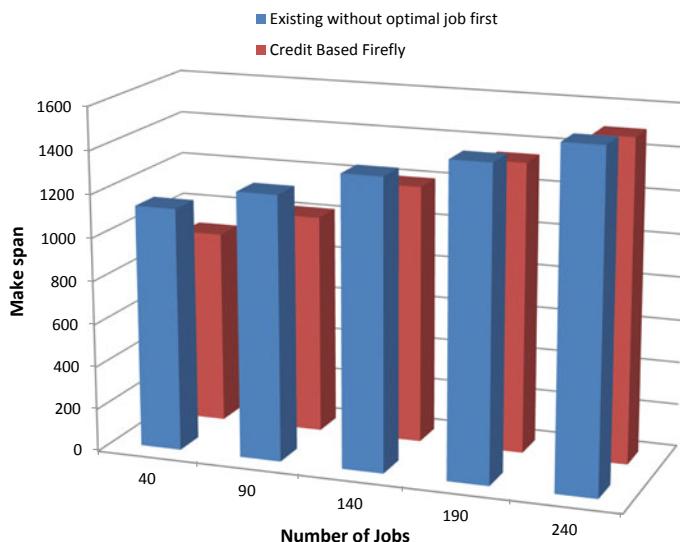


Fig. 1 Comparison of overall execution time of existing and proposed overall execution time

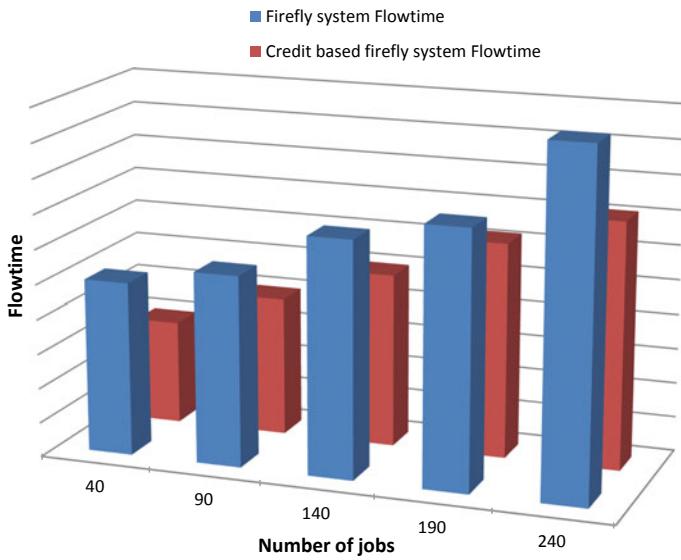


Fig. 2 Comparison in terms of individual schedule time

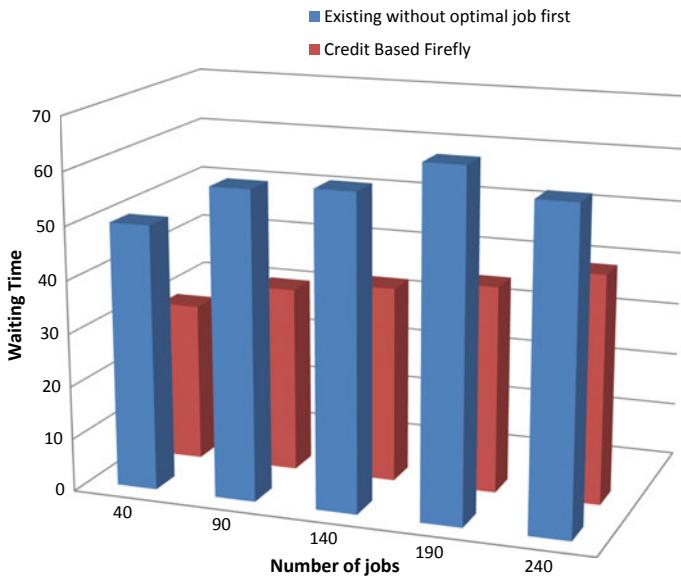


Fig. 3 Comparison of waiting time of existing and proposed waiting time

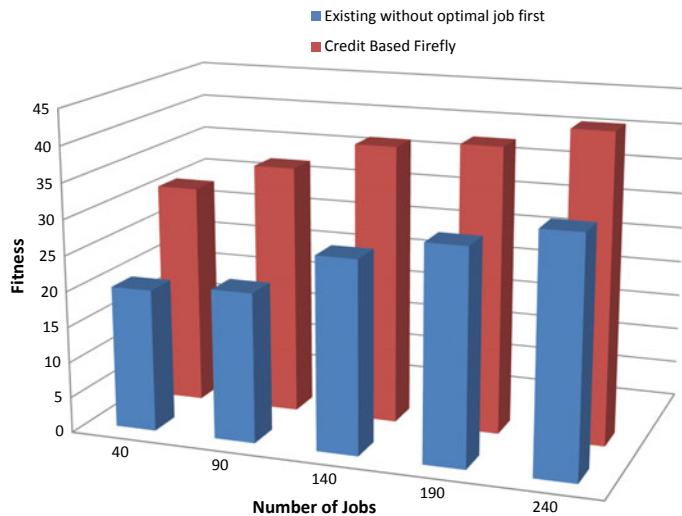
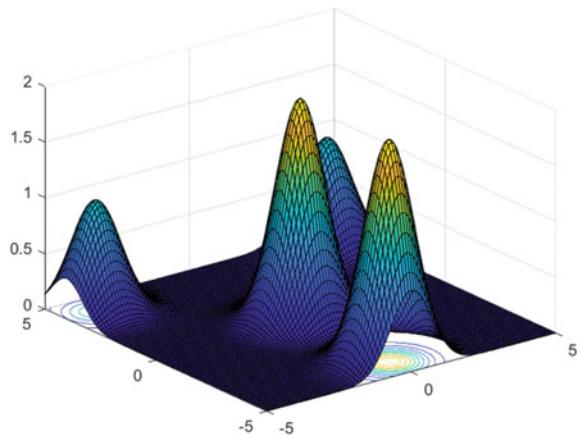


Fig. 4 Comparison of fitness of existing and proposed fitness

Fig. 5 Number of jobs coming to process



5 Conclusion and Future Scope

Job scheduling becomes need of the hour since resources are limited within advanced computing system. Job scheduling collaborated with multi-heuristic approach allows best possible solution to be generated; however, convergence rate is sometimes poorer. To tackle the issue, best possible solution with firefly approach is proposed. The simulation result shows optimization in terms of overall execution time and individual schedule time. Job sorting is initiated with the help of cost comparison.

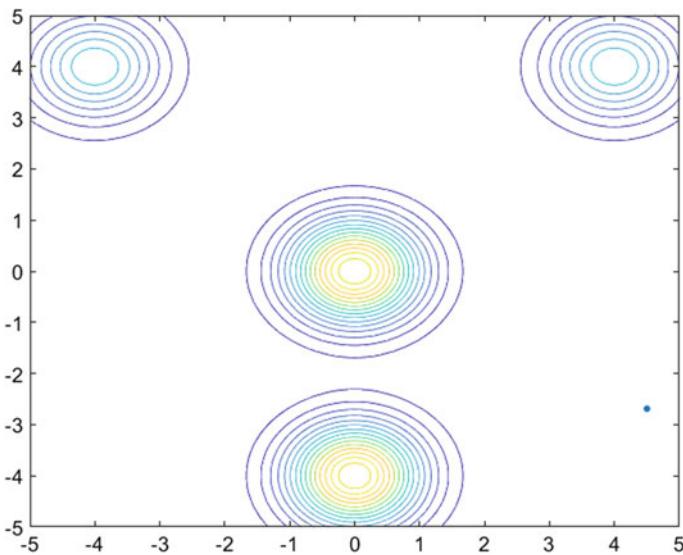


Fig. 6 Firefly convergence

List obtained after sorting is presented o the firefly algorithm for execution. Steps are repeated until desired level of optimization is met.

In the future, particle swarm optimization and genetic approach will be tested along with the proposed approach for optimization and performance comparison.

References

1. Cui H et al (2017) Cloud service scheduling algorithm research and optimization
2. Fidelis MB, Arroyo JEC (2017) Meta-heuristic algorithms for scheduling on parallel batch machines with unequal job ready times. In: 2017 IEEE international conference on systems, man, and cybernetics, SMC 2017, 2017-January, pp 542–547
3. Gonzalez NM et al (2017) Cloud resource management: towards efficient execution of large-scale scientific applications and workflows on complex infrastructures. *IEEE Access*
4. Gopalan NP, Suresh S (2015) Modified delay scheduling: a heuristic approach for Hadoop scheduling to improve fairness and response time. *Parallel Process Lett* 25(04):1550009. Available at: <http://www.worldscientific.com/doi/abs/10.1142/S0129626415500097>
5. Javanmardi S et al (2014) Hybrid job scheduling algorithm for cloud computing environment. In: Fifth international conference on innovations in bio-inspired computing and applications IBICA 2014, pp 43–52
6. Keskinturk T, Yildirim MB, Barut M (2012) An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times. *Comput Oper Res* 39(6):1225–1235. <http://dx.doi.org/10.1016/j.cor.2010.12.003>
7. Lin C. A new ant colony optimization for minimizing total tardiness on parallel machines scheduling 2:2–5

8. Lorpunmanee S et al (2007) An ant colony optimization for dynamic job scheduling in grid environment. *Int J Comput Inf Sci Eng* 1(4):207–214. Available at: <http://waset.org/publications/3389/an-ant-colony-optimization-for-dynamic-jobscheduling-in-grid-environment>
9. Singh SK, Vidyarthi DP (2015) Independent tasks scheduling using parallel PSO in multiprocessor systems. *Int J Grid High Perform Comput* 7(2):1–17
10. Tao Y et al (2011) Job scheduling optimization for multi-user MapReduce clusters. In: *Proceedings—2011 4th international symposium on parallel architectures, algorithms and programming, PAAP 2011*, pp 213–217
11. Wang M, Zeng W (2010) A comparison of four popular heuristics for task scheduling problem in computational grid. In: *2010 6th international conference on wireless communications, networking and mobile computing, WiCOM 2010*, pp 3–6
12. Zhang S et al (2016) Burstiness-aware resource reservation for server consolidation in computing clouds. *IEEE Trans Parallel Distrib Syst* 27(4):964–977. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7091926>. Accessed 3 May 2016

Chapter 10

A Novel Non-invasive Approach for Diagnosis of Medical Disorders Based on De Broglie's Matter Waves and Water Memory



Vijay A. Kanade

1 Introduction

In 1924, Lewis de Broglie proposed that matter possesses dualistic nature (i.e., dual characteristic) similar to radiation. This implies that moving matter shows wave-like properties (i.e., similar to diffraction, interference, etc.), and the matter at rest shows particle-like properties. The waves associated with moving particles are termed as matter waves or de Broglie waves [1].

Further, in 1990s, a Japanese scientist Dr. Masaru Emoto performed a series of experiments to identify the physical effect of words, prayers, music and environment on the molecular structure of water. Dr. Emoto exposed water to loving, kind, compassionate, fearful, discordant words. With kind and loving words, pleasing physical molecular formations in the water were observed. Further, on exposing water to negative and fearful words, ‘unpleasant’ physical molecular formations were observed. Some of his findings are presented in the images below [2] (Figs. 1).

The above illustration reveals that water has ‘memory.’ Further, human body contains about 70% water. Neutral water has no side effect and flows through the BBB to reach every human tissue within 30 min post drinking.

Thus, the proposed research work illustrates a novel treatment method that harnesses de Broglie’s matter waves along with water memory to provide an effective solution for various medical disorders. Specifically, the paper discloses water restructuring electronic device that imprints matter waves of a molecule (i.e., pill, drug) onto water.

V. A. Kanade (✉)

Intellectual Property Research, Pune, India
e-mail: kanade.science@gmail.com

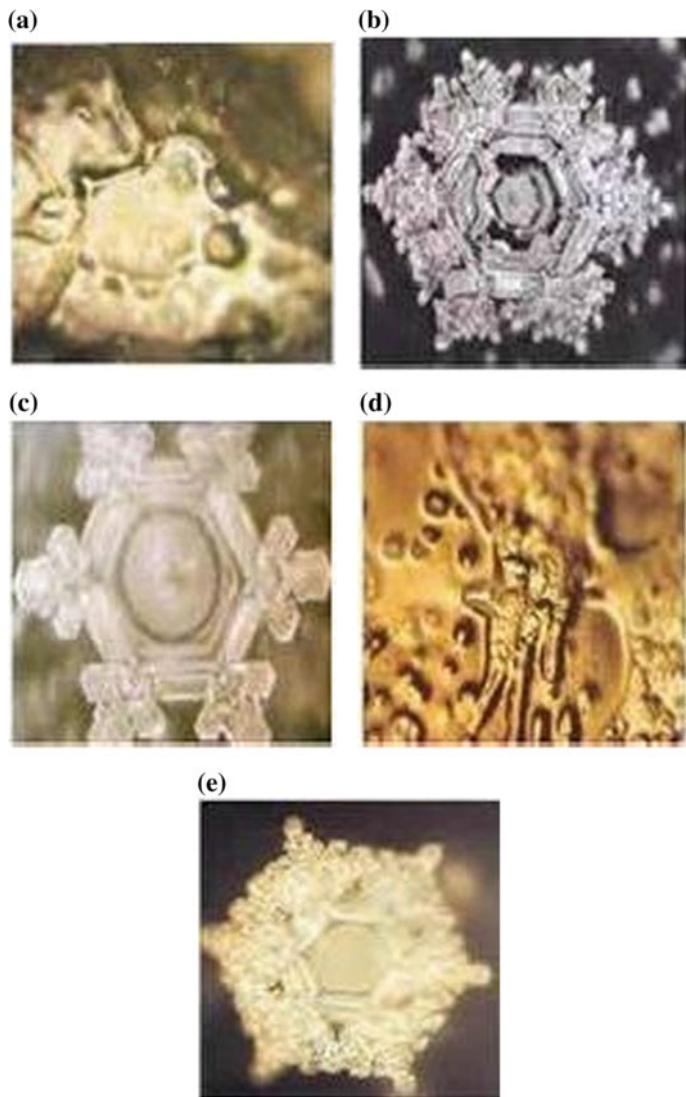


Fig. 1 **a** Water molecule, before offering a prayer. **b** After offering a prayer. **c** Water molecule, thank you. **d** You make me sick, I will kill you. **e** Water molecule, love and appreciation

2 Literature Survey

P53 is well known in the art as the tumor suppressor gene that codes for a protein which handles cell cycle regulation and thereby serves as one of the potent tumor suppressor protein. In a paper authored by Won H Kim, P53 matter wave was transferred to water by utilizing a carrier frequency of 7.8 Hz. In the study, information

wave of P53 was transferred to UM (mixture of ceramic balls which makes alkaline reduced water). The UM was then kept in contact with water to produce alkaline reduced water with P53 information [3].

In the above experiment, Kim used MDA-MB-435 and MDA-MB-231 cancer cell lines to validate the anticancer effect. These cell lines are known to induce tumorigenesis and metastasis.

The effect of P53 matter wave on cancer cells was assessed by maintaining the contact of MDA-MB-435 and MDA-MB-231 with P53 water for 1–5 days and monitoring their rate of proliferation. It was observed that both MDA-MB-435 and MDA-MB-231 cells showed significant decline in cell numbers. From the collected data, it was realized that water with P53 information almost blocked the cell growth. The observation thus demonstrated the tumor suppressing effect of P53 substance that worked against cancer cell growth [3].

In this research proposal, we intend to transfer the matter wave of a pill (i.e., drug) to water by utilizing an electronic device and observe its healing pattern on the target cell when the wave fuses with the appropriate cell receptors. The matter wave is transferred to water by using a pilot wave of the molecule under consideration. Further, since water possesses memory, the wave impressions on the water function like the real physical matter.

3 Pilot-Wave Theory

According to Pilot-wave theory, each particle has an associated physical wave that creates a pathway or track for the particles to follow [4]. R. M. Eisberg in 1961 revealed that a pilot wave of matter is faster than the speed of light. It was observed that for a mass particle traveling with a speed just below the speed of light, the pilot wave of the particle was faster than the speed of light [5]. This speed is given by the following equation:

$$w = c^2/v_p$$

where,

w Speed of pilot wave,

v_p Speed of particle,

c Speed of light.

According to the author of ‘Science and Human Transformation,’ pilot waves direct a mass particle by constantly moving through the matter from the back-end of the particle to the front-end, similar to water waves [6]. These pilot waves are termed as ‘information waves’ or ‘matter waves’ in the discussed research proposal.

4 Proposed Methodology

According to de Broglie's concept, every matter has an accompanying matter wave. The wave part of the matter contains information and functions like matter itself. The matter wave of matter (e.g., pills, drugs) can be transferred to water physically by shaking and diluting it. This has been used in traditional homeopathy to stimulate natural healing. However, the paper proposes an electronic device for transferring the matter wave to water distinct from the traditional homeopathic method. The device uses 7.8 Hz resonant frequency of earth to activate and transfer the information wave of matter to water via the pilot wave ($w = c^2/v_p$). The structure of the field created by the pilot wave in water is similar or related to that of original shape of the matter. The matter wave signature is thus imprinted onto water which exhibits a form of memory—implying, water retains the matter expression by changing its molecular structure. Further, we are also aware of the fact that water reaches every organ of the human body within 30 min of its ingestion. Thus, on consuming the above modified water, the molecular wave is transferred to the concerned cell receptors. The wave induces resonance into the receptors that initiate intra-cellular signal transmission leading to a neutralizing/suppressing effect. Thus, the matter wave of medically essential drug/pill stored in water could play a vital role in curing brain diseases like brain tumor, Alzheimer's, Parkinson's disease since the blood-brain barrier (BBB) allows the passage of water, however inhibits any drug from outside to pass through.

5 Experimental Setup

The experimental setup of the research is disclosed in this section, wherein observations have been drawn by correlating the proposed system with results disclosed in [3, 7]. The proposed work is novel in its applicability as the electronic device is distinctively designed for extracting the benefits out of medical pills/drugs. However, the research proposal is closely associated and linked to [3] since Kim's paper supports and validates the feasibility of the possible results that can be produced by the proposed system.

5.1 *Proposed System*

In the proposed technique, the matter wave associated with a pill or a drug is transferred to water by utilizing an electronic system that uses a carrier frequency of 7.8 Hz. The electronic system includes a 7.8 Hz frequency generator which generates a magnetic field around a flask that contains the pill. A coil is wrapped around the flask to activate pill, i.e., physical substance. Output of the first flask is fed to the signal amplifier where the output signal gets amplified and is relayed onto the second flask

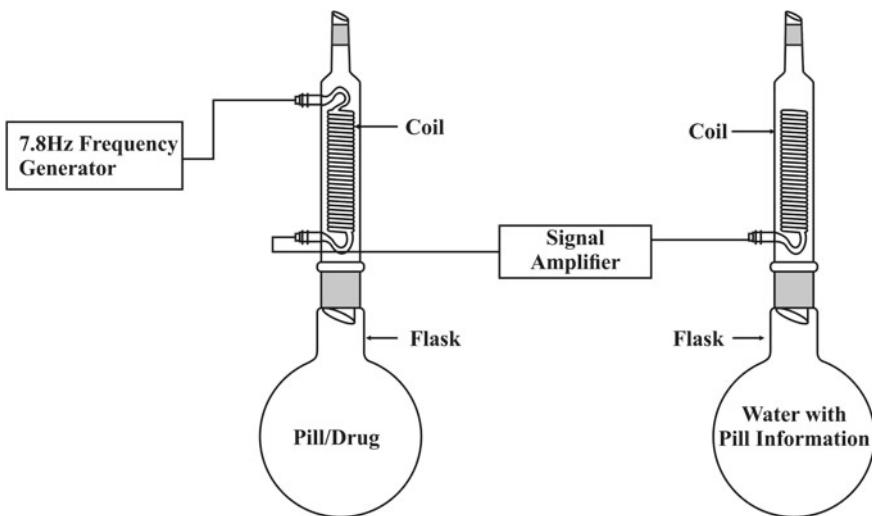


Fig. 2 Electronic apparatus transferring pill/drug matter wave to water

containing water. Activated information of the pill is thus transferred and imprinted onto water contained in the output flask where another coil is wrapped around the very flask. The coil on the second flask serves as a carrier of the electromagnetic data from first flask to the second.

The experimental framework of the proposed system is shown in Fig. 2.

5.2 Observations/Findings

The observations/findings pertaining to the proposed system are as disclosed here-with. The matter wave of the pill/drug (i.e., molecule) transferred to water exists as distinct expressions on water, similar to the original molecule. These expressions interact with target cellular receptors and induce intra-cellular signal transmission. Thus, the initiation of cellular signal transduction explains the phenomenon of ‘water memory’ which forms the basis of the research proposal.

5.3 Experimental Components

Table 1 discloses the specification factors of different experimental components that need to be considered while designing the water structuring electronic device. Customization of these specifications can be explored based on the output requirement (i.e., effect) desired from the electronic device.

Table 1 Specifications of the experimental setup

Experimental components	Specifications to consider
Flask (2)	≈250 ml
Frequency generator	7.8 Hz Magnetic signal (Earth's pulse)
Electromagnetic coil	Max. field Inductance Max. power Thickness Mass
Signal amplifier	Max. output level Output impedance Phase Power requirements Size (≈250 mm x 205 mm x 115 mm) Weight
Pill/drug	Any effective drug/pill/tablet/vaccine
Water	Normal/Mineral water Alkaline Reduced Water (for anticancer effect) [8]

6 Conclusion

Water is neutral by nature and has no side effects on its intake. Since human body contains significant amount (70%) of water, its memory and neutral nature could be utilized for our benefit. Thus, matter wave of any medically effective substance (drug/pill/tablet/vaccine) along with water memory could play a significant role in curing cancer, neurological diseases or any possible ailments that humans/living beings encounter. The novel approach therefore could shape the future of medical science in the coming years.

Acknowledgements I would like to extend my sincere gratitude to Dr. A. S. Kanade for his relentless support during my research work.

References

1. Weinberger P (2006) Revisiting Louis de Broglie's famous 1924 paper in the Philosophical Magazine. Philos Mag Lett 86(7):405–410
2. Masaru Emoto's Water Crystals. <https://www.scribd.com/document/313578632/Masaru-Emoto-s-Water-Crystals>
3. Kim WH (2013) New approach controlling cancer: water memory. J Vortex Sci Technol 1:104. <https://doi.org/10.4172/2090-8369.1000104>
4. Bennett J (December 1, 2016) The one theory of quantum mechanics that actually kind of makes sense
5. Eisberg RM (1961) Fundamentals of modern physics. Wiley, New York, pp 140–146
6. Tiller W (1997) Science and human transformation. PAVIOR

7. Kim WH (November, 2017) Water memory could be an effective and safe therapy for mind and brain related diseases. *Theranostics Brain Disorders* 2(4)
8. Lee K-J et al (January, 2004) Anticancer effect of alkaline reduced water

Chapter 11

Tamper Detection in Cassandra and Redis Database—A Comparative Study



Archana Golhar, Sakshi Janvir, Rupali Chopade and V. K. Pachghare

1 Introduction

Secure database is the key requirement of any enterprise. Database tampering is the act of interference by unauthorized user in original database. It causes unknown modification in database. To detect modification in database, forensics process is used. Database forensics is branch of digital forensics science, which is related to the forensics study of database [1]. Any change in the database can be detected by using the different techniques of database forensics. Any unauthorized person tries to access the data, or even the insider can modify the database purposefully with wrong intention. So, the confidential and private data gets hampered, and hence, the authorized users may get access to the wrong data. So, it is important to maintain the integrity of the database.

Log files can be used for tamper detection because every executable software, operating systems and programs has its own log file, which maintains the record of each and every action performed on the database [2].

Hashing is one of the methods used in database forensics for checking integrity of file. Hash value is unique for individual file, it is calculated after final operation on the database, and the computed value can be used for checking the integrity. Different

A. Golhar (✉) · S. Janvir · R. Chopade · V. K. Pachghare
Department of Computer Engineering and IT, College of Engineering,
Pune, Maharashtra, India
e-mail: golharas18.comp@coep.ac.in

S. Janvir
e-mail: janvirss18.comp@coep.ac.in

R. Chopade
e-mail: rmc18.comp@coep.ac.in

V. K. Pachghare
e-mail: vkp.comp@coep.ac.in

types of hashing algorithms are available like MD5, SHA-1, SHA-256, etc. In this paper, MD5 and SHA-1 hash algorithms are used for tamper detection.

Redis and Cassandra are both NoSQL database. Redis is in- memory data structure, which is used as a cache database, since it has append only feature for cache memory [3]. Redis enterprise is based on the simple client/server-based protocol over TCP. It keeps two log files [4], viz. RDB and append only file (AOF). Using this data structure, user can create strings, lists, hashes, sets and streams. Redis is being used by the companies like Apple, Zscaler, Comcast, Cisco, Snapchat, Verizon, Lockheedmartin, etc. Cassandra is a type of NoSQL database. It is used to store large amount of data across many servers. It comes under column family of NoSQL databases [5]. Cassandra contains a commitlog file, which contains data after final commit operation. Cassandra is always available, so it is used by organizations that cannot afford failure due to its feature of no single point of failure [6]. The big companies such as Facebook, Twitter, Cisco, Netflix, etc., are using Cassandra database.

2 Related Work

Wang et al. state [6] the importance of NoSQL database and described overview of Cassandra. Authors described various applications of Cassandra with companies which are using this database. Xu et al. [4] discussed the file format of Redis RDB backup file and AOF files. This paper implemented a prototype tool for extracting data in the RDB and AOF file. Kataria et al. [7] detected whether data is tampered or not, in Oracle 10g, using ORA_HASH function. To find tampering time, triggers are used. If security gets violated, there are steps to overcome almost all changes made in any database. Further, there are many forensics analysis algorithms which are explained in starting of the paper by authors. These algorithms find all the corrupted regions in database. The method explained here in this paper is very easy and basic. Kataria et al. state [8] that storing data in a database is not the only task but also we have to deal with all the issues related to database, and among these issues, one of the problems is database security, where security is major concern for database. This paper gives basic approach to detect whether data is tampered or not. And solution given in this paper is applying validation on data after fixed interval of time and check data integrity. Sindhu described [9] brief idea about the database forensics. It is given that database forensics is a forensics examination of database that relates to the timestamps of a row in a table which can be tested for verifying actions of a database user. Miclau et al. [10] described how we can build an application which prevents the integrity of a data. Authors showed that strong cryptographic hash functions give guarantees of integrity of data in database. Sindhu et al. [11] contributed for identification of general location in a database and file system for collecting digital evidence. Authors investigated this using the WinHex tool and recovered the files. Khanuja et al. [12] represented a survey which describes various beliefs upon database forensics. Also, this paper states various challenges and threats for different database. Rajguru et al.

[13] described the architecture of Oracle 10g and proposed design of forensics tool to detect tampering of the database. This paper also analyzed all information related to tamper. This methodology does not check the liability of the evidences. Pavlou et al. describe [14] the four different insights for tamper detection in database. Among these methods, one method is using hashing techniques which are used to detect tamper in a database. This states that hash value at commit time, transactions hash value and previous hash value can be added together to obtain new hash value, and in this way, hash value of each transaction is linked in a sequence [15]. Mario et al. [1] described the challenges and problems in database forensics, also focusing on current options for performing forensics on different databases. Yao et al. state [2] that audit logs are considered good practice for business systems and are required by federal regulations for secure system. A secure database must give guarantee of integrity of audit log files. Authors proposed a mechanism, based on hash functions that prevent an intruder, including an auditor or an employee or an unknown bug from silently corrupting the audit log.

3 Proposed Model

In Fig. 1, architecture of problem statement is given. It describes the exact execution of problem statement which is used in this paper. First, we must create database, and then, after creation, we must ensure that values are reflected in log file. Then, calculate hash value of file and use this value for checking integrity.

4 Methodology

Execution of whole problem is divided into four steps.

Step 1: Creating database

For performing experiment on dataset, we must create our dataset. In Cassandra, by using Cassandra Query Language (CQL), we can create our database [16]. CQL allows different types of operations such as CREATE, INSERT, ALTER, UPDATE, DELETE, etc., on database. For Cassandra, we have performed CREATE, INSERT, ALTER, UPDATE operations only, and DELETE operation is not performed here because it takes time to reflect changes in commitlog file.

Redis is a document type database which is created by adding various key-value pairs in the database [17]. We have created Strings, Sets, List and Hashes in Redis database and performed various operations on it to update the key-value pairs. Operations like SET, MSET, HSET, HMSET DEL, LPUSH, LPOP, RPUSH and RPOP [17] are evaluated on Redis database.

Step 2: Connect the database with the python environment.

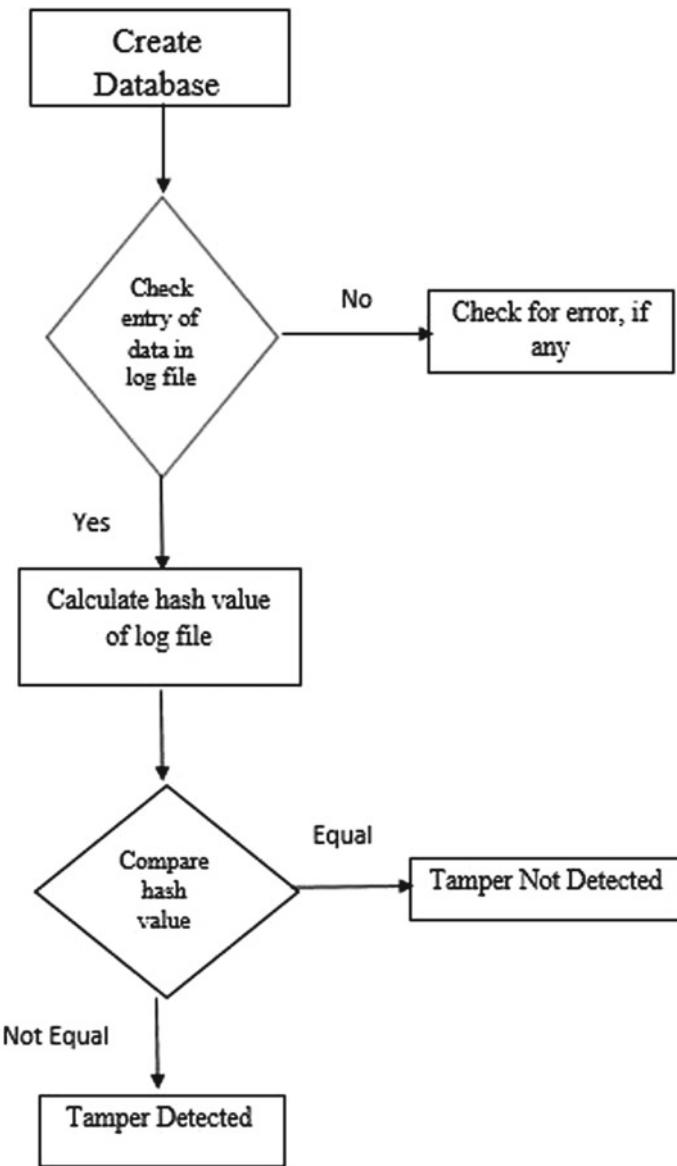


Fig. 1 Proposed model

For accessing the log file, we have installed the Redis and Cassandra set up in python and established the connection with database.

For setting up Redis connection in python 3.3, we have used the command mentioned below.

```
$python setup.py install
>>import redis
```

So that we can update the database remotely and able to evaluate the changes made in the RDB file.

For Cassandra, we have to install Cassandra driver and connect it with python 2.7 [18] only because other versions of python are not compatible with Cassandra.

```
>> pip install cassandra-driver master [20].
```

Step 3: Calculate the hash value of the log file using hash library

We have generated hash value of the log file using the hash library. We have used MD5 function and SHA1 function. Estimated time requirement in the generation of hash value is evaluated for each method.

Step 4: Tamper detection

After the update in the database, the corresponding hash value of the log file also gets changed. Changes in the log file denote the tampering of the data. Tamper detection notifies the organization about unauthorised activities, if there is no change in the calculated hash value, then log file is unchanged, and no modifications were performed in database.

5 Experiment and Result Analysis

The software requirement for this model is Redis 3.2, Cassandra 2.2.8 and python 2.7.

5.1 Tamper Detection in Redis Database and Time Required for Each Algorithm

Initially, Redis database is created by adding various key-value pairs of strings, sets, hash sets, lists [19].

```
Redis>LPUSH MYLIST "FIRST" A B C 1 2 3 4 5 6 7 8 9
```

This query will create list name MYLIST with 13 values identified by the index 0 (Fig. 2).

Generate hash value of the log file using the SHA1 and MD5 algorithm. Hash value for the Redis database is given in Fig. 3.

Update in the RDB file.

```
127.0.0.1:6379> lrange mylist 0 -1
1) "First"
2) "A"
3) "B"
4) "C"
5) "1"
6) "2"
7) "3"
8) "4"
9) "5"
10) "6"
11) "7"
12) "8"
13) "9"
```

Fig. 2 Created Redis list

```
In [208]: runfile('C:/Users/DELL/.spyder-py3/hashh.py', wdir='C:/Users/DELL/.spyder-py3')
SHA1 value :
58e6c9341eef61c85b11b9bd8d956a949e5e2c32
Time for SHA1 :
0.11684513092041016
MD5 hash value :
dd13701129013a0ccf55b57cfb766d27
Time for MD5 :
0.004238605499267578
```

Fig. 3 Hash value of log file

Now, user makes changes in the existing database by adding new content or removing the existing content of the database.

Redis>LPOP MYLIST

This query will delete leftmost value stored in the list “MYLIST”

Redis>RPOP MYLIST

This query will delete rightmost value stored in the Redis list “MYLIST”

Resultant Redis list will look like as in Fig. 4.

After updating the database, corresponding hash value of the log file also gets changed. Changes in the hash value denote the tampering of the data (Fig. 5).

```

127.0.0.1:6379> rpop mylist
"9"
127.0.0.1:6379> lpop mylist
"First"
127.0.0.1:6379> lrange mylist 0 -1
1) "A"
2) "B"
3) "C"
4) "1"
5) "2"
6) "3"
7) "4"
8) "5"
9) "6"
10) "7"
11) "8"
127.0.0.1:6379>

```

Fig. 4 Updated Redis list

```

In [201]: runfile('C:/Users/DELL/.spyder-py3/hashh.py', wdir='C:/Users/DELL/.spyder-py')
SHA1 value :
e46d7a5ff86467eda65ebbb39969c60f2bc66ed1
Time for SHA1 :
0.003997087478637695
MD5 hash value :
c0e4e86dec7c09a268878d4a0f2b4364
Time for MD5 :
0.003998756408691406

```

Fig. 5 Change in the hash value after modification in the database

5.2 A Tamper Detection in Cassandra Database and Time Required for Each Algorithm

Figure 6 shows the hash values of each algorithm. Both values are different, and time required for calculation of these algorithms is slightly different.

When we insert new data in the database, there is a change in the hash value of log file as shown in Figs. 6 and 7. From this, we can say that there is tamper detection in the database.

In this section, a graph is generated depending upon the time required for execution of the algorithm as shown in Fig. 8. We have updated the database by inserting the

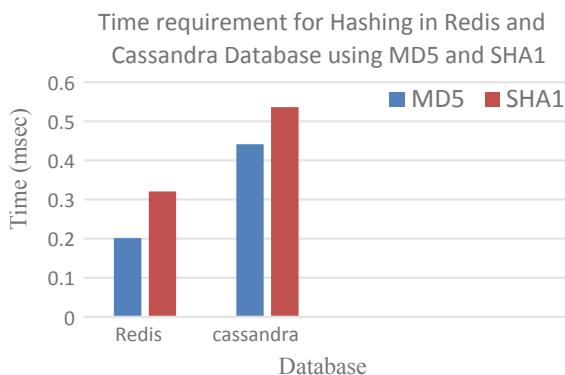
```
(7, u'Pragati', u'Nagpur')
(6, u'Anshul', u'jabalpur')
(9, u'Aditi', u'Pune')
(12, u'Pooja', u'Mumbai')
(3, u'Priyanka', u'Latur')
0
True
SHA1: 'a68a9f0103be867f69fda680219c1df14c44100d'
0.385999917984
True
MD5: '0f8c95a80e61d5663bf572330b0dd2c8'
0.481999874115
```

Fig. 6 Database of Cassandra with hash value log file and time required for calculation

```
(7, u'Pragati', u'Nagpur')
(6, u'Anshul', u'Nagpur')
(9, u'Aditi', u'Pune')
(12, u'Pooja', u'Mumbai')
(3, u'Priyanka', u'Latur')
0
True
SHA1: '2fd899c332e899042faee106e938808df0e8a4aa'
0.440999984741
True
MD5: 'aaed7ae1b84ec79cdf3c56794c3d8624'
0.536000013351
```

Fig. 7 Changes in the values after updating data in the database

Fig. 8 Time requirement analysis



equal number of entries in both the database and kept track of time required for calculating hash value.

6 Conclusion

Tamper detection in Redis and Cassandra database is new field with little literature. In this proposed system, MD5 and SHA1 hashing technique is applied on log file of database to detect tamper in database. As described in result section, integrity of the data is checked. From Fig. 8 graph, it can be observed that time required for hash calculation is more for Cassandra as compared to Redis. As not much work is done in tamper detection in these two databases, we can expand this idea to another level and can build a software which can be used for tamper detection. We can build a tamper proof database system.

References

1. Mario AM, Austin R, Huwida S (2010) Database forensics. ACM digital library
2. Yao SS, Christian C (2014) Tamper detection in audit logs Researchgate
3. Redis Tutorial. www.tutorialspoint.com
4. Xu M, Xu X, Xu J, Ren Y, Zhang H, Ning Z (2014) A forensics analysis method for Redis database based on RDB and AOF File. *J Comput* 9(11) (November 2014)
5. <http://cassandra.apache.org/>
6. Wang G, Tang J (2012) The NoSQL principles and basic application of cassandra model. In: 2012 international conference on computer science and service system
7. Kataria C, Kanwal G (2015) To detect who and when tamper data in database. *Int J Eng Res Technol (IJERT)* 4 (06) (June 2015) ISSN: 2278-0181 IJERTV4IS060187 www.ijert.org
8. Kataria C, Kanwal G (2015) Database tamper detection. *IJARCSSE*
9. Sindhu KK, Tripathi S, Meshram BB (2012) Digital forensics investigation on file system and database tampering. Research Gate
10. Miclau G, Dan S (2014) Implementing a tamper-evident database system. ResearchGate
11. Sindhu KK, Tripathi S, Meshram BB (2012) Digital forensics investigation on file system and database tampering. *IOSR J Eng (IOSRJEN)* 2(2):214–221 (Feb 2012) www.iosrjen.org ISSN: 2250-3021
12. Khanuja HK, Adane DS (2012) Database security threats and challenges in database forensics: a survey. ACM Digital Library
13. Rajguru S, Deepak S (2014) Database tamper detection and analysis. *Int J Comput Appl* 105(15) (Nov 2014) ISSN: 0975-8887
14. Pavlou K, Richard TS (2006) Forensics analysis of database tampering. ACM digital Library
15. Chopade R, Pachghare VK (2019) Ten years of critical review on database forensics research. *Digital Invest* 29 (April 2019)
16. https://docs.datastax.com/en/cassandra/3.0/cassandra/configuration/config_TOC.html
17. <https://radis.io>
18. <https://stackoverflow.com/questions/13217434/insert-to-cassandra-from-python-using-cql>
19. Redis in action. By Josiah L. Carlson foreword by Salvatore Sanfilippo
20. <http://datastax.github.io/python-driver/installation.html>

Chapter 12

Tamper Detection in MongoDB and CouchDB Database



Rohit Kumbhare, Shivali Nimbalkar, Rupali Chopade and V. K. Pachghare

1 Introduction

The enterprise or organization collects a large amount of valuable data, like customers, suppliers, competitions, etc. As per the information in the databases, the company can make a strategy for their further development and profit. Therefore, considering the information in the database of any organization is an important asset for company, thus to maintain its authenticity and integrity in database forensic is important role. It uses database forensic system which is related to study of databases and their related metadata [1]. For the forensic examination of a database, the data which is been inspected and should have a timestamp that is used to the update time of a row in NoSQL database and it also tested for validity in order to verify the actions of a database user. Moreover, a forensic examination may focus on identifying transactions within a database system or application that indicate evidence of wrong-doing, such as fraud [1]. Henceforth who, when and how the data is been tampered or modified. Log files can be used for tamper detection because every executable software, operating system and program have its own log file, which maintains the record of each and every action performed on the database. Hashing and Checksum are the methods which are used in the database forensic for checking integrity of a

R. Kumbhare (✉) · S. Nimbalkar · R. Chopade · V. K. Pachghare
Department of Computer Engineering and IT, College of Engineering,
Pune, Maharashtra, India
e-mail: kumbhareru18.comp@coep.ac.in

S. Nimbalkar
e-mail: nimbalkarsm18.comp@coep.ac.in

R. Chopade
e-mail: rmc18.comp@coep.ac.in

V. K. Pachghare
e-mail: vkp.comp@coep.ac.in

file. Hashing is a technique in which each file gets a unique hash code which can be used for checking its integrity. There are different types of hashing algorithm; some of them are MD5 and SHA. Checksum is a technique which is also used for tamper detection of a file. In this method, a transmitter generates a numerical value according to the number of set or unset bits in a document and sends it along with each document.

CouchDB and MongoDB are NoSQL database. CouchDB is HTTP-based REST API. This database is distributed and scalable that is it works for small set of documents as well as large set of documents [2]. In CouchDB [2], data is self-contained that is it contains everything it required like real-world document. As CouchDB is document-oriented storage, the documents are stored in JSON. Each document has its own id which is unique per database. MongoDB is a NoSQL database in which data is stored in the form of key-value pairs. It is a document-based database which provides scalability and sharding along with data modeling and data management of huge sets of data in an enterprise application with high performance [3]. MongoDB also provides [3] the feature of auto-scaling. Since, MongoDB is a cross-platform database and can be installed across different platforms like Windows, Linux, etc.

2 Related Work

Chahal et al. state that [1] what are security concerns in NoSQL, and the data which is at the rest should be protected by physical layer using some kind of encryption techniques. The data which is in motion like financial banking can use tagging from protecting from unauthorized person. Also, it has security vulnerabilities connection pooling and key brute forcing, i.e., it is key-value database, we can also create our own key but if unauthorized person gets key, then data will be big problem, for this data has to be segmented by application itself. That is called modeling the data.

Bhardwaj states [4] the comparative study of MongoDB and Apache CouchDB. The mechanism to store the data is different is document oriented database. Installation of both databases is easy but CouchDB installation is faster. Data storage in MongoDB is BSON and CouchDB is JSON. So learnt different aspects of both databases. In this paper, the performance evaluation of CouchDB as well as MongoDB is carried out.

Mariam et al. state that [5] (oracle 10g) tampering is one of the main issues to solve in database security. Oracle 10g is used to deal with problem of who, when and how tampered data. It shows results that the time when data got tampered and also the name of culprit. Here, ORA_HASH function is used to compute hash value for given expression which is unique for every tuple in the table.

Literature survey also explores the basics of architecture and vulnerabilities of MongoDB and CouchDB

MongoDB Architecture:

See Fig. 1.

The architecture contains [6]:

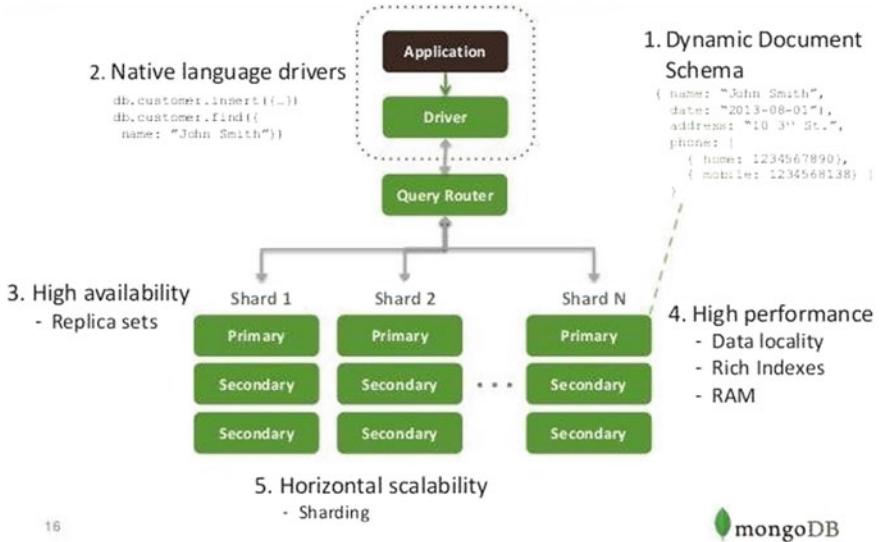


Fig. 1 MongoDB architecture [6]

Dynamic Document Schema: MongoDB provides a document data model that lets you store and combine any type of data in an unstructured format. With MongoDB, you do not have to change the database structure as it is dynamic.

Native language drivers: The drivers which are supported to mongo and which can help to interact mongo with other programming languages.

High availability: High availability refers to the improvement of system as well as application availability by minimizing the latency caused by maintenance operations and sudden system crashes. MongoDB provides high-availability cluster solutions along with several high-availability cluster configurations.

High Performance: MongoDB introduced a free performance-monitoring tool in the cloud for standalone instances and replica sets. MongoDB Atlas is one of the tools and it also provides cloud instance to the users.

Horizontal Scaling: Horizontal scaling involves dividing the system dataset and load over multiple servers as per the system performance, adding additional servers to increase capacity as required.

CouchDB Architecture:

See Fig. 2.

HTTP Request: By using this, the communication with database becomes very easy. It has simple structure of HTTP request also the commands such as GET, PUT, and DELETE are easy to use.

CouchDB Engine: It is NoSQL database which uses JSON to store data. It is document-oriented storage which has key-value store, each document will have a unique key, and using this key, we can retrieve documents.

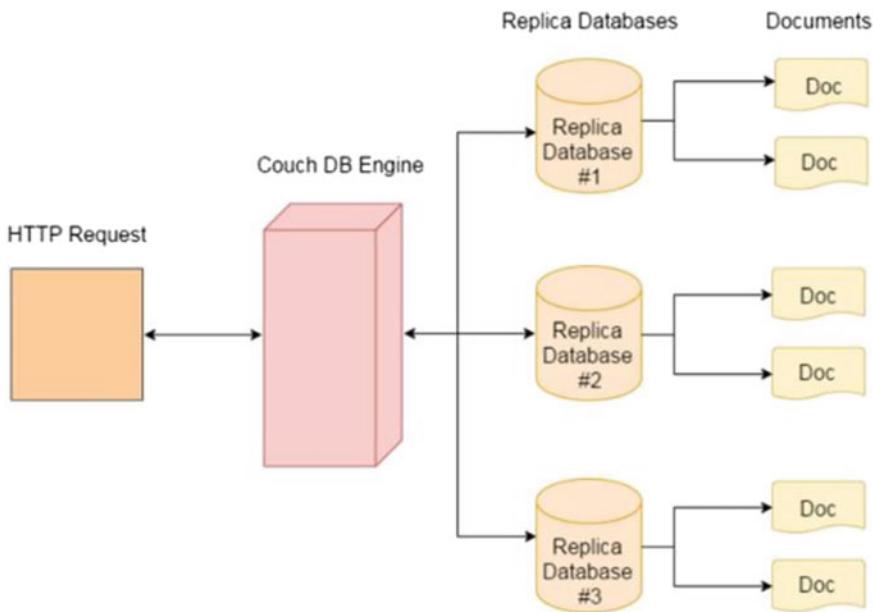


Fig. 2 CouchDB architecture [7]

Replication Database: CouchDB is replicated in different storage. It has replica databases, and, from this, we get to know if differences appear in database.

Documents: Each database has collection of documents in it. In CouchDB documents are self-contained. Documents have unique revision id assigned to it, which is used to retrieve the documents as well as check for tampering.

Vulnerabilities in NoSQL Database:

Authentication Weakness: By default, the DB installs with NO password credentials! Reading the NoSQL database manual the NoSQL developers have put the security entirely in the hands of the application developers and running it in a trusted environment.

Authorization Weaknesses: Any created user has read-only access to all documents in database. That essentially means that once you have a user, you have provided access by default to everything stored in the database...not the most secure.

Admin Authorization Weakness: A user who has given admin access also has access to the read/write commands as well. There is no granularity, i.e., level of detail in a set of data. By default, there is access to everything in database.

Clear Text: The data is sent in the clear form, because this data can be captured in an ARP poison attack.

Multiple Interfaces Weakness: The available interface will have its default service bind to it. So installing it in dual home environment is quite difficult. Exposing whole database to less trusted DMZ is risk factor too.

3 Proposed Methodology

For MongoDB and CouchDB, there are different proposed methodologies created as per the requirements.

In Figs. 3 and 4, the architecture of proposed methodology of MongoDB and CouchDB is given. It describes the exact execution of problem which is used in this paper. First, we have to create database then after creation, we must ensure that values are reflected in log file. Then calculate hash value of file and use this value for checking integrity.

MongoDB:

Input: Input contains the basic commands which performs the CRUD operation and saves the data in the MongoDB. There are various commands for different operation in MongoDB in which CRUD operation is basic.

MongoDB Database: It is an unstructured document-oriented database. It stores the data in document format. Collection of documents is called as collections and each collection has its own collection file saved in .wt extension format which is tamperproof externally but internally, using commands, we can change the database in collections.

MongoDBLogFile: The MongoDB log file contains all the logs regarding server connection, port connectivity, database queries and operations. All the DB logs are not visible as MongoDB due to different profiling level and authentication; to get all DB logs, we must set the profiling level to 2. Simple Task Manager: A simple task

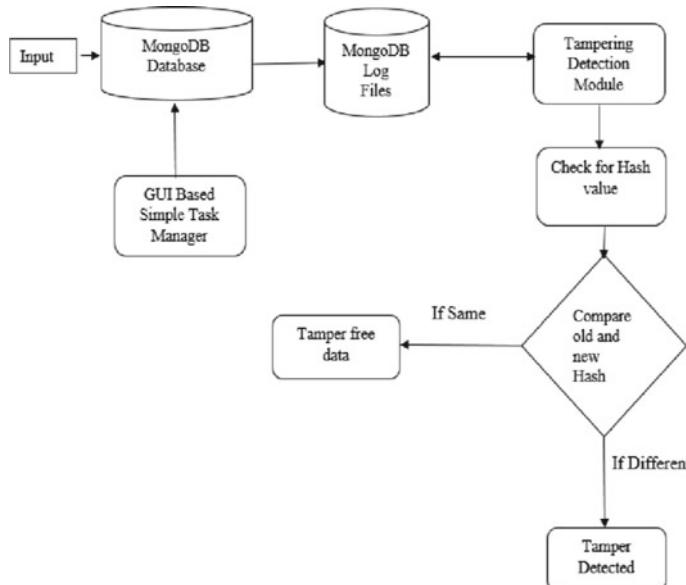


Fig. 3 Model for tamper detection in MongoDB

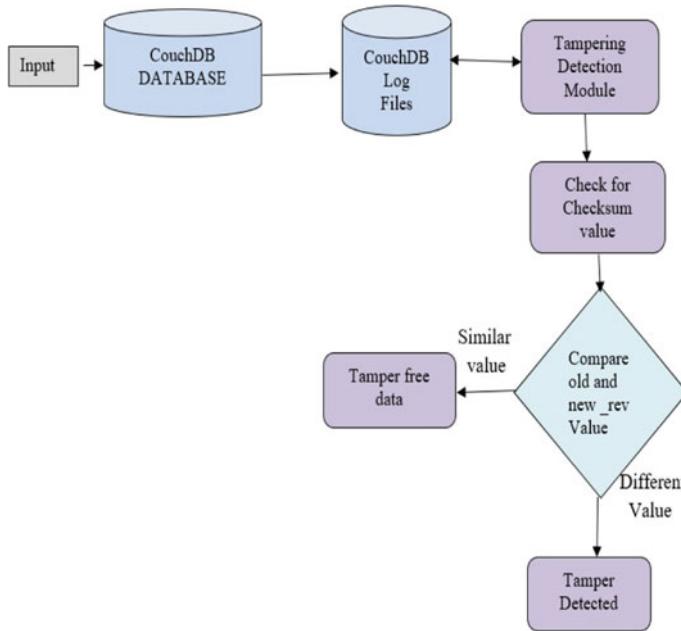


Fig. 4 Model for tamper detection in CouchDB

manager is been created to display the real-time CRUD operation and its logs. The logs are also saved in MongoDB.

Tampering Detection Module: It contains the Python code for generating the hash value of all collection and index files with .wt extension. Once all the Hash value is generated, it is been save in a text file. This text file is taken as an input for comparison with hash values generated after some period of time to check any tamper in database

Compare old and new Hash: When we add document, the collection gets modified, the hash value is generated but when unauthorized user tries to change the document then hash of that document gets change. So to check whether data is tamper or not, we compare the old as well as new hash of documents. **Tamper Detected:** If hash is appeared to be different, the unauthorized user tried to update the document. **Tamper-Free Data:** If hash appeared to be similar, then no data changes happen/no unauthorized access taken, and the database is tamper-free.

CouchDB:

Input: Input contains basic commands which performs the CRUD operation and saves the data in CouchDB. There are various commands for different operation in CouchDB in which CRUD operation is basic.

CouchDB Database: CouchDB is document-oriented NoSQL storage. It stores data in key-value format where each document will have a unique key assigned to it.

Once confidential data has been entered in database, its integrity and privacy must be protected on the servers where it resides.

CouchDB Log Files: Log contains server connectivity and port connectivity. All the inserted documents will be stored in log files also the updates/modifications done in the documents. Suppose the document contains id, name, age, and designation, then all other fields will be in encrypted format, it will only show id in log files. Available levels in log files are debug, info, error, warning, alerts, etc.

Detection Module: It contains the Python code for generating the checksum value of all the documents which is unique for all the documents in log files. Once all checksum value is generated, it is saved in text file, it is then taken as input to compare after some period of time to check whether someone tampered data in log file.

Compare old and new _rev: when we add document, the _rev generated but when unauthorized user tries to change the document then _rev of that document gets change. So to check whether data is tampered or not, we compare the old as well as new _rev of documents.

Tamper Detected: If _rev is appeared to be different, the unauthorized user tried to update the document.

Tamper-Free Data: If _rev appeared to be similar, then no data changes happen/no unauthorized access taken and the database is tamper-free.

4 Experiment and Result Analysis

MongoDB Experimentation:

1. Change the profiling status of the MongoDB for the user.
2. Writing the commands in command prompt to create the collections and documents such that it would be used to calculate the hash and to run a project to show the logs of basic crud operations
3. There is always a need to check the profiling status of MongoDB to get the logs according to the client's needs (Fig. 5).

MongoDB Result Analysis:

1. The Python code is used to run basic CRUD operation of MongoDB.
2. The Python created HTML will display the basic crud operation along with the insertion operation and deletion operation.
3. As there is a separate wt extension file for each collection and index, the hash value is generated as per the document creation (Fig. 6)

CouchDB Experiment:

1. Checksum field is created with each document.
2. The checksum of prior row is stored in new row of checksum.
3. Then, to verify the content of document, walk through the database computing the checksum as we move forward and then compare both rev generated.

```

Python 3.5.4 Shell
File Edit Shell Debug Options Window Help
----- RESTART: C:\ELT\SYSTEM FILES\MONGODB\SELEVENT\7.0\data\mongodump.py -----
collection-0--4434731558351645685.wt 843890757ca01150c4edc8967b410a9b
collection-0--7288630082128086404.wt 19921b253230abc02f7c4276c0262ace
collection-0--7693094748297505261.wt 4c69c95efa1890ad9948761a85c30589
collection-0-2524198644758996101.wt c8a039273cdf6f59fecbb7774e33283b
collection-10--7288630082128086404.wt 9af97f7280e40075ffb0575fe2baebcc
collection-12--7288630082128086404.wt 07ab756f01blf4a83151a2f50de0c6d
collection-14--7288630082128086404.wt 3d1c4434eebb9clbdc4377b3566870ee
collection-16--7288630082128086404.wt 9af97f7280e40075ffb0575fe2baebcc
collection-2--4434731558351645685.wt 8e8c5cc9f39511223d292d3ffae1e802

```

Fig. 5 Hash values of collection and indexes



Fig. 6 Task manager of MongoDB

4. Like in above example, the checksum, i.e., revision id gets changed.
5. If the calculated checksum does not match the value in the table, then some value has tampered. Likewise it will check for tamper detection.

CouchDB Result Analysis:

1. Tampering done by unauthorized user (Fig. 7).
2. Tamper detection (Fig. 8).
3. Change in checksum(_rev id) (Fig. 9)

```

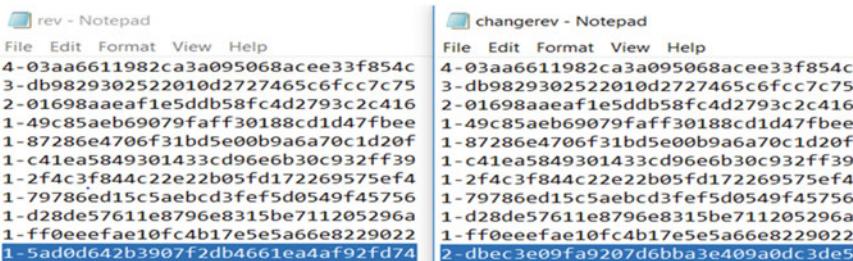
C:\Users\Shivali\Desktop>curl -X PUT http://127.0.0.1:5984/mydb/011 -d "{'Name': 'Shivali NW', 'age': '23', 'Designation': 'Tech student', 'Account No.': '1234567890'}"
{"ok":true,"id":"011","rev":1-5a086542b39072d461ea4af92f074"}
```

```

C:\Users\Shivali\Desktop>curl -X GET http://127.0.0.1:5984/mydb/011
{"_id":"011","_rev":1-5a086542b39072d461ea4af92f074,"Name": "Shivali NW", "age": "23", "Designation": "Tech student", "Account No.": "1234567890"}
```

Fig. 7 Tampering by unauthorized user

```
C:\Windows\System32\curl-7.63.0-win64-mingw\bin>python C:\Users\Shivali\Desktop\other.py
Tampering is detected at document id 011
<Document '011'@'2-dbec3e09fa9207d6bba3e409a0dc3de5' {'Name': 'Sonali MN', 'Account No.': '0987654321'}>
```

Fig. 8 Tamper detection**Fig. 9** Change in checksum

5 Conclusion

In today's world technology, how the tamper detection would help to capture any changes in database will show clear indication of changes happen. We presented some of the most important aspects of "Tamper Detection in database" with particular focus on what is being done and what are the issues that require further research. Security is the major concern in every field so as in the field of database. Database forensic field provides algorithms to find who and when tampered the data. If security gets violated, there are steps to overcome almost all changes made in any database. In a nutshell, this report is intended to draw attention toward database forensics with the hope of stimulating research in tamper detection which is one the most important part these days used for tamper detection for tamper-proof system.

References

- Chahal D, Kharb L, Gupta M, Challenges and security issues of NoSQL databases 2(5) ISSN: 2456-3307
- <https://www.javatpoint.com/couchdb-tutorial>
- <https://www.javatpoint.com/mongodb-tutorial>
- Bhardwaj ND (2016) Comparative study of CouchDB and MongoDB—NoSQL document oriented databases 136(3) (February 2016)
- Nair SM, Roy R, Varghese SM, Performance evaluation of MongoDB and CouchDB databases 3(7) Print ISSN: 2395-1990
- <https://www.mongodb.com/mongodb-architecture>
- <https://docs.couchdb.org/en/stable/intro/index.htm>

Chapter 13

Recommender System in eLearning: A Survey



Pradnya V. Kulkarni, Sunil Rai and Rohini Kale

1 Introduction

India is known for academics from ages. Conventional teaching–learning in ancient time in India was one of the best in the world. Since 700 B.C, Takshashila Nalanda University was the first university across the world which was established. The classroom teaching–learning method was adopted by Indians. But this kind of learning was limited and did not have exposure all over the world. Multiple issues [1] are available like old-fashioned syllabus, lack of hands-on learning, shortage of best tutors, and other issues formed the core of the problem. However, in the past few years, one could certainly observe instructive helpfulness progress with the support of digital means. eLearning has played the role of a catalyst for bringing about an equitable approach to high-quality education. It has done so by providing unlimited teaching and learning opportunities [2, 3], while also improving student's learning outcomes, participation, and pedagogical innovation tremendously.

Recommender system methods have been an actual approach to overwhelmed information burden [4]. Explosive development of data or material available on the network. Users of eLearning are frequently greeted with uncountable products and eLearning materials. So, customization is the required plan for providing a great user experience. This type of recommender tools is significant tools in different numerous Web domains like eLearning Web sites. This paper addressed the challenges of

P. V. Kulkarni (✉) · S. Rai · R. Kale
Department of Computer Science and Engineering, MIT School of Engineering, MITADT
University, Rajbaug, Loni, Pune, India
e-mail: pradnya.kulkarni@mitpune.edu.in

S. Rai
e-mail: sunil.rai@mituniversity.edu.in

R. Kale
e-mail: rohini.kale@gmail.com

eLearning recommendation systems. Some of the challenges in eLearning are information overload and relevant information; correct information should be provided to learners. There should be a common platform for students for eLearning. eLearning has been established [5] since many years with the help of teaching principles. This gives lots of benefits to learners. To promote eLearning, many colleges, universities, businesses, and organizations worldwide provide students distance learning courses, online certifications, and online degree. MIT OpenCourseWare, Learndirect.com, NPTEL, and MOOC [5] provide and announce many online courses and certifications. Perhaps, increasing to this personalization, education systems exceed the quantity of data or material that students have to use before they are able to decide their need. Recommendation system is one of the solutions to reduce the information burden.

Each learner has his own ability to deal with complexities, pace of learning, and integrating vast knowledge with appropriate correlation. eLearning recommendation system is a good tool for enhancing individualized learning. In this review, we aim to present the detailed study of the eLearning and eLearning recommendation system challenges and methodologies and also the survey of available technologies for recommendation systems. The system using these technologies is required for enhancing the learning.

The following sections of this paper are as follows: Sect. 2 presents the literature study about eLearning technologies, trends, and approaches of recommendation system, Sect. 3 presents discussion, and Sect. 4 presents conclusion.

2 Literature Review

A. eLearning technologies and trends

eLearning recommendation system is widely used nowadays for enhancing learning. eLearning gives us the opportunity to learn almost from anywhere (at home, colleges, schools, universities, during traveling, in the garden) [1] at any time and from any devices (desktop, laptops, iPad, smartphones, etc.). Education system is improved with the help of Internet and digital learning by means of eLearning. eLearning is based on Learning Management System [2, 3] (LMS). Several colleges and organizations started using LMS for providing various educational needs. eLearning has been developed for individuals (for single user), for multiusers, for similar users, and for hybrid user. Nowadays, eLearning has become an important part of the education system [3, 6, 7].

The advantages of using eLearning are:

- User can learn from anywhere and at any time, that is, worldwide connectivity.
- Learning access to user is fast.
- eLearning can support any communication language, that is, language flexibility.
- User can have flexibility in time, place, and language.

- Courses and learning materials can be created easily, can be upgraded, and can be revised.

The author gave emphasis on all the types of learning like virtual learning, classical classroom learning, and blended learning. It compares all the three leanings and shows the significant relation between them. Use the technology to improve effectiveness, efficiency, and convenient learning was addressed by author [8].

By combining more representative side information into the recommender system, recommendation precision can be improved. These systems can help for filtering the right information to the user's commerce (Amazon), movie recommendation (Netflix), and eLearning (Byju, Gooru) which are some of the examples of recommendation systems. Traditionally, existing methods for recommendation systems are roughly of three classes: content-based methods, collaborative filtering (CF)-based methods, hybrid methods.

Figure 1 explains the collaborative filtering and content-based filtering. Table 1 gives an overview of different algorithms used under collaborative filtering, content-based filtering, and hybrid filtering. It also tells pros and cons of the three methods.

Most of the recommendation system like eCommerce (Amazon) and movie recommendation (Netflix) makes recommendations based on feedback [4, 9, 10].

There are two types of feedbacks: explicit feedback and implicit feedback.

Recommendation system in an eLearning circumstances is a software means that tries wisely recommending activities to a student which are using their previous knowledge of activities. Recommendation in eLearning recommendation system [11] is done based on feedback from users. In Table 1, we summarize the feature-wise comparison of feedbacks, that is, implicit feedback and explicit feedback.

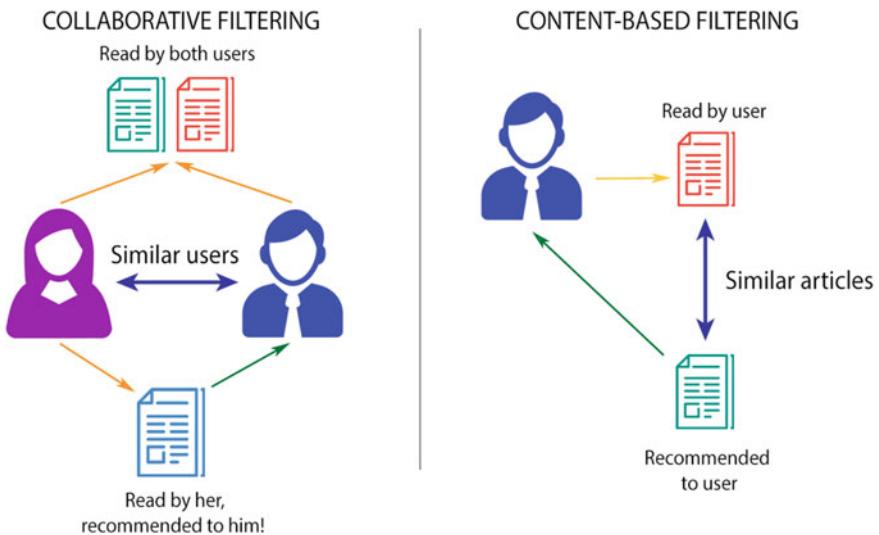


Fig. 1 An overview of recommendation systems

Table 1 Overview of recommendation system techniques

Techniques	Representative algorithm	Advantages	Disadvantages
Memory-based collaborative filtering	1. User-based CF 2. Item-based CF	1. Easy to implement 2. Data addition is simple 3. Need not consider content	1. Depends on explicit feedback 2. Cold start problem and sparsity problem 3. Limited scalability for large dataset
Model-based collaborative filtering	1. Slope-one CF 2. Matrix factorization	1. Good scalability 2. Improve prediction performance 3. Improve scalability and sparsity problem	1. Model is expensive 2. Loss of information in matrix factorization
Hybrid collaborative filtering	1. Combination of memory-based and model-based	1. Overcome limitations of sparsity 2. Improve prediction performance	1. Increased complexity 2. Complicated for implementation
Content-based filtering	1. Content-based filtering algorithm using Hidden Markov Model	1. No scarcity and cold start problem 2. It ensures privacy	1. Needs detail information of items 2. Needs well-prepared user profile

In an explicit feedback, learner explicitly rate courses. In an implicit feedback, learners' actions are monitored like browser history, time spend on Web page, mouse movement, bookmarks, click-stream behavior of the learners. The proportion of actual studying hours to the total hours of the course is recorded as the implicit rating scores and transformed to corresponding explicit rating scores from 1 to 5 rating [12].

The system addressed normally gathers the actions of the users in the systems including evaluation, clicking, purchasing, remarks, etc. Content-based methods make use of user's personal information or product descriptions for recommendation. Approaches depending on collaborative filtering use the previous actions or preferences, such as user evaluation on items, without using user or product content information. Hybrid methods seek to get the best of both worlds by combining content-based and collaborative filtering-based methods. To produce the appropriate recommendations and ensure the real-time requirements of the system, researchers proposed several algorithms [12]. Collaborative filtering is a proven method normally used by various recommender systems. The user-based collaborative filtering algorithm is chosen as the primary recommendation algorithm, combined with online education [1]. Historical data is gathered by partially seen user and is one of the ways to complete matrix in collaborative filtering. For matrix completion, this paper introduced maximum margin factorization methods using a factor model, the singular value decomposition (SVD) method, SOFT-IMPUTE based on regularized

nuclear norm minimization, etc. Recommendations are made using evaluation provided to items by users as the source of information which uses predictable CF-based approach. Since the ratings are often very light in many applications, it causes CF-based methods to discredit extensively in their recommendation performance. User evaluation on items, historical actions, or preferences which would not be used the user or product content information is used in Collaborative Filtering based methods.

In this paper, three-level hidden Bayesian link prediction (3-HBP) model is addressed. It is a link prediction model for interpreting user behavior and user relationships [13]. Usual recommendation system allows learners to search novel learning resources that match their requirements and allows other eLearning system to focus the learning resources to the right learners. Content-based recommendation algorithm based on convolutional neural network (CNN) was used [14]. Content-based recommendation methods use characteristic units of users and items from their personal information. This information is used to recommend appropriate item to the user. These items are matching in content to items and user attributes [1].

User-item interaction is an important approach used in recommendation. Collaborative filtering is broadly used in different areas as it relies on user-item interaction history. Neighborhood-based and model-based are the two types of collaborative filtering approaches. The comparison between them lies in how to use the user-item evaluation. The former directly uses the stored ratings in the prediction. The later uses these ratings to learn a predictive model. Currently, Matrix factorization (MF) is one of the most popular model-based collaborative filtering methods. Matrix factorization techniques, including principal component analysis (PCA), singular value decomposition (SVD), regularized matrix factorization (RMF), and latent Dirichlet allocation (LDA) [13], have been in particular well implementation to recommender systems. But in this method, data sparsity is a problem. Deep hybrid recommender system based on auto-encoders can be used to incorporate more auxiliary information like knowledge ontology and images. In this paper [12], authors proposed a novel deep hybrid recommender system based on auto-encoders. In their paper, author proposed a hybrid recommender system and improved the performance of recommendation system by using deep learning. Denoising auto-encoders are used with neural collaborative filtering which corresponds to the process of learning user and item features from the secondary information to decide user preferences. Implicit feedback is used which comprises user and item from the secondary information to know the features of users and items. Recommendation precision can be further improved by combining more representative side information into the recommender system.

Works done till now have not utilized various side information in a widespread manner and take the full advantage of the available data. Deep learning would be used to design better inductive biases in an end-to-end fashion. This paper provides an ample review of the current research efforts on deep learning-based recommendation systems. Author provides hot developments and new viewpoints affecting to new exciting development in the field [15, 16].

Traditional collaborative filtering approaches use the feedback given to items by users as the entire information for making recommendation. Feedback or ratings given are very sparse, and it lowers the performance of recommendation systems.

In this paper, authors proposed hierarchical Bayesian model which is collaborative deep learning (CDL). It combines deep learning for the content knowledge and collaborative filtering for feedback matrix. It significantly improves the recommendation system performance. Convolutional neural network (CNN) is also one of the alternatives for collaborative deep learning (CDL) [17].

3 Discussions

To enhance learning, eLearning can be used in an efficient manner. eLearning recommendation system plays an important role in providing accurate and right information to the user. To recommend any course to the user, implicit and explicit feedbacks have to be considered. To avoid some of the shortcomings of explicit feedbacks as mentioned in Table 1, implicit feedback parameters [4, 9] are used like browsing/activity history, mouse movements, click history, event time, user name, and email ID. Different issues with recommendation systems can be handled like:

1. Cold start problem
2. Warm start problem
3. Repeat user.

All default users are considered as new users. As the user is new, nothing is known about him like behavior pattern and history of browsing. Tracking of such users and address them is a problem. This is one of the challenges. Something can be known about the warm start users, and everything can be known about the repeated user. eLearning recommendation system helps learners to decide the right choice without adequate personal experience of the substitutes, and it is considerably necessary in today's information explosion age. eLearning recommendation system tries to help the learners in better manner and in required manner. This recommendation mainly focuses on online activities and online learning. Many researchers have been done in recommendation systems for different applications like movie recommendation [18], ecommerce, etc., and now recommendation in eLearning application is widely used. To achieve eLearning, Learning Management Systems (LMS) [3] are used.

eLearning can be achieved with the help of LMS platforms like Moodle, Schoology, and Docebo LMS. Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open-source software for Learning Management System. It is highly adjustable to different situations and mountable easily from single, one-off uses on a particular course to serve the needs of large universities. It is an open-source LMS [7]. Moodle as a platform for Learning Management System has functionalities as:

- eAdministration
- eContent Management and Delivery
- eLearning
- eAssessment

Deep learning can effectively use in eLearning recommendation systems to get better results. Advantages of using deep learning technology in recommendation systems:

- Direct feature extraction from the content.
- Heterogeneous data handled easily.
- Dynamic/sequential behavior modeling with RNNs.
- More accurate representation learning of users and items.
- Recommendation system is a complex domain. Deep learning worked well in other complex domains.

Session-based recommendations with RNNs. A significant work has been done on eLearning recommendation systems but still is a naïve field in which a lot has to be contributed like topic-wise recommendation to individuals, not only advanced topics but also prerequisites can be recommended by checking user's capability through implicit feedback. Development of eLearning recommendation system using deep learning and using trust and other aspect can be one of the challenges. The importance will be on the better performance of the learners. eLearning recommendation system can be used to find out the best-performing users through the multiple-choice question test results as well as low-performing users also can be tracked. Depending on their performance, the next course or the course material can be recommended to the user.

4 Conclusion

This study provides an introduction to recommendation systems and the specialized field of eLearning recommendation system. Importance was given on the well-known approaches applied in this area till now. A significant work has been done on eLearning recommendation systems but still is a naïve field in which a lot has to be contributed in the future. Despite the progress made in the area, there lies a lot of scope for improvement. Development of eLearning recommendation system using trust and other aspect can be one of the researches for future. The importance will be on the better performance of the learners. Deep learning can be used in eLearning recommendation system based on implicit feedback and user profile. eLearning recommendation system's accuracy and relevancy should be improved. Recommendation and personalization in eLearning are important approaches for combating information overload as well as to enhance learning.

Acknowledgements The authors wish to thank Dr. Mangesh Bedekar, HoS, School of CET, MITWPU, Pune, for his valuable ideas that helped in refining the superiority of this paper. Support for this work has been given by MITSOE, MITADT University, Pune.

References

1. Wu H-J, Huang S-C (2007) A dynamic e-learning system for the collaborative business environment. In: Seventh IEEE international on advanced learning technologies, ICALT 2007

2. Pireva K, Imran AS, Dalipi F (2015) User behavior analysis on LMS and MOOC. In: IEEE conference on e-Learning, e-Management and e-Services
3. Patel C, Gadhavi M, Patel A (2013) A survey paper on e-Learning based learning management systems (LMS). *Int J Sci Eng Res* 4(6) (June 2013) ISSN: 2229-5518
4. Lai C-H, Liu D-R, Lin S-R (2018) Document recommendation with implicit feedback based on matrix factorization and topic model. In: Proceedings of IEEE international conference on applied system innovation
5. Aggarwal D (2009) Role of e-Learning in a developing country like India. In: Proceedings of the 3rd national conference; INDIACom-2009 computing for nation development, February 26–27, 2009
6. Irene K, Zuva T (2018) Assessment of E-Learning readiness in South African Schools. In: International conference on advances in big data, computing and data communication systems
7. Koneru I (2017) Administering MHRD guidelines-compliant eAssessments through Moodle. In: 5th national conference on E- Learning & E-Learning technologies (EELTECH). IEEE, New York
8. Suleri JI, Suleri AJ (2018) Comparing virtual learning, classical classroom learning and blended learning. *Euro J Sustain Develop Res* ISSN: 2542-4742
9. Wu G, Swaminathan V, Mitra S, Kumar R (2017) Digital content recommendation system using implicit feedback data. In: IEEE international conference on big data
10. Jawaher G, Szomszor M, Kostkova P (2010) Comparison of implicit and explicit feedback from an online music recommendation service. In: Proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems, Held at the 4th ACM conference on recommender systems, RecSys, pp 47–51
11. Tan H, Guo J, Li Y (2008) E-Learning recommendation system. In: International conference on computer science and software engineering
12. Liu Y, Wang S, Shahrukh Khan M, He J (2018) A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering. *Big Data Min Anal* 1(3):211–221 (September 2018) ISSN: 2096-0654 03/06
13. Xiao Y, Li X, Wang H, Xu M, Liu Y (2018) 3-HBP: a three-level hidden Bayesian link prediction model in social networks. *IEEE Trans Comput Soc Syst* 5(2) (June 2018)
14. Shu I, Shen X, Liu H, Yi B, Zhang Z (2018) A content-based Recommendation algorithm for learning resources. *Multimedia Syst*
15. Zhang S, Yao L, Sun A, Tay Y (2018) Deep learning based recommender system: a survey and new perspectives. *ACM Comput Surv* 1(1): 35 p Article 1 (July 2018)
16. Liu J (2017) Deep learning based recommendation: a survey. In: ICISA, March 20–23 2017 in Macau China
17. Wang H, Wang N, Yeung D-Y (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21st ACM SIG KDD international conference on knowledge discovery and data mining. ACM, New York, NY, USA, pp 1235–1244
18. Taheri M, Irajian I (2018) DeepMovRS: a unified framework for deep learning-based movie recommender systems. In: 6th Iranian Joint Congress on fuzzy and intelligent systems. IEEE, New York

Chapter 14

A Realistic Mathematical Approach for Academic Feedback Analysis System



Onkar Ekbote and Vandana Inamdar

1 Introduction

Students' feedback has been demonstrated improve learning process. Teachers get to know about their teaching skills and overall opinions about them. Teachers can understand about both positive and negative aspects of their performance. Students also think that their opinions are considered somewhere. It helps to bridge a gap between teachers and students and improves overall learning process. But it has been observed that, students sometimes give the feedback with less attention, and the actual, real feedback won't get registered. Such feedback doesn't help much in improving academic system. Hence, it is important to get feedback as realistic as possible.

Analysing the feedback is one of the key aspects of learning process. For that, feedback data needs to be processed and analysed. Various techniques are used for analysis. There are various parameters on which the feedback can be analysed. From the analysis, conclusions can be drawn. Systematic analysis of feedback generates proper outcomes which can help to improve the learning system. Number of strategies are used for feedback analysis.

From the literature survey, various gaps in the traditional systems are identified. To overcome these gaps, hybrid mathematical model is proposed to get more realistic feedback value based on various real-time parameters. The model is bi-directional where teacher's opinions are also considered for feedback evaluation. Realistic conclusions are drawn from the analysis. Descriptive feedback is generated for better clarity.

O. Ekbote (✉) · V. Inamdar
College of Engineering, Pune, India
e-mail: ekboteo17.is@coep.ac.in

V. Inamdar
e-mail: vhj.comp@coep.ac.in

This paper incorporates survey of various feedback analysis systems. Section 2 describes literature survey. Section 3 states research gaps. Section 4 discusses implemented system. Section 5 explains results and analysis.

2 Literature Survey

Various universities have their own feedback systems. Students can provide feedback through online portal or offline feedback forms. Few feedback systems contain only objective type of questions. Few systems allow students to rate the teacher based on various parameters like teaching effectiveness, punctuality, communication skills etc. Some of the feedback questionnaires consist of descriptive types of questions where students can write their opinions through comments. Number of students entering feedback varies from 50 to 1000. The data from the feedback system is collected and pre-processed, if required. Pre-processing involves cleaning of data, separation of numeric and descriptive data, conversion of the data into required formats. Pre-processing is an important step for analysis of the data.

Kolchinski et al. [1] has conducted a proof-of-concept experiment to illustrate the importance of the technique to calculate the feasibility of natural-language feedback targeting. They have performed the survey in four stages such as pre-test, exercises, feedback and post-test. In pre-test, all participants have answered six multiple-choice questions where each question was given with three options as answers. During exercise section, participants were asked to enter some descriptive feedback about same questions in the pre-test. In feedback, paragraphs were given to the participants and they had to enter the feedback which was uniformly randomized. The last step was post-test in which the same questions from pre-test were asked in different order. Scores were given to the correct answers. The learning gain was calculated by subtracting the pre-test score from the post-test score. They have implemented and tested four policies: Oracle, multiple-choice targeted policy, and two NLP targeted policies (word vector sums and bag of words). All policies mentioned above were trained and tested on respective four types of responses and one exercise response at a time. NLP policies described in this experiment were also tested out of sample, that is, they were trained on first three responses and tested on the remaining fourth. Unlike the multiple-choice models, the NLP models could generalize to questions which were unseen and interactions by learning how students respond to prompts depending on what they understand and remember.

Luo et al. [2] have used integer linear programming (ILP) formulation and ROUGE scores for feedback analysis. As there was a lexical diversity in the responses given by students, they have suspected that the co-occurrence matrix may not establish a trustful similarity between concepts from sentences. A concept would have conveyed using multiple bigram expressions, but calculated co-occurrence matrix only captured a binary relationship between sentences and bigrams. The existing matrix was highly sparse. Only 2.7% of the figures were non-zero in the generated dataset. So, they have proposed to impute the co-occurrence matrix by completing missing

values. By approximating the dataset co-occurrence matrix using a low-rank matrix, values are fulfilled. The low-rankness urged comparative ideas to be shared crosswise over sentences. They have utilized every one of the sentences in 25 addresses to build the idea sentence co-event grid and performed information ascription. It has enabled them to use the co-event measurements both inside and crosswise over addresses. They have exhibited outline results assessed by ROUGE and human judges. ROUGE is a standard assessment metric that thinks about framework and reference outlines dependent on n-gram covers. Their proposed methodology has outflanked every one of the baselines dependent on three standard ROUGE measurements. They have attempted to condense understudy criticism utilizing a whole number straight programming system with information attribution. Their methodology has enabled sentences to share co-event measurements and eases sparsity issue.

Kassim et al. [3] have used Student Feedback Online (SuFO) ratings as a data for feedback analysis. In the analysis, they have generalized the ratings against various parameters and tried to get the results. Respective graphs are plotted for the mathematical representation of the feedback data. They have checked different scales for plotting. Firstly, they have checked whether the ratings were influenced by an experienced/non-experienced teacher. They have tried to find a relation between student's rating, teacher's experience and teaching quality. Then they have plotted the results of students' academic performance with respect to teaching quality based on SuFO ratings for teaching skills. They have collaborated the graphs and experimented for students' academic performance against teacher's experience. In all these experiments, students' grades are divided into four categories like excellent (A+, A), good (A–, B+, B), fair (B–, C+, C), weak (C–, D+, D, D–, F). From the plotted graphs and various permutations, results are concluded.

Mishra et al. [4] have discussed how rough set theory is useful in analysing large student data, and for generating classification rules from a set of observed samples. Utilizing rough set decrease strategy, all reducts of the information which contains the negligible subset of characteristics that are related with a class name for characterization were determined. They at first viewed as 1000 examples, of criticism which was gathered from different sources cases and seven restrictive qualities, for example, direct methodology, backhanded methodology, given task, input on research centre execution of understudy, consequence of the end semester, irregular determination from expansive example size and accumulation of criticism testing understudy IQ, quick criticism soon after single class end and the qualities characterized as low, moderate and high and choice properties are sure, negative. They renamed the properties and qualities for better understanding ($a_1, a_2, a_3, a_4, a_5, a_6, a_7$) as contingent characteristics, (b_1, b_2, b_3) as estimations of the restrictive traits and (c_1, c_2) as choice qualities separately. The beginning stage of rough set hypothesis was the ambiguity connection, produced by data concerning objects of intrigue. The incoherence connection was proposed to express the way that because of the absence of learning, that was unfit to recognize a few articles utilizing the accessible data. Approximations was additionally other an essential idea in rough sets theory, being related with the importance of the approximations topological tasks. The lower and the upper approximations of a set were inside and conclusion tasks in a topology

created by the incongruity connection. From these approximations, they have registered fundamental segments of basic leadership in the event of input framework to give precision.

Williams et al. [5] have concentrated on the movement of a model, named Klass-Base, went for improving commonly low understudy commitment and adjusting for weaknesses of presently utilized commitment strategies. The model was a cell phone application intended to boost genuine, visit sharing of feedback among understudies and teachers, and give understanding into the territories of a course which need improvement. They have tried their suppositions about which highlights of the model would decidedly affect commitment, first with meetings to refine the methodology, at that point with online overviews to quantify the execution of our model against one at present used technique for empowering understudies to give criticism customary end obviously assessments. In their test, respondents were first made a few statistic inquiries and after that were questioned on their present status concerning understudy commitment, just as how educators use criticism. At that point, members were given one of two situations: The first trotted around end obviously assessments, and the second around the KlassBase model. In the two cases, members were approached to envision themselves as an understudy having simply left a class through the afternoon and are given the offered device to give input. In the wake of watching the instrument and its highlights, members were then questioned on how they felt it would influence their in-class commitment and the class' input circle. At the point when gotten some information about KlassBase, respondents were likewise solicited to rate the convenience from some of the model's highlights. Members were then given the other situation, contingent upon which one was first served.

Dhanalakshmi et al. [6] have utilized Rapid Miner apparatus to mine the understudy criticism and arrange it as positive and negative. So as to comprehend the extremity of conclusion regarding different highlights, the terms module, teacher, exam, resources were picked. These words and their equivalent words were connected to the administrators SelectAttributes and FilterRows in Rapid Miner to comprehend the extremity include shrewd. The administered learning calculations that were utilized are SVM, NB, K-NN and NN. The validation administrator from Rapid Miner which permits to at the same time train and test the classifiers was utilized. In particular, the facility of cross validation was exploited for the input dataset by setting the value to 10. This implied the informational index was isolated into 10 gatherings of which initial 9 bunches were utilized as preparing and one gathering was utilized for testing. In the second run, an alternate mix of 9 sets turned into the preparation information and another set turned into the testing information. The procedure was proceeded until every one of the changes will be done. To look at the execution of the four calculations utilized, accuracy, precision and recall esteems were determined for every one of the classifier calculations by utilizing the Performance Operator of Rapid Miner.

Som et al. [7] have suggested that, to survey the consequences of feedback in a college, utility hypothesis was taken as a key device to support evaluators and the board in assessing the general utility. They have utilized Cronbach's alpha, which is the most generally recognized proportion of inward dependability. Considering

the total anticipated utility esteem, a general proportion of the utility for all of the properties taken with everything taken into account. The general utility measure has helped in educational foundation and individuals in a couple of courses by standing out it from the most exceedingly terrible and the best-case circumstances of the end.

Balahadia et al. [8] proposed the framework which utilizes a lot of language assets (Dataset Polarity that they have made) to distinguish the suppositions communicated in a volume of messages composed by understudies. The Dataset Polarity has comprised of a rundown of lexicons of words physically accumulated for their undertaking which incorporated the Tagalog and English words. At that point the feelings separated were examined to discover the extremity utilizing Naïve Bayes Algorithm. Each notion word in the database has been given an esteem. At the point when an estimation word was distinguished in a sentence, the esteem spared in the database was utilized for assessing the total supposition esteem. At the point when a sentence was examined, for every estimation word found in the sentence, its assessment esteem was brought from the database. At that point the worked together assessment estimation of that sentence was evaluated. On the off chance that there was refutation in a sentence the estimation of conclusion score was diminished/expanded by a specific sum.

3 Research Gaps

In the literature survey, various methods and techniques are discussed to analyse the students' feedback. From the analysis, several things can be concluded. Targeting feedback outputs in education domain to natural-language inputs [1] is a more precise and context independent function than multiple-choice targeting. Students' feedback rating values should not be used as the only parameter to comment about teacher's teaching skills [3]. In comparison of various machine learning algorithms, Naïve Bayes algorithm performed best in case of accuracy and K-Nearest Neighbour algorithm performed the best in case of precision [6]. Sentiment analysis plays an important role in feedback analysis to get more accuracy [8].

It has been observed from the papers, there is no concrete system for student's feedback which can justify student's feedback and maintains trust in the system. There is no correlation between attendance of the student and feedback entered by same student. There is no impact analysis of student's academic performance and his/her feedback. Teacher's feedback for students is not considered anywhere as a part of two-way feedback system. No system had generated the conclusive, summarized descriptive feedback for the teacher which can help for better performance.

4 Implemented Solution

4.1 Objective of the Solution

The objectives of the proposed solution are to generate a realistic feedback value considering few real-time parameters like student's attendance, student's academic performance, and student's current progress and to make a bidirectional feedback system where teacher's opinion about a student is considered.

4.2 Implemented System

The system is implemented which contains few real parameters in order to get more realistic feedback. In this system, student's academic performance, student's attendance, student's current progress is taken in account. In addition to it, teacher's feedback about the student is also considered. These parameters are expressed in terms of weights and then the weights are taken to find the feedback value using linear regression. Hence, the feedback value will be more realistic as compared to traditional system.

Student's feedback is collected for eight parameters (p_1, p_2, \dots, p_8) where the parameters are teacher's clarity of subject, teaching skills, communication and interaction skills, punctuality, availability beyond normal classes and co-operation to solve individual problems, doubts solving capacity, being updated about contents of subject, and overall effectiveness, respectively. Students are supposed to rate the teachers for each course in the semester on the scale of 1 to 5 (1-Low/5-High).

Weights (w_1, w_2, w_3, w_4) are considered for student's attendance, academic performance (CGPA is considered here), score of intermediate tests of that course and teacher's feedback about students, respectively. Teacher is also supposed to rate each student registered for that course on a scale of 1 to 5 (1-Low/5-High). Hence, the system has become bi-directional feedback system. Weights are calculated by dividing the respective data in different ranges. Weight values are in between 0 to 1.

Calculated weights are used with student's feedback ratings to get realistic feedback using multiple linear regression.

4.3 Multiple Linear Regression

Multiple linear regression consists of dependent and independent variables. In this system, p_1, p_2, \dots, p_8 (parameter rating values) are treated as independent variables and R (realistic feedback value) is considered as dependent variable. Here, 8 features are taken in account hence $N = 8$.

$$R = \{(w_2 * w_3 * p_1) + (w_1 * w_3 * p_2) + (w_1 * p_3) + (w_1 * w_4 * p_4) \\ + (p_5) + (w_1 * w_4 * p_6) + (w_2 * w_4 * p_7) + (w_4 * p_8)\}/N \quad (1)$$

Equation (1) represents multiple linear regression to get the realistic value of the feedback.

4.4 Implemented Algorithm

Input: Parameter values p_1 to p_8 of N students, Attendance of N students, CGPA of N students, intermediate exam scores of N scores, Teacher's ratings of N students

Output: Realistic Feedback Value (R)

Algorithm:

1. Calculate w_1 for N students by dividing attendance into spectrum of 0 to 1.
2. Calculate w_2 for N students by dividing CGPA into spectrum of 0 to 1.
3. Calculate w_3 for N students by dividing intermediate exam scores into spectrum of 0 to 1.
4. Calculate w_4 for N students by dividing teacher's feedback ratings into spectrum of 0 to 1.
5. Use Eq. (1) to get realistic feedback value (R).
6. Repeat steps 1 to 5 for all courses to get respective values of R for each course.

4.5 Descriptive Feedback

Descriptive feedback is generated in this model to get more clarity of feedback analysis. The proposed mathematical model generates some mathematical conclusions which lead to descriptive feedbacks. In addition to this, comments entered by students while registering feedback are summarized using sentiment analysis. Sentiment analysis provides a strong way to know student's opinion about the teacher. It is done using natural language processing (NLP). As the text data is unstructured type of data, it is pre-processed by noise removal, lexicon normalization, object standardization. This combined hybrid model concludes descriptive feedback which will be helpful for better understanding of feedback analysis.

Table 1 Feedback values by traditional system and proposed system

Course name	Feedback value by traditional system	Realistic feedback value generated from proposed system (R)
Subject1	3.8803	2.6732
Subject2	4.5632	4.1904
Subject3	4.1258	3.3287
Subject4	3.5571	2.9874
Subject5	3.9714	3.0816

5 Experimental Results and Analysis

5.1 Results of Feedback Values

In traditional system, feedback value is calculated just by average of all ratings given by all students. In the proposed system above, realistic feedback value is calculated by considering weighted real-time parameters. So, realistic feedback value will always be less than or equal to feedback by traditional systems. Table 1 compares respective values.

5.2 Results and Analysis of Multiple Linear Regression

Figure 1 represents a graph of multiple linear regression where C_{15} represents realistic feedback value for Subject 1. The same can be represented for other courses also. Experimental results are tested using tool NCSS.

From Fig. 1, it is observed that, most of the data fits to the regression line. This means, the weighted parameters lead to the conclusive realistic feedback. There is a co-relation between realistic feedback (output of multiple linear regression) and weighted parameters (inputs of multiple linear regression). The fitting of the line proves that, the realistic feedback value is dependent on the factors like student's attendance, CGPA, intermediate scores and teacher's feedback also. The points which are on or closer to the regression line show that most of the students have entered feedback correctly, as all other parameters support their feedback ratings. Few points are deviated from regression line. This means, few students have not registered feedback properly. There is a mismatch between their feedback ratings and values of other factors.

The estimated equation of results from multiple linear regression is

$$\begin{aligned}
 C_{15} = & -1.54855182884243 + 0.225135078808338 * C_2 \\
 & + 0.536365013497631 * C_3 + 0.242842223242684 * C_4 \\
 & - 0.0976611619670068 * C_5 + 0.113908603944585 * C_6
 \end{aligned}$$

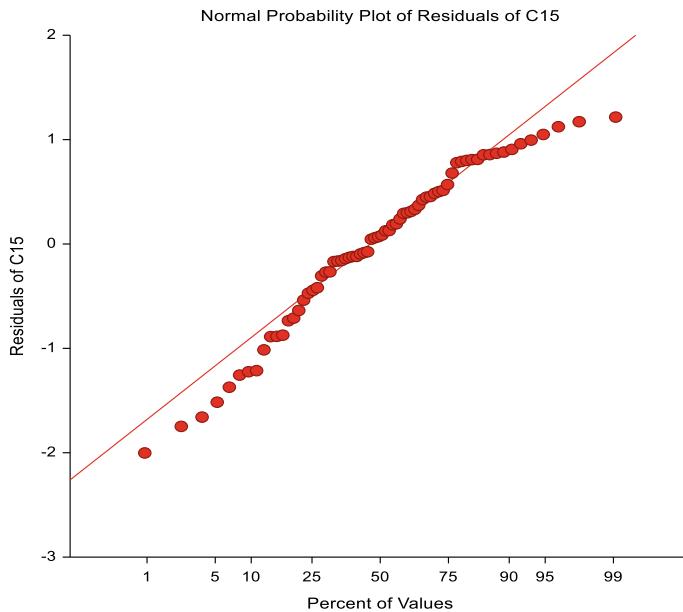


Fig. 1 Multiple linear regression plot

$$\begin{aligned}
 & + 0.000747938210716322 * C_7 \\
 & - 0.0460338710527741 * C_8 - 0.0763965254867361 * C_9
 \end{aligned} \quad (2)$$

Here C_{15} is realistic feedback value (R). Columns C_2 to C_9 of the dataset represent parameters p_1 to p_8 , respectively.

From question (2) and Table 2, it has been observed that, C_3 is more directly proportional to R and C_5 is more inversely proportional to R . That means, for the course ‘Subject1’, teacher is having good teaching skills (C_3), but the teacher can

Table 2 Standard coefficients and hypothesis rejection results

Regression independent variable	Standard coefficient	Reject H_0 at 5%?
C_2	0.225135078808338	No
C_3	0.536365013497631	No
C_4	0.242842223242684	No
C_5	-0.0976611619670068	No
C_6	0.113908603944585	No
C_7	0.000747938210716322	Yes
C_8	-0.0460338710527741	No
C_9	-0.0763965254867361	No

improve on punctuality (C_5). Positive/negative coefficient value shows importance of respective parameters. This analysis can be used continuously for few years to judge whether the teacher has made any improvement/s from the feedback or getting same feedback for same parameter. Continuous evaluation can add more values in learning system.

These conclusions provide descriptive analysis of feedback. Coefficients of the parameters express positive/negative intent for that parameter. The positive/negative intents of all parameters when combined provide descriptive feedback. From Table 2, descriptive feedback is generated like—‘Teacher is good in teaching skills. Teacher is good in clarity of subject. Teacher should improve on punctuality. Teacher should improve on being updated about content of subject’. Here 4 values of coefficients of parameters are considered (C_2, C_3, C_5, C_8 from Table 2), and from positive/negative intent, sentences are made to get conclusive descriptive feedback.

From Table 2, it has been observed that, 7 out of 8 parameters are having hypothesis rejections result as ‘No’. So almost all parameters are accepted from null hypothesis in multiple linear regression. This concludes for normal distribution of data where all possibilities of data are considered. Hypothesis acceptance proves the validity of dataset.

6 Conclusion and Future Scope

The proposed realistic mathematical model generates more realistic feedback value as compared to traditional feedback systems. The model also ensures bi-directional feedback system where teacher’s feedback is also considered. This approach helps in getting more concrete, more informative and more specific feedbacks which can help to improve overall learning process.

This model can further be expanded in terms of handling anonymity issues as student’s marks and attendance are taken in account here. This approach can be helpful to have better clarity if teacher’s feedback is taken in a descriptive manner.

References

1. Kolchinski YA, Ruan S, Schwartz D, Brunskill E (2018) Adaptive natural-language targeting for student feedback. In: L@ S, pp 26–1
2. Luo W, Liu F, Liu Z, Litman D (2018) Automatic summarization of student course feedback. arXiv preprint [arXiv:1805.10395](https://arxiv.org/abs/1805.10395)
3. Kassim RA, Buniyamin N (2015) Evaluating teaching quality using data from student online feedback system. In: 2015 IEEE 7th international conference on engineering education (ICEED). IEEE, New York, pp 64–68
4. Mishra S, Mohanty SP, Pradhan SK, Hota R (2015) Generation of rules for student feedback system by the use of rough set theory. Int J Comput Appl 131(18):54–57
5. Williams N, Mondschein J, Farmer M, Twyman N (2018) Mobile course feedback system for improving student engagement

6. Dhanalakshmi V, Bino D, Saravanan AM (2016) Opinion mining from student feedback data using supervised learning algorithms. In: 2016 3rd MEC international conference on big data and smart city (ICBDSC). IEEE, New York, pp 1–5
7. Som S, Majumdar R, Ghosh M, Malkani C (2017) Statistical analysis of student feedback system using Cronbach's alpha and utility measurement process. In: 2017 international conference on Infocom technologies and unmanned systems (trends and future directions) (ICTUS). IEEE, New York, pp 391–395
8. Balahadia FF, Fernando MCG, Juanatas IC (2016) Teacher's performance evaluation tool using opinion mining with sentiment analysis. In: 2016 IEEE Region 10 symposium (TENSYMP). IEEE, New York, pp 95–98

Chapter 15

Fake News Classification on Twitter Using Flume, N-Gram Analysis, and Decision Tree Machine Learning Technique



Devyani Keskar, Sushila Palwe and Ayushi Gupta

1 Introduction

Social media sites and online social networks, for example, Facebook, Twitter, etc., are the main mediums for spreading rumors and are held responsible for being unable to restrain fake news being spread at very fast rate. Various reasons for creation and dissemination of rumors and fake news are making gains in politics and business by harming the reputation of businesses or individual and for seeking attention. Fake news detection is a very complex job as people believe misleading information very easily and cannot control the urge to spread the fake content for the purpose of either fun or with malicious intentions. Widespread use of the Internet, social media, and the advancement in technology makes it easy to create and spread fake news. With the aim of detecting fake news, we are studying N -gram analysis using DT machine learning technique which will be applied on livestream data collected from Twitter.

2 Related Work

Horne [1] in his experiment proved how easily news can be classified into fake and real news articles. It was observed from their experiments that the fake news

D. Keskar (✉) · A. Gupta

Department of Computer Engineering, MITCOE, Pune, India

e-mail: devyani.keskar@gmail.com

A. Gupta

e-mail: ayushimg9@gmail.com

S. Palwe

School of CET, MITWPU, Pune, India

e-mail: sushila.palwe@mitwpu.edu.in

articles have titles with less stop words as nouns, while the number of verbs used is more. The feature that extracted was classified as follows: Readability of text and its complexity were described by complexity features. Psychological features measured the concerns which varied from person to person in their writings which include use of emotional words, casual words, and formal scientific words. Syntax editing features like use of verbs and nouns in the text which describes that the style of writers is the third category. Using them as base, SVM classification model was built. Dataset used in this experiment was BuzzFeed. It contained real news articles. For testing the model, Burfoot and Baldwin's dataset which is satirical was used. As per their results of their experiments when real news was compared with satirical articles (humorous article), the accuracy obtained was 91%. When fake news and real news were predicted, the accuracy lowered to about 71%. Okoro [2] in his paper has explained how hybrid approach which is the combination of models based on human-based approach and machine-based approach can be used to detect fake news. In it, he has proposed the use of human-based model and machine-based model to assign values to the ten factors stated by news literacy education tool for the detection of fake news. According to him, the sum of all factors should be less than 100 for news to be truthful news. Hadeer Ahmed [3] in his paper has proposed using N -gram analysis along with six techniques that use machine learning concepts for detection of fake news. He has compared the six machine learning techniques on the basis of accuracy of results produced for unigrams, bigrams, and trigrams. As per his study, if LVSM, i.e., linear support vector machine, is used for classification then maximum accuracy is obtained. Wang introduced LIAR [4]. It is a new dataset which can be used for detecting fake news automatically. Although big enough, it has a few articles. Statements (12,800) from politicalFact.com which are labeled manually are a part of it. For identifying articles that have a base of humor and satire, Rubin [5] put forth a method for which he took into account 360 news articles with satire which consisted of domains like civics, business, science, and soft news. Punctuation, humor, grammar, negative effect, and absurdity are the features that are used by him for SVM machine classification technique. By combining three features which are grammar, absurdity, and punctuation, maximum precision that could be obtained was 90%. Kataria [6] in his paper has explained the use of flume for extracting online data and storing it in HDFS, organizing the data using Hadoop file system or Google file system, and then finally analyzing the data using map reducers in Pig, Hive, and Jaql [7].

3 Data Extraction Using FLUME

Apache flume is a system which is used to gather and transfer large amount of log data from various sources to a centralized store like HDFS. It also provides end-to-end reliability for the system and is also a distributed system (Fig. 1).

Here, social networking sites like Twitter and Facebook are the data generators. The data generated by the data generators is then collected by the agents of the flume

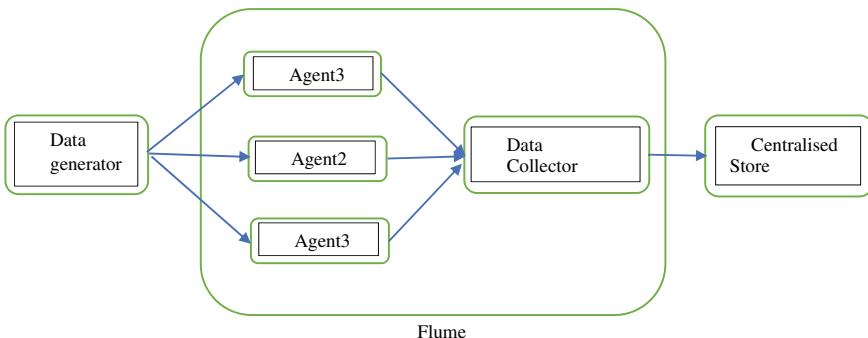


Fig. 1 Data flow in flume

running on these sites. This data is then collected by collector. The collector of data is also a flume agent, and it stores the collected data on a centralized store like HDFS. Some important terms related to flume are:

- **Flume Event:** The data that travel through flume or data that transported through flume is called the data event.
- **Flume Agent:** It is a non-contingent background process which receives data from agents and forwards it to the destination or the sink. The source, channel, and the sink are the main components of the flume agent.
- **Flume Source:** It receives event from data generators and hands it over to the channel.
E.g., Twitter 1% source
- **Channel:** A channel acts like a buffer between source and sink. It holds the data sent by source until buffer removes it for transporting it ahead.
- **Sink:** It removes data from the channel and transports it to next agent. Sinks also transfer data to the final destination. Such sinks are known as terminal sink. Example of a terminal sink is HDFS sink.

Various source types are supported by flume which includes Netcat source, Spooling Directory source, Syslog source, HTTP source, Avro source, Exec source, Twitter source, etc. Various types of sinks supported by flume include Null sink, Avro sink, HDFS sink, IRC sink, Logger sink, Thrift sink, Morphline Solr sink, File Roll sink, HBase sinks, Elastic Search sink, etc. FLUME has the following features:

- Flume can be scaled horizontally.
- It can collect data from large number of servers and store it in HDFS.
- Importing and analyzing data generated from online sites such as Twitter, Facebook, and few business Web sites become feasible.

4 N-Gram Analysis with Decision Tree Machine Learning Technique

4.1 Introduction

N-gram feature-based approach for detection of fake news performs analysis of text based on features and uses one of the techniques of machine learning for the purpose of classification, i.e., decision tree machine learning technique in this research. Classifier based on *N*-gram is generated to classify the document under study as fake news or real news. The accuracy of the algorithm changes as per the size of *N*-gram.

4.2 N-Grams

N-grams are the *n*-character slice or the *n*-word slice of the text that is taken into consideration. For a sentence of length *k* along with blanks, *k* + 1 bigram, *k* + 1 trigram, and so on are present.

4.3 Data Preprocessing

Various functions involved in data preprocessing are removal of stop words, conversion of statements into tokens, conversion of document into lowercase, and removal of punctuation. It is performed for the ease of processing in order to remove unnecessary details of the text. First, the punctuations and special characters are removed, and then, conversion of all the letters to lower case is done.

Stop-Word Removal

In stop-word removal step, all the stop words which are unimportant and unnecessary words in text that cannot be used for analysis are removed. Some examples of stop words are pronouns, articles, conjunctions, prepositions, etc.

Stemming

In stemming, the words are transformed into their original form (e.g., ate to eat). Various algorithms can be used for the purpose of stemming but the most commonly used algorithm is porter stemmer.

Feature Extraction

It is a technique used for the selection of features. For this purpose, term frequency (TF) and inverted document frequency (TF-IDF) are used. Term frequency keeps the track of the count of words appearing in document for establishing similarity between documents. A vector is used for document representation whose length is

equal to the number of N -grams. This vector maintains the count of N -grams which is created after tokenization appearing in document. TF-IDF states the importance of term in the document. TF-IDF basically defines the inverse probability that a word might appear in text. TF-IDF assigns some weight to words in document as per their importance in document.

5 Methodology

Data is extracted from Twitter using flume. For this purpose, we have to configure the Twitter app. Once we create an application, Twitter provides consumer key, consumer secret, access token, and access token secret. They are four data elements that are used to extract tweets from Twitter. The data extracted using flumes gets stored in HDFS. However, the data which is obtained has some parts which are not required for classification purpose. Hence, we need to remove the irrelevant details and preserve the main sections of tweets.

Once we get all the necessary data, further processing is performed on it. Initially, data preprocessing is performed on the text in order to reduce its size and for the ease of processing. How the size of N -gram affects accuracy of results is studied. Feature extractions are performed on unigrams which is based on TF-IDF. TF-IDF is basically used to assign weight to N -grams in the order of their importance. TF-IDF is calculated using the following formula:

$$W_{mn} = tf_{mn} * (\log N / df_{mn}) \quad (1)$$

Here, W_{mn} stands for weight of word m in document number n . N is the total documents considered, whereas df_{mn} stands for the count of documents that contains the word. Tf_{mn} is the count of word m in document n .

Once weights are assigned to words in the documents, a matrix of these weights is prepared which gives details of weight associated with a word in a particular document. After feature extraction, the decision tree machine learning technique is applied for classifying the tweets as tweets with fake content and tweets with truthful contents. The accuracy of decision tree method for the results of unigram is then calculated.

We calculate the accuracy based on the following metrics:

1. True Positive (TP): News articles predicted to be fake are fake news
2. True Negative (TN): News articles predicted to be true are true news
3. False Negative (FN): News articles predicted to be true are fake news
4. False Positive (FP): News articles predicted to be fake are true news.

Using these metrics, the following formula is used to calculate the accuracy:

$$\text{Accuracy} = \frac{|TN| + |TP|}{|FP| + |TP| + |FN| + |TN|} \quad (2)$$

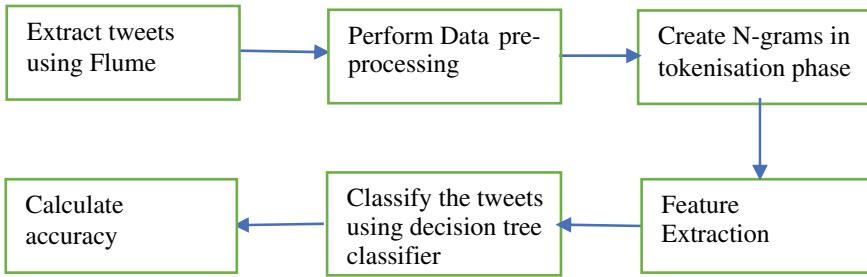


Fig. 2 Flow diagram for classification of tweets

The dataset which is generated by extracting and filtering the Twitter data is used for the experiments (Fig. 2).

6 Result Discussion

Out of the ten records considered, five records have true class and five records have fake class. When there is binary classification, i.e., if only two classes are considered, then the entropy is high, viz. 1. Entropy can be calculated by the formula:

$$H(x) = -\sum p(x) \log p(x) \quad (3)$$

Initially, entropy of the target is calculated after calculating the entropy of target; entropy of each attribute is calculated. We have considered eight attributes. After calculating entropy of each attribute, it is subtracted from entropy of target to obtain the information gain. Attribute with highest information gain is used as root node to construct decision tree (Figs. 3 and 4).

After analyzing the test data, the accuracy obtained is $[TP + TN/Total] = [7/10] = 0.7$, i.e., 70%.

True positive rate is $[TP/\text{Actual Fake Class}] = 0.6$, i.e., 60%.

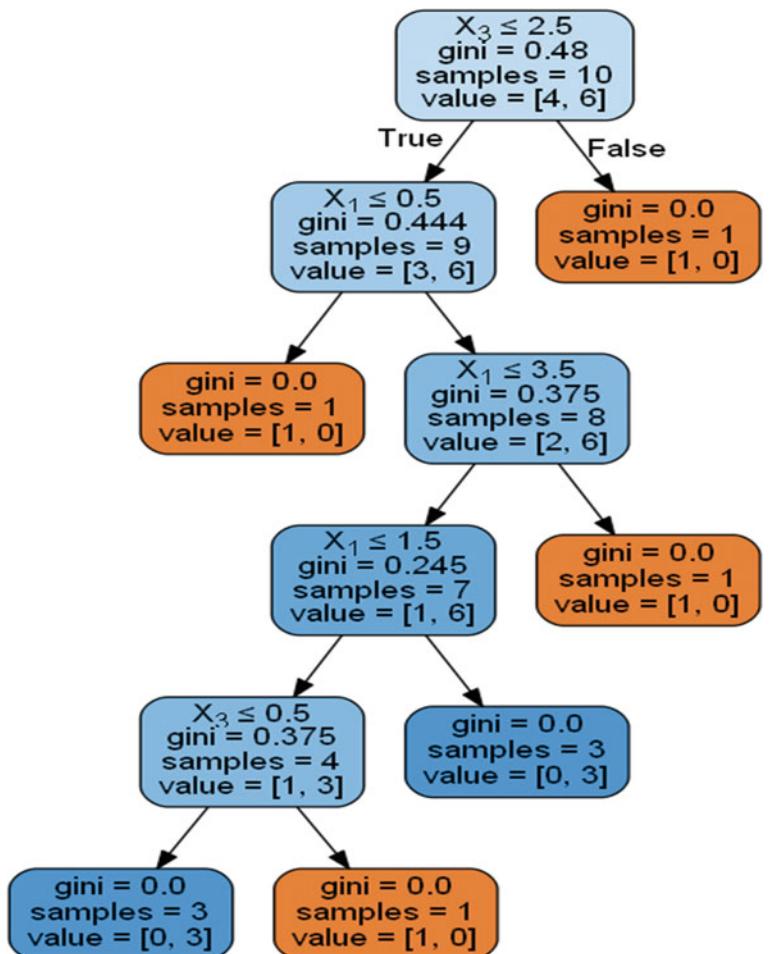
False positive rate is $[TN/\text{Actual Fake class}] = 0.8$, i.e., 80%.

Precision obtained is $[TP/TP + FP] = 0.75$, i.e., 75%.

Recall obtained is $[TP/TP + FN] = 0.6$, i.e., 60%.

Hence, the F-measure is $[2 * \text{Recall} * \text{Precision}/(\text{Recall} + \text{Precision})] = 0.66$, i.e., 66%.

High precision low recall in our analysis indicates that there is a considerable amount of false negatives (FN) but the ones we predict as true are indeed true.

**Fig. 3** Decision tree for training dataset**Fig. 4** Confusion matrix of training dataset

		PREDICTED CLASS	
		TRUE	FAKE
ACTUAL CLASS	TRUE	3	4
	FAKE	1	2

7 Future Scope

If decision tree is used along the N -gram analysis, then accuracy obtained is 70%. Other machine learning techniques can be used along with the N -gram analysis to check if the accuracy increases. Since currently very a smaller number of datasets are available that can be used for classification of fake news, a new dataset can be created which will contain Twitter data extracted using flumes which is already classified. So that when needed, one can cross-check the results of his research with an existing result. Additionally, feedback from user can also be considered to classify news as fake or real which will help to reduce the inefficiency of machine in prediction and provide an edge over methods which use only human-based model or only machine-based models for classification. It is also possible to develop a Web site where user can input a news, he is suspicious about. After cross-checking the database of classified news, the Web site will display the result to the user. The database will have classification of news on Twitter which is extracted using flume and classified using N -gram analysis and one of the machine learning techniques that gives highest accuracy. The user can then provide feedback over the result displayed by the Web site which will be considered to make the required changes in database.

8 Conclusion

The rate at which fake news is spreading and its detection is now becoming a serious problem affecting each and every sector of life. The use of flume system to extract data from online sites is studied to overcome the problem of insufficient literature or database for the fake news detection. Using flume, the data which is obtained can be directly stored in HDFS which is also the solution to storage problem as the data obtained is heavy and large in number. Once the data in required format is extracted and stored, it can be processed further to classify it as fake news and truthful news. If N -gram analysis used along with decision tree machine learning technique for classifying the document as fake or real, then the accuracy obtained is 70%. For the same purpose unigram, TF-IDF features are taken into consideration. Using this approach, classification of news articles into various types based on their heading is also possible.

Acknowledgements I am thankful to Malaya Sir and Ritesh Shinde sir for their time and insightful tips in the work.

References

1. Horne BD, Adali S (2017) This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: The 2nd international workshop on news and public opinion at ICWSM
2. Okoro EM, Abara BA, Umagba AO, Ajonye AA, Isa ZS (2018) A hybrid approach to fake news detection on social media. Nigerian J Technol (NIJOTECH) 37(2):454–462
3. Ahmed H, Traore I, Saad S (2017) Detection of online fake news using N-gram analysis and machine learning techniques. In: Conference paper (October 2017)
4. Wang WY (2017) Liar, Liar Pants on fire: a new Benchmark dataset for fake news detection. arXiv preprint [arXiv:1705.00648](https://arxiv.org/abs/1705.00648)
5. Rubin VL et al (2016) Fake news or truth? Using satirical cues to detect potentially misleading news. In: Proceedings of NAACL-HLT
6. Kataria M, Mittal P (2014) Big data and Hadoop with Components like Flume, Pig, Hive and Jaql. Int J Comput Sci Mob Comput. Available Online at www.ijcsmc.com
7. Deshmukh R, Kirange MD (2013) Classifying news headlines for providing user centered e-newspaper using SVM. Int J Emerg Trends Technol Comput Sci (IJETTCS) 2(3)

Chapter 16

Swarm Intelligence-Based Systems: A Review



Vedant Bahel, Atharva Peshkar and Sugandha Singh

1 Introduction

Swarm Intelligence: Over the last few years there have been numerous advancements in the field of robotics, automation, artificial intelligence and their uses. Comparatively, swarm intelligence has been in a relatively low focus. Swarm intelligence has its roots in the way in which some social insects interact with the nature in a unique and smart way despite having subordinate capabilities. Figure 1 shows a swarm of auklets that can be depicted into system of swarm robots.

These swarms of insects, fishes or mammals, allocate different tasks amongst themselves in order to construct, collect food, safeguard territory and countless similar tasks, thereby depicting great flexibility and vitality. In these insects or animals, there exists some obscure mechanism which divides the one complex task into simpler tasks for individual to perform and then finally integrate everything demonstrating collective behaviour.

Similarly, this concept can be reproduced to create group of robots adapting the identical behavioural strategies for search and rescue as well as in defence operations.

Although certain tasks might be herculean for an individual to perform single headedly, but swarm of animals manage to do it without any hassle.

This vitally requires a better communication and information interpretation amongst all the elements of the swarm.

V. Bahel (✉) · A. Peshkar
Department of Information Technology, G.H. Raisoni College of Engineering,
Nagpur 440016, India
e-mail: bahel_vedant.ghrceit@raisoni.net

A. Peshkar
e-mail: peshkar_atharva.ghrceit@raisoni.net

S. Singh
CSE, G.H. Raisoni College of Engineering, Nagpur 440016, India
e-mail: sugandhasinghhooda@gmail.com



Fig. 1 Swarm of Auklets

This paper concentrates on concepts, design and features that required to build an operative swarm robot.

The synergy between the robots and the interaction of robots with the environment forms the foundations for functioning of swarms.

The intricacy is reduced by using the principle that establishing and following a fixed set of rules at individual level can lead to exhibition of complex behaviours by the swarm [1] (see Fig. 2).

2 Comparison of Swarm Robots with Traditional Robots

Swarm robots are multiple, small sized and unfussy robots apart from the complex, giant and traditional robots. Swarm robots constitute of simple software and hardware as well. Swarm is the channel of distributed robots where each robot has a specific task. They work in a locally communicated environment. And have local sensing. Swarm robots may or may not be self-learning, whereas traditional robots generally have self-learning algorithm embedded in them.

Moreover, traditional robots generally have controlled and centralised system, whereas swarm robots have decentralised and autonomous system which helps in increasing the response time making them more efficient. Thus, the major difference between both lies in population, control, homogeneity, flexibility and functional extension [1].



Fig. 2 Swarm robots following one another in a defined path

There is also the difference of application of both the system: swarms are generally proposed to be used in, defence and health care, whereas traditional robots have diverse uses right from cooking to industrial production.

3 Features of Swarm Robots

The following points describe the main features or characteristic of swarm robots and intelligence.

- (i) *Decentralised*: The robots must be decentralised and self-organising. That is, the functions that they perform are not controlled by external agent. The robots must have internal communication and sensing abilities.
- (ii) *Size*: The number of robots in a particular artificial swarm must be huge in number. They must be present in a flock. But each robot is expected to have size corresponding too few millimetres.
- (iii) *Homogeneous*: The robots in a swarm must be mainly homogeneous. There shall not be many different types of robots. However, some researches demonstrate few swarms of robot having different type of robots designed to carry out different interdependent functions.

- (iv) The robots must mainly focus on the task allocated to them but be operational to smart decision-making for task shift with use of automation [2].
- (v) Robots in a swarm demonstrate limited area communication and sensing only. It ensures scalability of the robots [2].
- (vi) Restricted sensing abilities and communication also ensure reduced price of robot in the swarm environment.
- (vii) *Flexibility:* As the swarm can adapt multiple functionality with same hardware and by making only few changes in the software, it ensures the flexibility of the same. By using the advanced technology of machine learning the robots can improve themselves from their past moves [1].
- (viii) *Flying:* Previously, flying swarm generally had externally tracking and operation system. But this does not make the swarm feasible. Instead, it can have global positioning system (GPS) to relocate and create its own map in the environment. Laser scanners can be used to detect the robot's pose and motion in a plane.
In [3], they have used robotic sensor network paradigm to get navigated without global information. Robots are of two types: beacons or explorers. Beacons were made to have the aerial view [4], forming nodes in the network. Explorers were made to map the environment and perform necessary operations.
- (ix) *Sensors:* Sensors sense information about the surrounding areas. They may able to store the information locally or forward it at the control areas for further analysis as per the requirement. Robots can also use communication to share the information with other robots in the system [5].

4 Architecture of Swarm Robots

Swarm as a whole for performing a particular task rely on the number of member robots rather than the complexity of task the can perform. Thus, a well-defined architecture is necessary for the proper and efficient swarm functioning. Authors have described the basic architecture of swarm as follows: Swarm robotics model is an essential component of cooperative algorithm which controls the behaviours and interactions of all individuals in the swarm. According to the model, the robots must possess some basic functions related to communication, motioned.

On the basis of functionality utilised by the module to perform certain tasks, the model can be divided into three parts namely *data exchange*, *basic behaviour* and *advanced behaviour*.

The *data exchange* module sets up the basic framework on which the entire swarm relies. The individuals in the swarm propagate the information throughout the swarm autonomously producing cooperation amongst the individuals. General model of swarm robotics is shown in Fig. 3. Occasionally, global and centralised commands are set up, and it is ensured that the swarm is still able to complete the task even if the global communication is blocked.

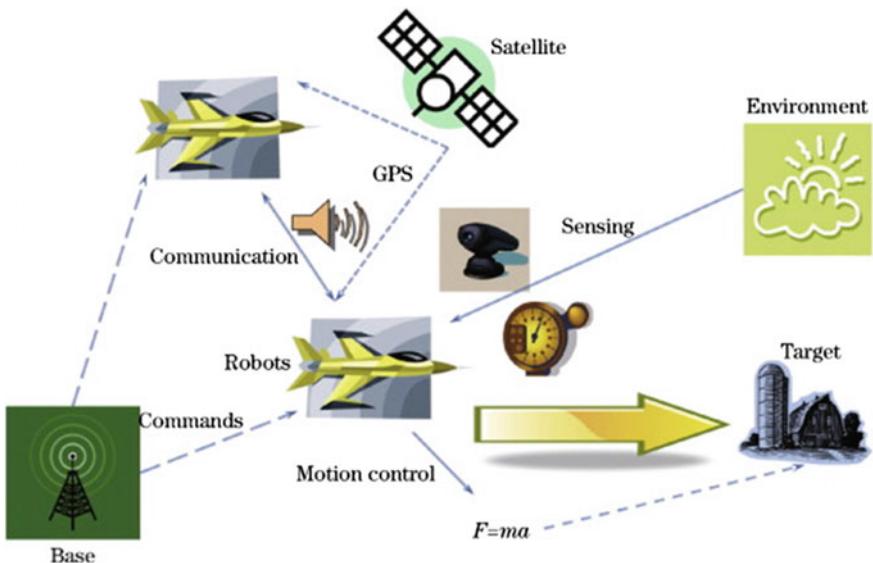


Fig. 3 General model of swarm robotics [1]

The architecture of the swarm is a structure for robotic activities and interactions and determines the topology for information exchange amongst members of the swarm. Thus, it is essential to choose the appropriate architecture of the swarm as the architecture directly affects the performance of swarm.

Locating Global coordinating systems not accessible every time. Therefore, every member of the swarm has to be equipped with a local coordinating system to incorporate the ability to distinguish, identify and locate the nearby robots. Thus, a method for rapidly locating other robots using on-board sensors is very important in swarm robotics [6]. The sensors can sense different waves, including ultrasonic, visible light, infrared ray or sound. However, the relative positioning of swarm robots is more realistic since the abilities of the robots are limited and no global controls exist.

Physical connections: Physical connections are used in the situations that single robot can overcome, such as overpassing large gaps or cooperative transportation. In these tasks, the robots should communicate and dock before they go on to execute their tasks.

Self-organisation and self-assembly: Self-organisation is an ability crucial for building a global module through only local interactions of the basic units. The basic units are not governed by a centralised control or have an external commander, rather the swarm level structure emerges from the synergy between the solitary member robots of the swarm, through already established structures.

Whereas, one method for localisation which also demonstrates the self-organisation and assembly used in the case of swarm of flying robots for indoor navigation with the use of robotic sensor network paradigm [5], where each of the

robot operated in either of the two states: beacon or explorer. The beacon robots formed the nodes in the network and are passively attached to the ceiling maintaining an elevated view and sensing the local environment and communicating it to the explorer robots in the swarm. Thus, requiring the explorers to maintain a beacon to beacon flight and flying under the beacon robots. The area where a no beacon robot is present, an explorer robot switched its state to that of a beacon whilst the previous beacon which was no longer required got converted to an explorer, constantly expanding the network and making it a dynamic system.

5 Application of Swarm Robotics

(a) *Search and rescue*

Swarm robotics has an amazing application in search and rescue.

In [5], they did an experiment in the domain of search and rescue by the use of heterogeneous groups of robots. The experiment aimed of locating immobile humans in a building caught with fire. They used temperature, visibility and toxicity sensing in the system.

(b) *Defence*

There is a lot of speculation about advanced drone and robots bought in work by military and defence purposes.

We also propose a kind of high-speed swarm of robots which can penetrate deep inside the trespasser's body at the military borders so as to destroy the blood vessels eventually leading to death.

(c) *Space Science*

The National Aeronautics and Space Administration (NASA) is also using the advanced system of swarm robots. NASA formerly used swarm robotics to fuel space mining missions. These robots were inspired by the behaviour of ants. NASA started with four robots, but slowly advanced its application [7].

(d) *Medicine*

Robotic technology is improving non-invasive medical processes using clarity, firmness and adroitness. The early experiences of using robots in the medical sciences came from the development of technologies to enhance endoscopic procedures of the intestinal tract [8].

Micro-robots interaction based on the stochastic diffusion search (**SDS**) algorithm is a great tool to identify and detect certain pathological parameters of a human body [9, 10]. Machine learning and artificial intelligence experts predict to have advanced micro-robots that goes deep inside the body and kills all the harmful cell as it enters. Eventually, increasing the life expectancy with an appreciable aggregate. The special feature of micro-robots is that they can penetrate through the blood vessels and carry out many important operations that are not possible currently.

If a group of nanoparticles can be programmed to create a swarm, they can potentially travel to target cancer cells, destroy affected tissue very effectively. Whereas drugs simply diffuse into tissues, swarm nanoparticles would have more intelligence and could more effectively target diseased tissue [11].

6 Conclusion

Swarm is cost effective, miniaturised enough to work in tough inconspicuous spaces (search and rescue operations), adaptive to operate in changing terrains/workplaces. The buzzing VLSI might be furthermore helpful in reducing the size of swarms. Whatsoever, it has numerous individual units, proficiency in connectivity and coordination amongst them elevates their rank on a comparative analysis with standard robots. The most captivating feature of swarms is the ability to relocate themselves with the global positioning system and create its own map environment unlike the standard robots which have external tracking system. Another notable feature is the ability to perform an operation even after partial equipment failure.

Introduction of automated systems in bio-medical instrumentation will increase the surgery precision. As far as the conditions of warfare and security are concerned swarms will reduce the loss of life in such situations to a great extent. Looking at the bigger picture artificial intelligence and machine learning.

Gesture robotics, embedded systems are the quintessential of technology theories on which the twenty-first century would be based upon.

Through this paper, authors tried to introduce basic swarm robotics-based systems and their difference from the traditional robots. Features of swarm intelligence were reflected that are capable of successfully deploying advanced application of swarm intelligence.

In conclusion, using swarm robotics, we can solve complicated time and space consuming problems. This work presents a strategy to build an autonomous, decentralised and self-organising system of swarm robots. Authors have also discussed the architecture of the robots along with their sensor requirements.

Swarm intelligence is a amazing fusion of hardware technology, software technology, biological observances and artificial intelligence.

Authors look forward to experiment upon advanced swarm robotic system in future.

Acknowledgements We thank Mr. Shreyas Malewar for his valuable comments and content of the paper. We would also like to show our gratitude to G. H. Raisoni College of Engineering, Nagpur and are also immensely grateful to our families for their continuous support and motivation.

References

1. Tan Y, Zheng Z (2013) Research advance in swarm robotics. *Defence Technol* 9(1):18–39
2. Roberts J, Zufferey J-C, Floreano D (2008) Energy management for indoor hovering robots. In: Proceedings of the international conference on intelligent robots and systems (IROS '08). IEEE Press, Piscataway, pp 1242–1247
3. Kantor G et al (2003) Distributed search and rescue with robot and sensor teams. In: Field and service robotics. Springer, Berlin, Heidelberg
4. Ferrante E (2009) A control architecture for a heterogeneous swarm of robots the design of a modular behavior-based architecture. Universite Libre de Bruxelles
5. Stirling T et al (2012) Indoor navigation with a swarm of flying robots. In: 2012 IEEE international conference on robotics and automation. IEEE
6. Borenstein J et al (1997) Mobile robot positioning: sensors and techniques. *J Robotic Syst* 14(4):231–249
7. Krishnan A (2016) Killer robots: legality and ethicality of autonomous weapons. Routledge
8. al-Rifaie MM, Aber A, Raisys R (2011) Swarming robots and possible medical applications. In: International society for the electronic arts (ISEA 2011), Istanbul, Turkey
9. al-Rifaie MM, Bishop M, Blackwell T (2011) An investigation into the use of swarm intelligence for an evolutionary algorithm optimisation. In: International Conference on evolutionary computation theory and application (ECTA 2011). IJCCI
10. al-Rifaie MM, Bishop JM, Blackwell T (2011) Resource allocation and dispensation impact of stochastic diffusion search on differential evolution algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2011). Springer, Berlin, Heidelberg, pp 21–40
11. Li M et al (2008) Robot swarm communication networks: architectures, protocols, and applications. In: 2008 Third international conference on communications and networking in China. IEEE

Chapter 17

Internet of Things: A Survey on Distributed Attack Detection Using Deep Learning Approach



Saraswati Nagtilak, Sunil Rai and Rohini Kale

1 Introduction

The next-generation network is moving toward the Internet of Things (IoT) where all kinds of network objects will be connected to each other seamlessly [1–3]. Many researchers in the field expect that IoT will drive the deployment of trillions of connected devices in the near future. The pillars of IoT are shown in Fig. 1. IoT is nowadays extensively deployed in a variety of fields together with power grids, smart buildings, health care, transportation, and entertainment. Due to rising IoT [4, 5] devices and data connected with them is huge in quantity; hence, the concern of security in the IoT was increased [6]. IoT security involves security for IoT infrastructure and IoT applications. IoT devices could be targeted easily by attacks as they are associated with exterior resources, and they do not include appropriate security at the network layer. In the same way, an attacker could be able to negotiate the network layer and IoT devices could be controlled [7], those can be used malevolently, or it can compromise other close-by gadgets associated with it.

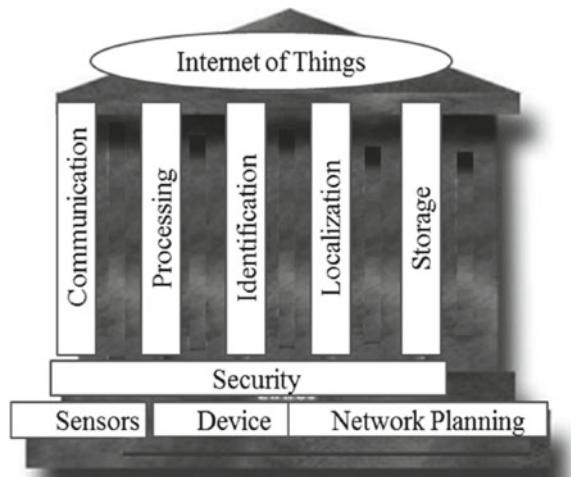
Several chances for attackers will be offered due to rising count of IoT devices to compromise them via collusion attacks, malevolent electronic mail, and denial of service (DoS) attacks among numerous kinds of attack [6].

S. Nagtilak · S. Rai (✉)
Department of Computer Engineering, School of Engineering,
MITADT University, Rajbaug, Loni, Pune, India
e-mail: vicechancellor@mituniversity.edu.in

S. Nagtilak
e-mail: sarakarande@gmail.com

R. Kale
MIT Polytechnic, Pune, India
e-mail: rohinikale@gmail.com

Fig. 1 Different pillars of IoT



A research has identified that diverse network attacks are offered by 70% of IoT devices, which utilize different susceptibilities, namely, password security encryption, etc. Therefore, system assaults are now being raising as a significant obstacle for the extensive deployment of networks using IoT, which could be identified at the network layers. On considering the “NSL-KDD dataset,” the network layer attacks are categorized into four kinds namely, (a) DoS, (b) Probe, (c) U2R, and (d) R2L [8]. IoT services that are based on cloud offer high ubiquity and reliability. Cloud computing remains ineffective to maintain the tasks, which require low latency, high storage, and costly computation in the IoT. These tasks include anomaly detection based on IoT cloud, in which the recognition of attacks will be carried out poorly [9, 10]. Furthermore, in central attack recognition system due to the extended distance of IoT devices from central system, it consumes more time for attack detection. Thus, more sophisticated framework should be designed so as to detect and defend the IoT devices from the attacks. The attack recognition mechanisms in wireless network perform on the cloud that could not gratify the different needs of the IoT, namely distribution, scalability, low latency, and resource limitations, and so on. As a result, security glitches of IoT cannot be resolved by neither cloud nor standalone attack discovery mechanism [6]. Consequently, for narrowing the disparity, a novel developing modus operandi centered on distributed intelligence, recognized as fog computing—a progression of cloud computing, should be explored. In fog computing, data handling and communication are done nearby the data sources [11]. In IoT, a number of control functions are carried out among varied devices. IoT networks can employ SVM for identifying the spoofing attacks and network interference, it can deploy K-NN for malware detection and network intrusion and it can exploit Neural Network (NN) to identify the DoS attacks and network intrusion [12]. Figure 2 shows the machine learning classification tree. In that supervised and unsupervised algorithms are listed of shallow and deep learning.

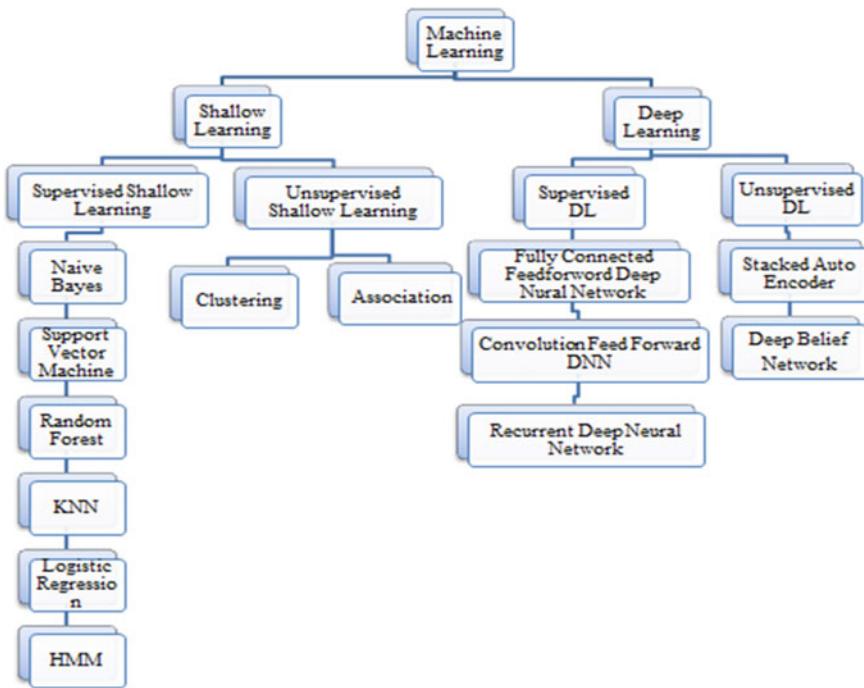


Fig. 2 Machine learning classification tree

1.1 Advantages of Deep Learning Model

- Reduces the need for feature engineering
- Possible to train massively generated data
- Unsupervised pre-training and compression capabilities in resource constraint nets
- These are able to find out nonlinear dormant variables with heterogeneous data.

1.2 Motivation

IoT is comprised of billions of people, things, and utilities along with the prospective to intermingle with every one and their surroundings. Because of the expanding number of IoT gadgets and the huge measure of information related to them, the issue of security has been raised. Different difficulties from a security, trust, and protection point of view are presented in IoT. Hence, to enable several existing and upcoming uses, security for IoT will be a basic worry that must be tended to. IoT comprises devices which have inadequate resources like PDAs, cell phones, RFIDs, and sensor nodes. Devices can be secured by using design process which is showed by

some aspects like great execution, low vitality utilization, and sturdiness to assaults. In the world of ubiquitous devices, some challenges present which need to be tackled. Following are challenges:

- Administration, extensibility, and diversity of gadgets
- Networked information and setting
- Protection, security, and trust should be improved to both devices and data.

2 Literature Survey

In 2018, Rathore and Jong [1] have suggested an attack recognition system, which was based on the fog computing concept and accordingly an ESFCM system was implemented. In this research work, attack detection was enabled by fog-based model at the network edge and it helps in detecting the distributed attacks which offers a faster rate of detection.

In 2018, Diro and Naveen [2] have established a methodology based on deep learning, which facilitates the attack detection in social IoT. In addition, the performance of the suggested distributed attack recognition method was evaluated against other conventional machine learning schemes.

In 2018, Kozik et al. [3] have revealed the reliability of cloud-based structures, alongside the current developments in the machine learning area for incorporating more storage-demanding and computationally costly tasks to the cloud so as to attain better computing performances for carrying out traffic classification effectively depending on ELM models, which were constructed over the cloud.

In 2019, Li et al. [4] have introduced a scheme, which intends to build up CBSig intrusion detection systems (IDSs) that could assist in incrementally building and updating a “trusted signature database” in a combined IoT surroundings.

In 2018, Yin et al. [5] have implemented an approach for software-defined IoT (SD-IoT) depending on software-defined anything (SDx) concept. In addition, a scheme was presented for identifying the DDoS attacks by means of the introduced model, and accordingly, here, the cosine similarity at the boundary of SD-IoT switch ports was deployed to find out if the DDoS attacks take place in IoT.

In 2018 Sun et al. [6] used Honeypots to store attack activities collected from Internet. Attacker groups are clustered by frequently merging the analogous nodes in a greedy approach has been confirmed effective and efficient. A Bayesian probabilistic graphical model and graph-based clustering algorithm are combined in a framework.

In 2018 Diro et al. [7] feature engineering is done by using pre-trained stacked autoencoder, while for classification softmax regression is used. NSL dataset is used for evaluation metrics like accuracy, DR, and ROC curve. Fog nodes are the most proficient spot for attack detection, accuracy and efficiency are increased due to DL in attack detection.

In 2015 Aggarwal et al. [8] random tree is used as a binary classifier for simulation on Weka classifies the instances as attack or normal. Attacker groups are clustered

by frequently merging the analogous nodes in a greedy fashion has been confirmed effective and efficient.

Table 1 shows the literature reviews on attack detection models.

Table 1 Features and limitations of attack detection in IoT systems

[Reference] Year of publication	Adopted methodology	Features	Limitations
[1] 2018	Fog-based attack detection framework with ESFCM scheme is developed	<ul style="list-style-type: none"> Lower time for attack detection Better accuracy Performance improved because load is distributed to several fog nodes 	<ul style="list-style-type: none"> Gives Lower performance. Random assignment of input bias and weights due to this ill posed problem occurs
[2] 2018	Fog nodes are used to train model and attack detection systems are implemented at the edge of the distributed fog network. DBN is used. Evaluation metrics accuracy, DR, and FAR are used	<ul style="list-style-type: none"> Reduced false alarm rate Better speed 	<ul style="list-style-type: none"> Requires analysis on real-time applications
[3] 2018	Extreme learning machine (ELM) models are used for classification	<ul style="list-style-type: none"> Proficient computations Better precision 	<ul style="list-style-type: none"> Time increases with increase in samples
[4] 2019	To reduce the influence of malicious nodes collaborative blockchained signature-based IDS (CBSigIDS) threat model is implemented	<ul style="list-style-type: none"> Improved robustness Better efficiency 	<ul style="list-style-type: none"> No deployment of blockchains for anomaly detection
[5] 2018	SD-IoT framework based on SDx is provided	<ul style="list-style-type: none"> Easier handling of attacks with less time SD-IoT quickly manages and mitigates the DDoS attack in IoT 	<ul style="list-style-type: none"> Requires analysis to secure against DDoS attacks in SD-IoT

(continued)

Table 1 (continued)

[Reference] Year of publication	Adopted methodology	Features	Limitations
[6] 2018	Honeypots are used to store attack activities collected from Internet. Attacker groups are clustered by repeatedly merging the analogous nodes in a greedy approach has been confirmed efficient and effective	<ul style="list-style-type: none"> • Cluster the attackers effectively based on their activity information • Great generality 	<ul style="list-style-type: none"> • Accuracy affected in case of unsynchronized attack
[7] 2018	For feature engineering, a pre-trained stacked autoencoder is used, while for classification softmax regression is used. NSL dataset is used for evaluation	<ul style="list-style-type: none"> • Fog nodes are used for attack detection • Accuracy and efficiency are increased due to DL in attack detection 	<ul style="list-style-type: none"> • Complexity
[8] 2015	KDD data set is selected Weka Tool is chosen for simulation. Analysis is done w.r.t. some evaluation metrics DR, FAR, accuracy, precision, specificity, and F-score	<ul style="list-style-type: none"> • Random tree algorithm, a tree-based classifier is used 	<ul style="list-style-type: none"> • NSL-KDD dataset should be used

3 Research Needs

- To prohibit profiling and tracking, localization and tracking by security framework.
- To detect attack accurately in less amount of time.
- To achieve better performance than centralized attack detection framework.
- To achieve better classification accuracy of normal data and attack data using deep learning.

4 Proposed Methodology for Attack Detection in IoT

Along with the evolvement of IoT, day by day security attacks are also growing. In IoT, various mechanisms of centralized attack recognition have been anticipated for attack identification in which an attack detection system is installed. Data from the system can be gathered centrally and categorizes it as attack data or normal data with the help of a supervised machine learning algorithm. Though, most of the mechanisms have unsuccessful to accomplish substantial outcomes because of the distinctive necessities of IoT devices, like scalability, dissemination, resource restrictions, and low latency. Hence, this proposal intends to develop a new model for enhanced distributed attack detection in IoT systems. Compared with other comprehensive security frameworks, here, the proposed framework will be able to detect attack in distributed manner and the evaluation can be done by using NSL-KDD [13] dataset, NSL-KDD is existing in csv format for model confirmation and estimations.

The proposed framework will be performed under two phases, the first is the feature extraction and the second one is the classification. The features namely, duration, service, protocol, flag, source bytes, destination bytes, etc., will be extracted and subjected for classification. The performance will be evaluated by using confusion matrix to classify the attack data and normal data.

4.1 Unsupervised Learning Method (Deep Belief Network)

Deep belief networks (DBN) are type of unsupervised learning strategies. Significant feature representation is achieved in DBN by trained unlabeled data iteratively. To reduce computational time DBNs uses contrastive convergence, and these networks are yet unsuitable to on-board devices with restricted resources. Unsupervised training with unlabeled sensor streams which is naturally available through cyber-physical systems and IoT deep belief neural network or merely DBNs is a likelihood-based generative graph model that is made out of various leveled layers of stochastic dormant variables having binary-valued activations, which are mentioned as unseen units or feature detectors. In DBNs, topmost layers have directionless, symmetric associates between them creating associative memory. The learning stage of DBN figures out how to reform its input, each layer going about as feature detectors. Greedy layer-wise training can be used to train DBN starting from the uppermost layer with raw input, and input data from the previous visible layer is used to train subsequent layers. For classification purpose, a deep learning algorithm will be used for attaining better classification accuracy and less detection time. The general design of the projected scheme is demonstrated by Fig. 3.

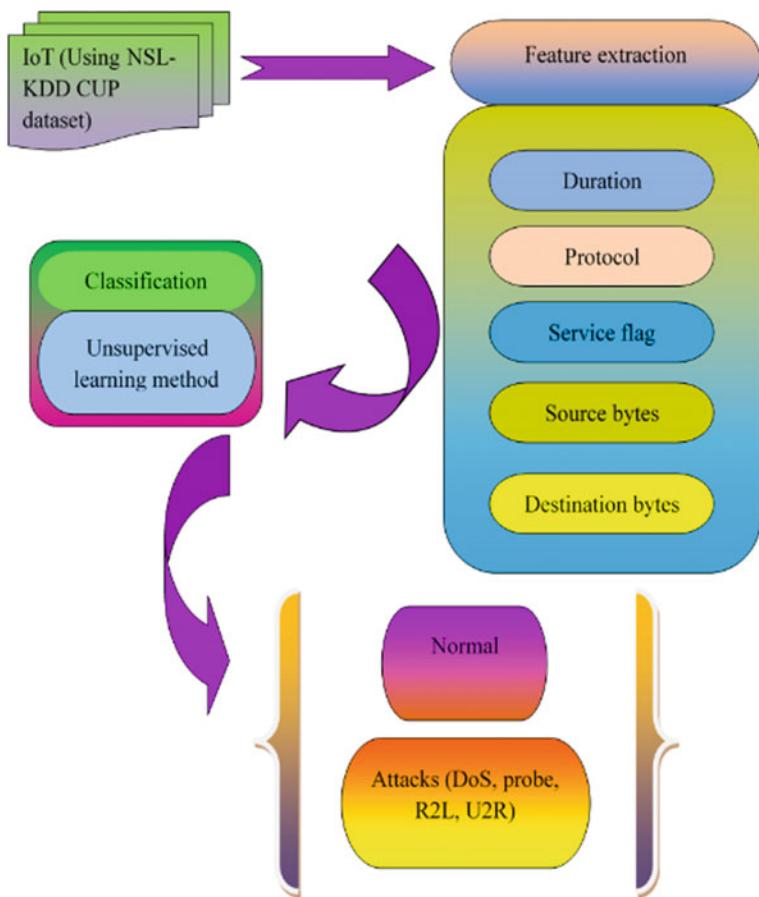


Fig. 3 General design of proposed attack detection in IoT

5 Conclusion

An enhanced distributed attack detection model is proposed. Feature extraction and classification will be done using deep learning. In IoT system, the attacks can be detected effectively using the proposed distributed attack recognition model. The performance of this attack detection model will be compared against conventional algorithms.

References

1. Rathore S, Park JH (2018) Semi-supervised learning based distributed attack detection framework for IoT. *Appl Soft Comput* 72:79–89
2. Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener Comput Syst* 82:761–768
3. Kozik R, Choraś M, Massimo F, Palmieri F (2018) A scalable distributed machine learning approach for attack detection in edge computing environments. *J Parallel Distrib Comput* 119:18–26
4. Li W, Tug S, Meng W, Wang Y (2019) Designing collaborative blockchained signature-based intrusion detection in IoT environments. *Future Gener Comput Syst* 96:481–489
5. Yin D, Zhang L, Yang K (2018) A DDoS attack detection and mitigation with software-defined internet of things framework. *IEEE Access* 6:24694–24705
6. Sun P, Li J, Bhuiyan MZA, Wang L, Li B (2018) Modeling and clustering attacker activities in IoT through machine learning techniques. *Inf Sci* 479:456–471
7. Diro AA, Chilamkurti N (2018) Deep learning: the frontier for distributed attack detection in Fog-to-Things computing. *IEEE Commun Mag* 56(2):170–175
8. Aggarwal P, Sharma S (2015) Analysis of KDD dataset attributes classwise for intrusion detection. In: 3rd International conference on recent trends in computing, pp 842–851
9. Singh A, Batra S (2018) Ensemble based spam detection in social IoT using probabilistic data structures. *Future Gener Comput Syst* 81:359–371
10. da Costa KA, Papa JP, Lisboa CO, Munoz R, de Albuquerque VHC (2019) Internet of things: a survey on machine learning-based intrusion detection approaches. *Comput Netw* 151:147–157
11. Nesa N, Ghosh T, Banerjee T (2017) Non-parametric sequence-based learning approach for outlier detection in IoT. *Future Gener Comput Syst* 82:412–421
12. Abawajy J, Huda S, Sharmin S, Hassan MM, Almogren A (2018) Identifying cyber threats to mobile-IoT applications in edge computing paradigm. *Future Gener Comput Syst* 89:525–538
13. Tavallaei M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the NSL-KDD cup 99 data set. In IEEE symposium on computational intelligence for security and defense applications, CISDA 2009. IEEE, pp 1–6

Part II

Image, Voice and Signal Processing

Chapter 18

Precise Orbit and Clock Estimation of Navigational Satellite Using Extended Kalman Filter Applicable to IRNSS NavIC Receiver Data



H. S. Varsha, Shreyanka B. Chougule, N. V. Vighnesam and K. L. Sudha

1 Introduction

An orbit is, mathematically, specified in terms of six orbital elements, determining the motion of satellite within a specified accuracy for a definite interval of time. In orbit estimation, the initial orbital parameters are refined. Observations or raw data are fed into orbit estimation algorithms. These observations may be ground-based time-tagged parameters like range, range rate, azimuth, elevation, etc. Having estimated the orbit, propagation techniques are used to predict the future positions of satellites. With time, the trajectory of satellite deviates from the predicted path owing to various perturbations experienced by it such as third body forces, and solar radiation pressure. The orbit estimation algorithm requires the modeling of these forces for precise prediction [1].

Satellite Application Centre (SAC), ISRO installed NavIC receiver in the Department of Electronics and Communication, Dayananda Sagar College of Engineering (DSCE), Bengaluru, India. Systematic and routine measurements are being carried out from the network of IRNSS satellites. The range measurements received by

H. S. Varsha (✉) · S. B. Chougule

Department of Electronics and Communication/Mathematics, Dayananda Sagar College of Engineering, Bengaluru, India

e-mail: varshahs29@gmail.com

S. B. Chougule

e-mail: shreyankabc594@gmail.com

N. V. Vighnesam

Department of Mathematics, Dayananda Sagar College of Engineering, Bengaluru, India

e-mail: vighnesam@gmail.com

K. L. Sudha

Department of Electronics and Communication, Dayananda Sagar College of Engineering, Bengaluru, India

e-mail: klsudha1@rediffmail.com

the ground receivers have multiple errors added due to ionospheric delay, tropospheric delay, multipath delays, satellite clock errors, satellite orbit errors, and due to some unmodeled effects. But, the major factor accounting to errors is the satellite clock error in IRNSS system. Due to lack of actual data from other ground stations, simulated data is being used in this paper for orbit estimation.

The process of satellite's orbit estimation includes trajectory generation [2] using initial state parameters by numerical integration, measurement modeling and finally estimation of the satellite position, velocity and clock error. For the trajectory generation, numerical integration method is used which is based on implicit Runge–Kutta methods (RADAU IIA) with variable order and step size control. Radau methods belong to the class of fully implicit Runge–Kutta methods.

Range measurements are generated for a network of tracking stations at every instant of time using satellite ephemeris and station coordinates. At any time t , station coordinates in cartesian form are computed using initial station latitude (ϕ), longitude (λ), and altitude (h) as,

$$X_s = G1 \cos \phi \cos \alpha \quad (1)$$

$$Y_s = G1 \cos \phi \sin \alpha \quad (2)$$

$$Z_s = G2 \sin \phi \quad (3)$$

where α is the right ascension given by $\alpha = \lambda - \theta$ and θ is the sidereal angle. For orbit estimation using multiple stations, tracking data has been simulated for seven Indian ground stations positioned at Ahmedabad, Bengaluru, Hassan, Lucknow, Trivandrum, Delhi, and Port Blair.

Most commonly used estimation techniques are batch estimation and sequential estimation [3]. Orbit estimation is used to improve apriori orbital elements from a large set of tracking data. The batch or least square estimator processes a whole set or batch of data at once and improves the epoch state estimate, but the sequential estimator yields estimates of the state vector at each and every measurement time by processing one measurement at a time. Extended Kalman filter (EKF) is a sequential estimation technique which estimates the state parameters at each epoch using range measurements. The EKF is powerful that it works well for non-linear functions and in order to accommodate models of error characteristics; it uses dynamical (process) noise models. In cases where the dynamical models are vaguely known and the sequential update of the state vector is required, orbit determination applications are highly useful [4].

In this paper, EKF is used for orbit estimation and the algorithm is explained in Sect. 2. The details of the ground stations considered; initial conditions and the entire process involved in the estimation are discussed under methodology in Sect. 3. The results observed are tabulated and discussed in Sect. 4.

2 Extended Kalman Filter (EKF) Algorithm

The EKF algorithm is a sequential estimating algorithm which is recursive in nature. It uses initial conditions and the present measurement to estimate the state vector. Filtering can be started by knowing the initial state, state noise variance and measurement noise covariance. The predicted measurement is compared with the current measurement (as and when it is available), the residual is computed, and current state is estimated using Kalman gain [5]. Some advantages of Kalman filter are it is accurate, fast, etc. when compared to differential correction algorithm; it is suitable for real-time application because of recursive nature. Most of the time, the variance does not increase infinitely and hence the EKF estimated state is stable.

Algorithm for EKF:

Notations:

- $[X_0, P_0]$ is the estimated state vector and its covariance, respectively
- f —state transition function
- g —measurement function
- Q —covariance of the process noise
- R —covariance of the measurement noise
- Z —set of pseudo ranges measured from the receiver to the satellite (actual measurement)
- P —‘priori’ estimated state covariance
- $X = [x \ V_x \ y \ V_y \ z \ V_z \ b \ d]$ is the initial state vector with x, y, z being the coordinates of the initial satellite position and V_x, V_y, V_z are the velocities in x, y, z direction, respectively.
- ‘ b ’ denotes the initial clock bias
- ‘ d ’ is the clock drift value.

Steps:

For an ordinary Kalman filter, to get the state transition matrix, f_y (jacobian of the process model) and the observation matrix, H (jacobian of the measurement model), it is required to linearize input functions f and g .

1. $X_p = f(X)$ is a one-step projection which provides linearization point
2. $f_y = f(X_p)$

$$f_y = \left. \frac{df}{dX} \right|_{X=X_p}$$

This is a linearized equation of state
3. $H = \left. \frac{dg}{dX} \right|_{X=X_p}$

$$H = g(X_p)$$

This is a linearized equation of observation
 $[g X_p, H] = g(X_p) : g X_p$ is the predicted measurement calculated using the station coordinates and the estimated satellite position in the previous step.
4. The covariance of X_p

$$P_p = f_y * P * f'_y + Q$$
5. Kalman Gain

$$K = \frac{P_p * H'}{H * P_p * H' + R}$$

6. The output state X_0

$$X_0 = X_p + K * (Z - gX_p)$$

The difference between the actual and predicted measurement ($Z - gX_p$) is called as residual. The present state is estimated using the predicted state, predicted measurement, and the new measurement.

7. Covariance of X_0

$$P_0 = [I - K * H] * P_p$$

3 Methodology

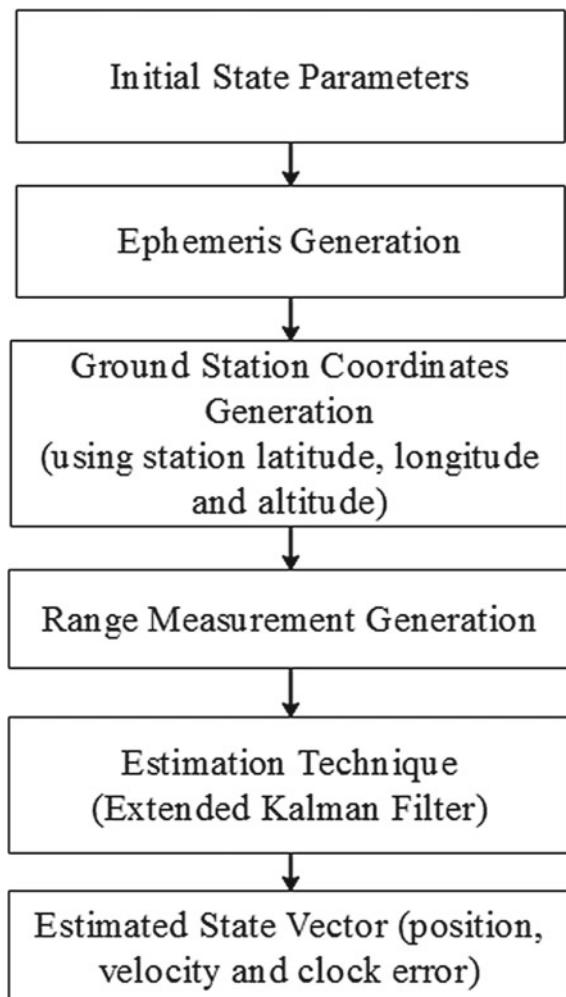
The satellite ephemeris file is generated using the trajectory generation (orbit propagation) program. This requires an initial state vector, start epoch to be given as an input. Also, the duration for which the ephemeris has to be generated should be mentioned along with the step size (time interval) (Fig. 1).

In this present study, the initial conditions considered for the ephemeris generation are:

- Start date (yyyy/mm/dd): 2006/11/27
- Start time (hr:min:s:ms): 00:10:00:000
- Time interval/Step size (s): 1 s
- Time duration: 1 week (Table 1).

The satellite ephemeris is generated in ECI reference frame for this study. The station coordinates are computed for every time instant ‘ t ’ using station latitude (ϕ), longitude (λ) and altitude (h). The start and end time is to be mentioned for the duration for which the station coordinates are computed. The (ϕ, λ, h) values for the seven Indian stations considered are given in Table 2.

Using the satellite ephemeris and the station coordinates, the range measurements are generated for every time ‘ t ’. The orbit estimation program using EKF algorithm, presented in Sect. 2, is written in MATLAB. The inputs for this program are the observed range measurements (generated from the measurement program), minimum of four station coordinates, initial state vector, state noise covariance (Q) and measurement noise covariance (R). Since only simulated data is used, the noise covariance is considered to be zero. It will vary when working with real-time data. EKF estimates the position (x, y, z) and velocity (x', y', z') of the satellite at every time instant and along with the clock bias present in the range measurement. Since there is no bias present in the generated measurements, clock bias of 10,000 is added to the generated range measurements. So the program is expected to estimate the clock bias present in the measurement as 10 km and also the satellite position at every time instant.

Fig. 1 Methodology

4 Results and Observations

In this section, different cases of input and their results have been discussed. The estimated position and velocity are compared with the satellite's true ephemeris to verify the satellite position determined by the EKF algorithm. The difference between the true satellite ephemeris and the estimated satellite positions at every time instant gives the error in position. Along with the state vector ($x, y, z, \dot{x}, \dot{y}, \dot{z}$), clock bias is also estimated at each instant. The four ground stations considered are all widely separated, namely, Ahmedabad, Bengaluru, Lucknow, and Trivandrum.

Table 1 Initial elements

<i>Orbital osculating elements</i>		
Semi major axis in meters	a	42,166,024.34
Eccentricity	e	0.00038904
Inclination in degrees	i	0.047595393
Argument of perigee in degrees	sw	131.8774784
Right asc. of node in degrees	co	260.1294659
Mean anomaly in degrees	m	205.0820557
<i>Cartesian coordinates</i>		
Initial position in meters	x	-22,930,044.95
	y	-35,403,950.77
	z	-13,724.42975
Initial velocity in m/s	x'	2579.982591
	y'	-1670.372217
	z'	2.349319

Table 2 Ground station considered for the study

Ground station	Latitude (ϕ)	Longitude (λ)	Altitude (h)
Bengaluru	77.51168611	13.03444722	0.83991
Ahmedabad	27.45612	23.0404	0.048
Lucknow	80.95731667	26.91307778	0.0731
Hassan	76.0973628	13.071152	0.9161465
Delhi	77.180376	28.6086864	0.2106
Trivandrum	76.87306	8.53639	-50
PortBlair	92.71242	11.63721	45

Case 1: Well separated ground stations and clock bias = 0

In this case, no clock bias has been introduced. The range measurements are generated for every second for one week. The EKF estimates the satellite position for each second. The estimated satellite position (X, Y, Z) in meters is compared with the true satellite position. Mean position error of 11.06 cm with a standard deviation of 0.07807 and variance of 0.00609 is observed. Also, the mean velocity error is found to be 0.2342 m/s and the clock error is estimated to be 0.0038 m. In this case, the standard deviation of velocity error is 0.1722 m/s. Position error (m), velocity error (m/s), and estimated clock bias (m) have been plotted for every one hour for one day's data. Figures 2 and 3 are the graph of position error and velocity error, respectively. The estimated clock bias has been plotted in Fig. 4.

Case 2: Well separated ground stations and clock bias = 10,000 m

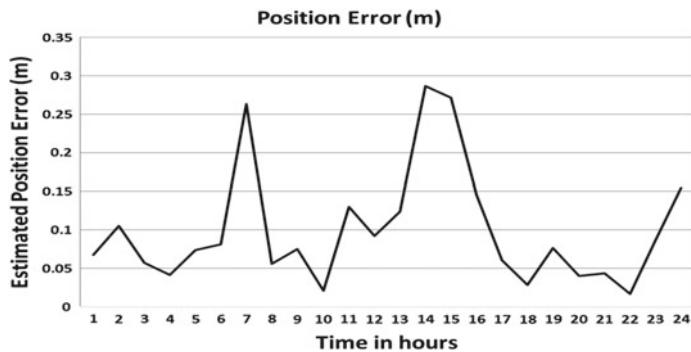


Fig. 2 Graph of position error (m) for case 3

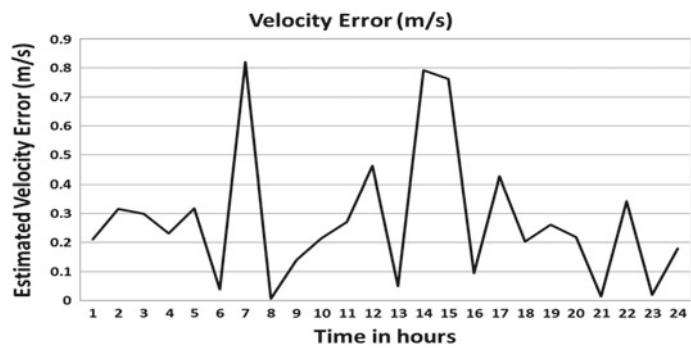


Fig. 3 Graph of velocity error (m) for case 3

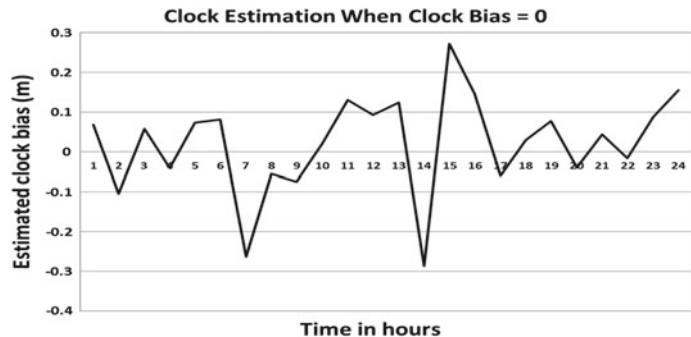


Fig. 4 Graph of estimated clock bias (m) for case 1

In this case, clock bias of 10,000 m has been added. The range measurements are generated for every 1 s for an entire week. Mean position error of 11.10 cm with a standard deviation of 0.07836 and variance of 0.00614 is observed when clock bias is 10,000 m. Also, the mean velocity error is found to be 0.2365 m/s and the average clock error is estimated to be 9999.997 m. In this case, the standard deviation of velocity error is observed to be 0.1741 m/s which is less when compared with standard deviation of velocity error 1.08 m/s observed in the case of closely spaced ground stations. Position error (m), velocity error (m/s), and estimated clock bias (m) have been plotted for every one hour for one day's data. The position error and velocity error have been plotted in Figs. 5 and 6, respectively. The estimated clock bias for this case has been plotted in Fig. 7.

In all the cases, the state parameters and the clock bias are estimated with actual initial condition as well as disturbed initial condition up to 100 km in (x , y , z) directions. It is observed that the algorithm is capable of estimating the position, velocity and clock bias precisely even with much-disturbed initial parameters.

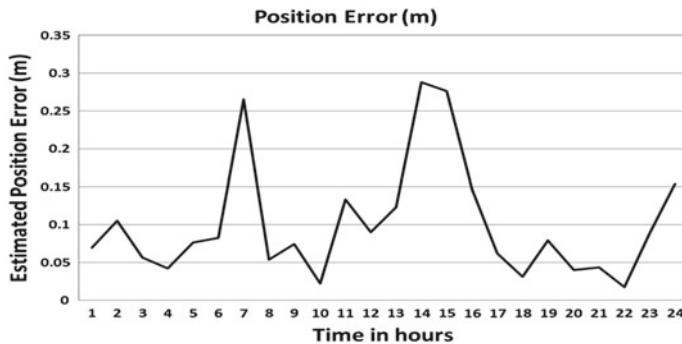


Fig. 5 Graph of position error (m) for case 2

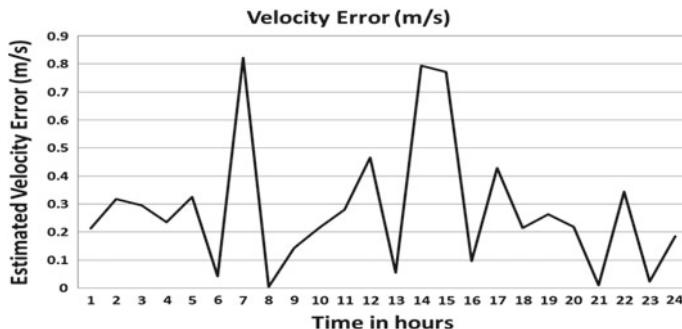


Fig. 6 Graph of velocity error (m) for case 2

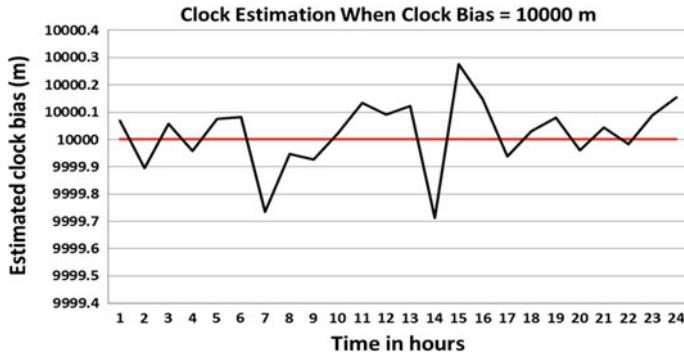


Fig. 7 Graph of estimated clock bias (m) for case 2

5 Conclusion and Future Scope

The aim of this paper is to determine the satellite orbit and to estimate the satellite clock bias. The methodology involved in orbit estimation is briefed and the algorithm adopted in this paper, EKF, is explained in detail in Sect. 2. To have a better understanding, orbit determination is done using closely spaced and widely separated ground stations. Summary of the results obtained is given in Table 3.

It is evident from the table that with well-separated ground stations, precise orbit estimation, and clock error estimation are successful even with disturbed initial guess.

We notice from this study that errors in position and velocity obtained using well-separated ground stations are reduced about 80% in both position and velocity in comparison with closely spaced ground stations data. These errors can be further improved by considering data from globally separated ground stations. This algorithm can be used for orbit estimation using actual IRNSS NavIC receiver data.

Table 3 Summary of results obtained

	Position error (m)			Velocity error (m/s)			Estimated clock bias (m)
	Mean	Standard deviation	Variance	Mean	Standard deviation	Variance	
Closely spaced ground stations with no clock bias	0.6963	0.49942599	0.249426	1.4446	1.07483048	1.155261	0.0126397
Closely spaced ground stations with clock bias introduced	0.6975	0.50045035	0.250451	1.4521	1.08082244	1.168177	9999.9875
Widely separated ground stations with no clock bias	0.1107	0.07807454	0.006096	0.2342	0.17226728	0.029676	0.0038324
Widely separated ground stations with clock bias introduced	0.1111	0.07836244	0.006141	0.2365	0.1742875	0.030376	9999.9968

Acknowledgements The work undertaken in this paper is supported by SAC/ISRO under NavIC GAGAN utilization program (Project ID: NGP 27), at Space Application Centre, Ahmedabad. Authors wish to acknowledge the support given by SAC/ISRO and wish to express their gratitude to the focal person of this project, Dr. Ashish Shukla, Scientist/Engineer-SF, Satellite Navigation Technology Division, Space Applications Centre, ISRO, Ahmedabad. Also, the authors would like to thank Dr. Radha Gupta, Professor & HOD, Department of Mathematics, Dayananda Sagar College of Engineering for her constant support and the Principal, Dayananda Sagar College of Engineering, Bengaluru.

References

1. Montenbruck O, Gill E (2001) Satellite orbits, models, methods and applications. Springer
2. Bate R, Mueller D, White J (1971) Fundamentals of astrodynamics. Dover Publications
3. Pratap M, Per E (2006) Global positioning system: signals, measurements and performance, 2nd edn. Ganga-Jamuna Press, New York

4. Hobbs D, Bohn P (2019) Precise orbit determination for low earth orbit satellites
5. Sarunic PW (2016) Development of GPS receiver Kalman Filter algorithms for stationary, low-dynamics, and high-dynamics applications. Defence Science and Technology Group Edinburgh Australia. Online: <https://apps.dtic.mil/dtic/tr/fulltext/u2/1010622.pdf>

Chapter 19

Effects of Color on Visual Aesthetics Sense



Shruti V. Asarkar and Madhura V. Phatak

1 Introduction

The study of psychology behind the color is nothing but the study of hue. Color affects perception that is not clear (e.g., food's taste). Color influences a person's mood. With the help of color, it is easy to convey information, create certain moods and also influence on the person's day-to-day decisions regarding activities like to decide which cloths to wear, which car to purchase, choosing color of house (e.g., if someone is preferred to paint the room with soft-green color, this soft-green color creates peaceful mood). Color impacts on productivity, attention and also learning. There is some connection or say relation between color and children's brain development. So in support of attention and learning, it is necessary to use colors in educational institutes. Colors generally encounter with environments, people as well as objects; Colors are also there in dreams [1].

Here in this study of visual aesthetics sense, we are considering visual sense, images, and videos that are important gradients in day-to-day life. It can elevate/depress mood of person. Visual aesthetics improve user satisfaction. With the rise in popularity of DSLR camera, mobile dual camera, the amount of visual data available on the Internet is expanding exponentially. We people capture, watch, share, compute, store photos and videos, and for appreciation, we upload these photos and videos on social media. Appreciation of photos and videos is totally based on "Likes" given by people, and "Likes" depends on how beautiful video and photos are. Finding aesthetics of videos is very important, for wide applications such as in cinematography, to show beautiful high content videos, search and recommendation, UI design,

S. V. Asarkar (✉) · M. V. Phatak

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

e-mail: asarkarshruti@gmail.com

M. V. Phatak

e-mail: Madhura.phatak@mitpune.edu.in

© Springer Nature Singapore Pte Ltd. 2020

S. Bhalla et al. (eds.), *Proceeding of International Conference on Computational Science and Applications, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-15-0790-8_19*

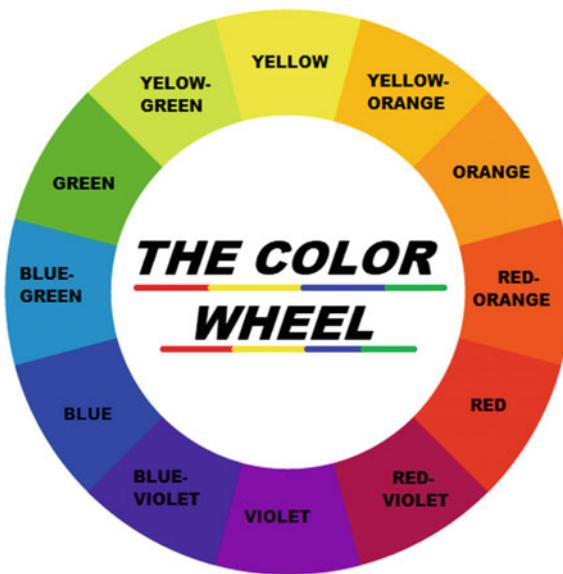
revenue generation in advertisement world, and social media Web sites too. Some of the videos and pictures are aesthetically pleasing and beautiful to human eyes. But there are some videos as well as photos which are uninteresting and of low quality. The reason behind those uninteresting and low-quality photos and videos may be color combination. Also in cinematography, color of movies and films creates mood of viewer. In the world of cinematography, it is very important to understand the psychological effects of colors that impact viewer's emotion and mood.

1.1 Psychology Behind Color Aesthetic Appreciation

In our daily life, we people are making some decisions or judgments on the basis of aesthetics things that are interested to our visual sense. Visual sense is more dealing with the color. For example, we prefer a particular cloth which has to be worn or purchased than another one, and here, we are actually dealing with the color of that cloth. Even if we are going to the park, then in which direction we have to sit is decided on the basis of which view is more pleasing to our eyes [2]. This means color aesthetics is having a great value in human psychology. Color adds aesthetic quality to visual electronic display. Also with the help of colors, it is easy to convey the complex form of information to viewers. In marketing field, color plays a major role, since color attracts the customer [3]. Color is an important variable in the marketing literature [3], fashion, cinematography, revenue generation in advertisement world, and many more.

1.2 Color Wheel

Color is also known as non-verbal form of communication. Hue, value, and chroma are three well-defined qualities of color [4]. In school times, we already learned about basic primary colors, that is red color, yellow color, and blue color. If we mix any of the two colors, we will get another color. Color wheel as shown in Fig. 1 is a easy way for visual understanding of the relationship between colors. If we mix basic primary colors, we will get secondary colors (that is, green, orange, and violet). Further, if we combine these primary colors with secondary colors, the result will be tertiary colors (that is, combination of orange-red, violet-blue, yellow-green, violet-red, green-blue, and orange-yellow).

Fig. 1 Color wheel

1.3 Color Schemes Used Commonly in Cinematography

The color scheme produces color harmony. Color harmony is nothing but when any two colors are shown in the color palette, places in the neighboring area generate an appealing effect, then say they are producing color harmony [5].

Two colors that are placed oppositely on color wheel are known to be visually pleasing to human eyes. Opposing colors are aesthetically appealing to humans and called “complementary color scheme” [6]. A very common example of the orange and blue colors produces vibrant and contrast results. It is actually a combination of warm color with cool color. Warm and cool actually create a mood. Warm colors contain red color, orange color, and yellow color. All the warm colors in the videos or photos are containing some red, or orange, or yellow within them. Warm colors are evident in fire or sun scenes, whereas cool colors placed on color wheel are oppositely located to the warm colors. Cool colors show relaxation and calmness. These cool colors are restful, positive, soothing, and due to this property, cool color creates harmony. Span of cool colors on color wheel is from green via blue to violet. These colors are known to be cool colors because these colors always remind we people about the scenes like ocean or water [7]. Warm colors psychologically make us feel warm [8], whereas cool colors psychologically make us feel peaceful, soothing, relaxed, and calm.

From color wheel, if we select any three hues that alongside to each other, then those colors are known as “analogous color scheme” [6], for example, selecting blue, blue-green, and yellow or selecting the colors such that those colors are generally from relatively family of colors. By selecting analogous colors, color harmony is

being created. The best examples of analogous color includes, (i) in fall season, changing the color of leaves (ii) colors that changes during sunset or sunrise.

Selecting evenly spaced three hues on color wheel creates “triadic color scheme” [6]. For example, if we select colors violet, orange, green, it creates a vibrant mood. If colors are carefully balanced in these color schemes, then it creates harmony. It is least used and less commonly used color schemes in cinematography world.

In traditional cinematography, whenever there is a need to show some strong sense of simplicity events, then monochromatic theme is mostly preferred. That means selecting a single or say mono color in video from color palette, to create harmonic atmosphere [9]. This monochromatic sense creates memorable impression on viewer.

1.4 Color-Mood Analysis in Video

Videos are important media that convey human emotions. Color preference, content, and motion of video are important components of visual experience. These aspects affect a wide spectrum of human mood. Considering color of videos, that sends “approach” signals (e.g., good colors of videos attract people to watch) and sometimes sends “avoid” signals (e.g., the color that creates irritation to eyes). There is some correlation in between color and mood. Color-mood analysis is a psychological study and study of cinematographic aspect, in which we have found studies about the color-mood association. According to [10], mood is far different concept than emotion. Emotions are for small period of time, whereas moods are usually prolonged time period. Every color represents specific moods: Brighter warm colors energize viewer and make more alert, whereas dark cool shade colors tend to be soothing and relaxing. Color association is not made by single term of “emotion.” For example, red color shows love, passion, violence, danger, anger, power, etc. Instead, not only red color is a color, which is related love, but the term called love is also allied with red and violet color combination. According to web survey, we have established association of color and mood as shown in Table 1.

1.5 Positive and Negative Psychological Properties of Colors

The term psychology of color is associated with the electromagnetic propagation or say radiation of light effects on human behavior or mood. A single color can evoke positive as well as negative emotion. For example, red color can evoke positive emotion like love and negative emotion like anger. This means, there are multiple and many more effects of a single color. Table 2 shows these positive and negative traits on each color.

Table 1 Color-mood association [11]

Color	Mood
Red	Anger, violence, passion, war, rage, blood, desire, fire, excitement, aggression, energy, love, speed, heat, power, strength
Pink	Love, feminine, innocence, delicate, healthy, soft, happy, playfulness, content, charming, romantic
Yellow	Wisdom, hazard, knowledge, illness, relaxation, deceit, joy, covetousness, happiness, jealousy, optimism, betrayal, idealism, cowardice, imagination, summer, sunshine, hope
Orange	Humor, flamboyant, energy, expansive, balance, vibrant, enthusiasm, warmth
Green	Healing, envy, soothing, inexperience, perseverance, jealousy, tenacity, fertility, self-awareness, generosity, proud, spring, unchanging nature, vigor, environment, youth, healthy, renewal, good luck
Blue	Faith, depression, spirituality, technology, contentment, cold, loyalty, water, fulfillment, peace, sky, tranquility, order, calm, cleanliness, stability, security, harmony, conservatism, unity, confidence, truth, trust
Purple/violet	Erotic, intimacy, royalty, sensitive, nobility, power, spirituality, mourning, ceremony, arrogance, mysterious, cruelty, transformation, enlightenment, wisdom
Brown	Reliability, materialistic, simplicity, sensation, stability, earth, endurance, home, comfort, outdoors
Black	No, anger, power, remorse, sexuality, sadness, sophistication, evil, formality, style, elegance, depth, wealth, unhappiness, mystery, anonymity, fear
White	Yes, sterile, protection, clinical love, cold, reverence, death(Eastern culture), purity, marriage(Western culture), simplicity, sterility, cleanliness, good, peace, snow, humility, youth, winter, precision, birth, innocence
Silver	Riches, high-tech, glamorous, elegant, distinguished, sleek, earthy, natural
Gold	Precious, grandeur, riches, prosperity, extravagance, wealth, warm

2 Color Preferences

For visual experience, color preference is the most powerful, significant, and important aspect. Color preference influences a wide diversity of human behaviors: buying assets, choosing cloths, designing websites, etc. If we focus on the process of the human's brain, we get to know that how sex difference, age difference and cultural variation around the human affect on their color choices. According to studies, color preference departs from person to person. In [12], authors made hypothesis about color preference—color preferences might also vary according to sexual orientation. According to author, females are choosier about softer colors and are sensitive to pinks, reds, and yellows, while males generally prefer bold colors and they seem to be more sensitized to colors in the blue-green light spectrum. Nearly 45% males vote their favorite color as blue or shades of blue, whereas 24% females gave their vote to blue and shades of blue. Not only gender affects color preferences; it also varies by

Table 2 Positive and negative traits of colors [8]

Color	Positive trait	Negative trait
Red	Intelligence, calm, communication, reflection, trust, coolness, efficiency, serenity, logic, duty	Lack of emotion, anger, aloofness, unfriendliness, coldness
Pink	Physical tranquility, species survival, nurture, sexuality, warmth, love, femininity	Emasculation, inhibition emotional, claustrophobia, physical weakness
Yellow	Optimism, creativity, confidence, friendliness, self-esteem, emotional strength, extraversion	Suicide, irrationality, anxiety, fear, depression, emotional fragility
Orange	Physical comfort, fun, food, abundance, warmth, sensuality, passion, security	Immaturity, deprivation, frivolity, frustration
Green	Harmony, peace, balance, equilibrium, refreshment, environmental awareness, universal love, restoration, rest, reassurance	Enervation, boredom, blandness, stagnation
Blue	Reliability and responsibility, loyalty, caring and concern, peaceful and calm, trust and integrity, idealistic and orderly, devotion and contemplation, conservatism and perseverance, tactful, authority, caring and concern	Depressed and sad, aloof and frigid, being rigid, unforgiving, deceitful and spiteful, predictable and weak, too traditionalist and old-fashioned, too passive, superstitious, and emotionally unstable, self-righteous. It also can indicate being untrustworthy, manipulation
Purple/violet	Quality, luxury, spiritual awareness, truth, containment, authenticity, vision	Inferiority, introversion, suppression, decadence
Brown	Earthiness, seriousness, support, warmth, reliability, support	Heaviness, lack of humor, lack of sophistication
Black	Sophistication, substance, glamor, efficiency, emotional safety, security	Coldness, heaviness, menace, oppression
White	Cleanliness, hygiene, efficiency, sterility, sophistication, clarity, simplicity, purity	Elitism, barriers, sterility, unfriendliness, coldness
Silver	Glamour, sleek, illumination, modern, reflection, wisdom, feminine power, insight, balancing, organization, calming, responsibility, soothing, self-control, dignity	Deceptive, dull, insincere, melancholy, indecisive, lonely, neutral, lifeless and colorless, negative, rigid
Gold	Optimistic, success, positive, and masculine, abundance, winning, wealth, charisma, compassion, understanding, passion, self-worth, love, wisdom	Fear of success, falseness, fear of wealth, lack of trust, self-centered, mean spirited, demanding

age group. As mentioned in [13], adults mostly prefer blue color than yellow. Sometimes cultural factors also imply color preference. According to the survey presented in [12], there is a tendency in Western cultures—pink color is preferred to dress baby girls and for baby boys blue is preferred. Research on cross-culture elucidates on the issues by discovering how differed gender makes differences in color preferences. According to [14], color combinations having same hues are generally preferred. Color preference in case of aesthetics differs basically with lightness contrast.

2.1 Color Preference According to Gender

When new babies are born, generally if baby is boy then we can see that he is wrapped in blue and if baby is girl then we can observe that she is wrapped in pink color [15]. Ellis and Ficek [12] collected relevant data from different studies and research on color preference by gender, in which the color preferences are distributed by gender. From colors red, pink, purple, yellow, orange, green, blue, brown, gray, black, white; resultant of survey shows 45% of males preferred blue color and 0.5% of males preferred pink color, whereas 27.9% of females preferred green color and 0.02% of females preferred gray color. Maximum males preferred blue color, and 24.9% females also preferred Blue. Less males preferred pink color, but 5.3% females preferred pink color. Green color is a favorite color for females, and 19.1% males also preferred green color. Least favorite color for females is gray, and only 1.1% males preferred that color.

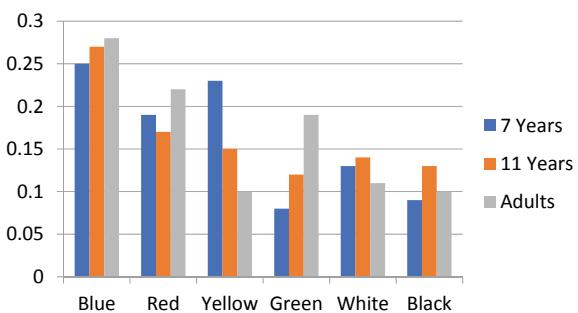
As technology is growing, we are more familiar to social sites like Facebook, Instagram, Twitter, etc. Many of the social sites provide services to change theme of different color. According to [16], from red, yellow, green, cyan, blue, magenta colors, 46.0% males and 46.1% females preferred cyan color for theme, whereas 1.4% of males and 1.5% of females preferred green color.

As per marketing point of view, Moss and Colman [17] prefer example of business card, 74% of male designer used white card, compared with 53% of female designer. According to this study, author finds some conclusion that women are more interested in colors than men. Men are concerned with functions, but women are more concerned about aesthetics. From all of this survey, we can observe that either males or females prefer the color according to situation, things, etc.

2.2 Color Preference According to Age

When any child is born, their parents often choose blue color toys or cloths for boys, whereas pink color of cloths or toys for girls. Do really newborn babies can interact with or can they see colors differently than adults? According to scientific research, at birth babies cannot see any person, anybody, or anything. When babies turn to 5 months, then their eye movements and skills of eye-and-body coordination

Fig. 2 Color preferences in all age groups [21]



improve continuously. Although the color vision of babies is not as sharp or sensitive as adults, babies generally have good color vision by five months of age [18]. Color affects the mood more so in children than in adults [19]. Color can impact learning and memory in kids. For pre-school and elementary school, warm color is ideal color schemes, and in secondary school, cool colors are ideal [19].

According to [13], among the light colors, infants mostly prefer red color than blue and among the dark colors yellow color is preferred than green. According to survey shown in [20], blue, purple, and green are mostly preferred among all age groups (age groups from 1 to 70+). As people get older, the purple color preference is increased and green color preference is decreased [20]. The following table shows the survey of color preference according to age [21] (Fig. 2).

2.3 *Color Preference According to Culture*

According to [21], as the child grows, their preferences change according to the result of social as well as cultural influence. Some colors are symbolism of every country. Like green color symbolizes Islam and Pakistan [22]. According to study [23], there is a comparative study in between the mentality of the people belonging from two different countries that is Indian and British. According to that, it is noted that Indians give higher ranking to light-green color and olive-green color than light-blue color and lavender color and British give higher ranking for light-blue color and lavender color. As per [20], color preferences according to culture and meaning of each color varied by cultural difference, noted in Table 3.

3 Results

The overall aim of writing is to communicate survey of color preferences and choices according to person's age group, gender, profession. This is the survey done in April 2019, and some questions were asked to people about their attitudes toward the color

Table 3 Color preference according to culture

Color	Western culture	Eastern and Asian culture	Latin America	Middle east	Around the World
Red	Passion and excitement	Joy and celebration [India—purity, China—luck and happiness, Japan—life (anger and danger)]	Religion color (when associated with white)	Danger and caution	Luck, good fortune, and prosperity
Orange	Color of harvest and autumn (USA—beginning of fall season, Netherlands—national color, color of royalty)	India—color of scared, Japan—courage and love	Sunny	Mourning and loss	Color of gluttony in Christianity
Yellow	Associated with warmth, summer, and hospitality (USA—transportation; school bus, taxies, Germany—associated with envy)	Color is considered sacred and imperial (Japan—courage, India—color of commerce)	Associated with mourning and death	Happiness and prosperity	Associated with money, quality, and success
Blue	<ul style="list-style-type: none"> • Trust and authority (accepted blue color logo for banks) • Calming, peaceful, and soothing • Represents baby boy birth 	Association with immortality (India—symbol of strength, China—feminine color)	<ul style="list-style-type: none"> • Associated with religion • Association with mourning so can cause emotional stir 	Heaven, spirituality, and immortality	Positive and safest color

(continued)

Table 3 (continued)

Color	Western culture	Eastern and Asian culture	Latin America	Middle east	Around the World
Green	<ul style="list-style-type: none"> Represent luck Color associated with nature 	<ul style="list-style-type: none"> Fertility and youth Negative connotations (China—wearing green hat symbolizes cheating on spouse) 	Color of death	Luck, strength, fertility, and wealth (majority of Islam)	Almost every military's color those are active in the world
Purple	<ul style="list-style-type: none"> Color of royalty Associated with fame as well as wealth Modernism and progressive (USA—color of honor) 	Wealth and nobility	Associated with death and mourning (South America)	Wealth	—
Pink	<ul style="list-style-type: none"> Color of femininity Represents birth of daughter Signify sweetness, Childhood, or fun 	<ul style="list-style-type: none"> Signify feminine Also signify marriage in East (Korea—color of trust) 	Associations with architecture (building colors)	No distinct meaning	Calming color
Brown	<ul style="list-style-type: none"> Associated with either barrenness or health Dependable, wholesome, stability, and color of grains (USA—for packaging and food containers) 	Shows mourning	Discouraging and disapproving	Tuneful with comfort and earth	Because of its neutral tendencies, it is known as non-color

(continued)

Table 3 (continued)

Color	Western culture	Eastern and Asian culture	Latin America	Middle east	Around the World
Black	<ul style="list-style-type: none"> Europe—and North America—color of death, finality, mourning, and formality Considered strong as well as powerful and also can imply force or control 	<ul style="list-style-type: none"> Masculinity Prosperity, health, and wealth China—color for boys. Thailand and Tibet—evil 	The color preferred for men's clothing and the color (or tone) with masculinity	<ul style="list-style-type: none"> Represents rebirth and mourning Evil and mystery 	Associated with magic and the unknown in almost all cultures
White	<ul style="list-style-type: none"> Color of purity and peace Color associated with weddings Clean and sterile (represent hospital and holiness) Italy—funeral 	<ul style="list-style-type: none"> In the East, it is a color of death Used at funeral Represent misfortune, sterility, unhappiness, and mourning 	Purity and peace	<ul style="list-style-type: none"> Purity and mourning Iran—holiness, peace Egypt—symbol person's high-ranking status 	White flag is the universal symbol of truce

The screenshot shows a survey form titled "Choose your Favourite color". It includes fields for Name, Email address, Full Name, Profession (Computer Vision professional or Non-Computer vision professional), Age group (18-30, 31-40, 41-60, 61-75, 76-90, above 90), Region (Globally), and Gender (Female, Male, Prefer not to say). The main section displays three rows of color swatches: Neutral Colors (White, Grey, Brown, Beige, I Don't Like Neutral Colors), Dark/Bright Colors (Black, Dark Red, Dark Green, Orange, Dark Blue, I Don't Like Dark/Bright Colors), and Pastel Colors (Red, Light Green, Light Blue, Light Pink, Light Yellow, I Don't Like Pastel Colors).

Fig. 3 Snapshot for survey

preferences. In this section, we will provide summary and overview of key analytical points of the color preference survey. In this research, we focus on three different shades of colors, that is Neutral colors, Dark or Bright colors, and Pestal colors. Our analysis of the survey data identifies key points about the attitude of public toward the color preferences that will inform how this color preference varies according to people's gender, age, profession, etc. (Fig. 3).

3.1 Color Preferences According to Age, Gender, Profession

Large majorities of the general public connect the colors with a number of positive words and phrases. As we heard, the colors are viewed as: relaxing; soothing; vibrant; relaxing; powerful; and fun. When asked about the favorite color from the different shades that is neutral color, dark/bright color, and pastel color, close to half of the males report that they prefer dark/bright colors, whereas women give preference to pastel or say light shades colors. Not only gender affects the color preference, but also choice may vary according to professional vision. In our survey, we categorize professional vision of people into two parts, namely computer vision professional and non-computer vision professional. According to the results noted in the survey, 23.8% people belong to non-computer vision professional category. The people belonging to such category often choose dark color shades. As we discussed above, color preference may vary according to age too. The survey result also indicates that color preferences vary according to age group. Also 14.3% people having age group above 30 took part in the survey generally preferred neutral colors. Detailed findings of this survey are mentioned below (Table 4).

Looking at the age we find that younger people mostly choose dark/bright color and middle-aged people (under 60) frequently choose neutral color. They both less

Table 4 Survey result of color preferences according to age, gender, and profession

	Age		Gender		Profession	
	15–30 (%)	Above 30 (%)	Male (%)	Female (%)	Non-computer vision (%)	Computer vision (%)
Neutral color	23.5	66.6	23.8	28.5	25	25
Dark/bright color	41.1	2	42.8	23.8	50	32.3
Pastel color	35.4	33.3	28.6	47.6	25	41.2

likely considered or preferred pastel colors. 47.6% women report pastel color as their favorite color shade, but almost 42.8% men prefer dark/bright colors as their favorite color. In focus, the data shows that 50% people belonging to non-computer vision professional category express greatest preference toward dark/bright shades of color, whereas 41.2% people from computer vision category are more likely to prefer pastel colors.

4 Conclusion

In this study of color's effect on visual aesthetic sense, we have reviewed color psychology that is useful for we people to make day-to-day decisions. As per given in [24], color is a message that is received and send by human (from babyhood through old age) for their being with color vision. Color preference varies with person to person. Representing aesthetic preference is a new, but interesting area with many applications like cinematography world, social websites, revenue generation in advertisement world, marketing, fashion designing world, etc. [25]. By considering these applications, color-mood analysis is done. Possibly, the most important finding about hues as well as color combination is that people mostly preferred and considered those hues that are harmonious to their visual sense. A finding of the result shows that there are many reasons that impact on people, for choosing different colors. Perhaps, color preference changes according to the human's gender and age or may be due to the influence of culture.

References

- Elliot AJ et al (2007) Color and psychological functioning: the effect of red on performance attainment. *J Exp Psychol General* 136(1):154
- Palmer SE, Schloss KB, Sammartino J (2013) Visual aesthetics and human preference. *Annu Rev Psychol* 64:77–107
- Labrecque LI, Milne GR (2012) Exciting red and competent blue: the importance of color in marketing. *J Acad Mark Sci* 40(5):711–727

4. Kumi R et al (2013) Research article learning in color: how color and affect influence learning outcomes. *IEEE Trans Prof Commun* 56(1):2–15
5. Datta R et al (2006) Studying aesthetics in photographic images using a computational approach. In: European conference on computer vision. Springer, Berlin, Heidelberg
6. <https://www.cinema5d.com/film-color-schemes-cinematic-color-design/>
7. <https://designshack.net/articles/inspiration/how-to-use-cool-color-in-design-projects/>
8. <https://www.empower-yourself-with-color-psychology.com/meaning-of-colors.html>
9. <https://www.bfloral.com/bfloral-blog/2018/05/15/2018-3-26-how-to-effectively-incorporate-a-monochromatic-color-scheme-for-your-event>
10. Wei C-Y, Dimitrova N, Chang S-F (2004) Color-mood analysis of films based on syntactic and psychological models. In: 2004 IEEE international conference on multimedia and expo (ICME) (IEEE Cat. No. 04TH8763), vol 2. IEEE
11. Palermo F, Hays J, Efros AA (2012) Dating historical color images. In: European conference on computer vision. Springer, Berlin, Heidelberg
12. Ellis L, Ficek C (2001) Color preferences according to gender and sexual orientation. *Pers Individ Differ* 31(8):1375–1379
13. Taylor C et al (2013) Color preferences in infants and adults are different. *Psychon Bull Rev* 20(5):916–922
14. Schloss KB, Palmer SE (2009) An ecological valence theory of human color preferences. *J Vision* 9(8):358–358
15. <https://digitalsynopsis.com/design/male-vs-female-color-perceptions-preferences/>
16. Fortmann-Roe S (2013) Effects of hue, saturation, and brightness on color preference in social networks: Gender-based color preference on the social networking site Twitter. *Color Res Appl* 38(3):196–202
17. Moss G, Colman AM (2001) Choices and preferences: experiments on gender differences. *J Brand Manage* 9(2):89–98
18. <https://www.aoa.org/patients-and-public/good-vision-throughout-life/childrens-vision/infant-vision-birth-to-24-months-of-age>
19. <https://www.color-meanings.com/color-psychology-child-behavior-and-learning-through-colors/>
20. <https://www.webdesignerdepot.com/2012/06/color-and-cultural-design-considerations/>
21. Terwogt MM, Hoeksma JB (1995) Colors and emotions: preferences and combinations. *J Gen Psychol* 122(1):5–17
22. Adams FM, Osgood CE (1973) A cross-cultural study of the affective meanings of color. *J Cross Cult Psychol* 4(2):135–156
23. Bonnardel V et al (2012) Color preferences: a British/Indian comparative study. In: Colour and environment: AIC2012 conference proceedings
24. Elliot AJ, Niesta D (2008) Romantic red: red enhances men's attraction to women. *J Pers Soc Psychol* 95(5):1150
25. Ou L-C, Luo MR (2006) A colour harmony model for two-colour combinations. *Color Res Appl* 31(3):191–204 (Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur)

Chapter 20

Performance Evaluation of Video Segmentation Metrics



Shriya Patil and Krishna K. Warhade

1 Introduction

With an exponential increase in the area of multimedia technology, daily creation of a large number of digital videos, the field of indexing and retrieval is becoming an active area. There are two types of retrieval, text-based video retrieval used for keyword search whereas content-based video retrieval is used for manipulating a video into frames. Shot boundary detection, motion vectors, etc., is mostly used for video segmentation. Shot boundary detection is used to identify the discontinuity between two frames inside a video/story unit. Figure 1 shows a structure of video. Shot transition is used to split up video into frames called shots. Shot consists of several frames whereas scenes consist of several shots. A frame is considered as a complete image which is presented in rapid succession.

Shot transition is classified as abrupt transition (AT) and gradual transition (GT). Abrupt transition occurs when there is an abrupt change from one shot to another shot and also known as hard cut. Gradual transition occurs when two shots are combined using chromatic or spatial effect. A gradual transition is classified as wipe, dissolve, fade-in, and fade-out.

The existing techniques for shot boundary detection include histogram difference, pixel difference, likelihood ratio, color spaces, edge differences, etc., are surveyed and presented in the literature. A Weber feature-based method for scene change having camera and object motion has been proposed by Kar and Kanungo [1]. However, this method cannot handle scene change in dark environment.

S. Patil (✉) · K. K. Warhade

School of Electronics and Communication Engineering, Dr. Vishwanath Karad,
MIT World Peace University, Pune, India
e-mail: shriyapatil01@gmail.com

K. K. Warhade
e-mail: krishna.warhade@mitwpu.edu.in

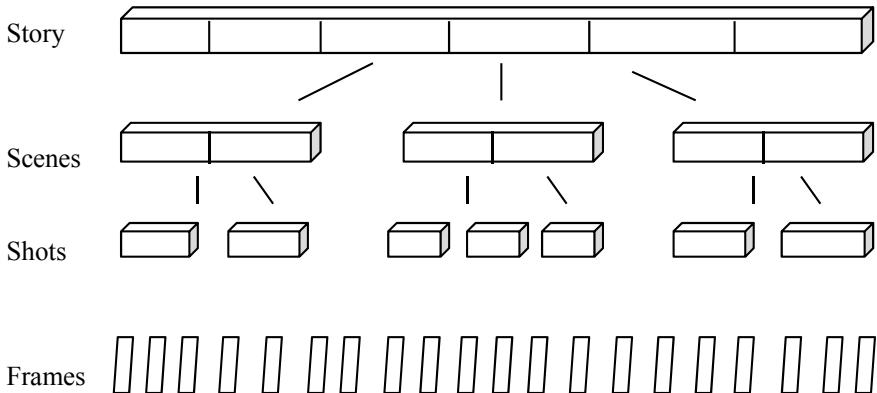


Fig. 1 Structure of video

Kavitha et al. [2] proposed wavelet-based feature vector for detection of shots with analyzing and extracting color, edge, motion, and texture feature for each video frame with neighboring left and right frame and identifies shot transition. The existing literature based on traditional metrics, i.e., likelihood ratio, histogram comparison, pair-wise comparison has been proposed by Zhang et al. [3]. A shot detection method based on gray level co-occurrence matrix has been proposed by Mounika and Khare [4]. In the proposed method, shots were detected by comparing the differences of contrast, correlation, energy, and homogeneity between two frames within a given threshold. SenGupta et al. [5] have presented different ideas for detection based on gradual transition. Some of the challenges, such as object and camera motion, fire explosion, are detected for gradual transition as it produces false cuts. Wu and Xu [6] have elaborated the theory on shot boundary and analyzed several shot boundary algorithm in abrupt transition as well as gradual transition. Gargi et al. [7] suggested that color histogram is used to detect shots of each frame equivalent to their color features.

2 Metrics for Shot Boundary Detection

We need to determine an appropriate metrics so that a transition is declared when it exceeds the given threshold. In this paper, we have evaluated the traditional metrics that are robust to illumination motion. Traditional metrics like likelihood ratio, wavelet-based method, chi-square test, and histogram difference are explained below.

2.1 Likelihood Ratio (LHR)

Likelihood Ratio is based on second-order statistics over a region computed by Jain et al. [8]. The performance of this metric can be improved by facets and quadratic surface to approximate intensity values. Higher-order allows better characterization of intensity values. Here σ_i , σ_{i+1} and μ_i , μ_{i+1} are the standard deviation and mean of intensity values f_i and f_{i+1} , respectively, where f_i is the reference frame and f_{i+1} is the next frame.

$$\text{LHR}(i) = \frac{\left[\frac{(\sigma_i + \sigma_{i+1})}{2} + \frac{(\mu_i - \mu_{i+1})^2}{2} \right]}{\sigma_i * \sigma_{i+1}} \quad (1)$$

2.2 Chi-Square Test (CS)

Nagasaki and Tanka [9] tested with pixel difference and histogram metrics and shown that histogram is more effective. They segmented video into 16 regions and found the best results using color histogram of those regions. Here, H_i and H_{i+1} represent the histogram value for i th and $(i+1)$ th frame, j is possible gray level.

$$\text{CS}(j) = \sum_{j=1}^G \frac{|H_i[j] - H_{i+1}[j]|^2}{H_{i+1}[j]} \quad (2)$$

2.3 Pixel Difference (PD)

Pixel difference is used to count the number of pixels in an image. It is very sensitive to noise and the motion. Zhang et al. [3] suggested that the effect of motion can be limited by using filtering before pixel comparison.

$$\text{PD}(i) = \frac{1}{PQ} * \sum_{x=1}^P \sum_{y=1}^Q |(f[x, y, i] - f[x, y, i + 1])| \quad (3)$$

2.4 Wavelet-Based Method (WDM)

Every frame is decomposed up to three level, and only LL part (LL3) is used to find metric.

$$\text{WDM}(k) = \sum_{i=1}^{32} \sum_{j=1}^{32} [\text{LL3}(i, j, k) - \text{LL3}(i, j, k+1)] \quad (4)$$

Here, size of image is 256 * 256.

3 Datasets and Performance Measure

The traditional metrics have been tested on the movies Pearl Harbor-I, Pearl Harbor-II, The Marine, Iron Man-III and Independence Day that have a lots of scene changes, fire explosion, and rapid camera motion. The ground truth for shot detection is manually determined using virtual-dub software. The shot detection result of metrics is computed with respect to recall, precision, and F1 measure.

Table 1 shows the dataset description used to detect shot boundary. Traditionally, the shot detection results are expressed with respect to recall and precision. It is given as

$$\text{Recall (R)} = \frac{C}{C + M} \quad (5)$$

whereas precision is given as

$$\text{Precision (P)} = \frac{C}{C + \text{FP}} \quad (6)$$

where C is the number of transition detected correctly, M is the missed transition that is manually present but not detected and FP is the false cut detected by metrics.

Table 1 Dataset

Movie name	Number of frames	Number of frames with abrupt transition	Number of frames with flashlight/fire explosion/motion
Pearl Harbor-I	1019	25	23
Pearl Harbor-II	1554	42	32
The Marine	2285	86	68
Iron Man-III	9568	245	215
Independence Day	9150	118	103

F1 measure is a measure of test's accuracy and is defined as the mean of recall and precision of the test.

$$\text{F1 measure} = \frac{2 * R * P}{R + P} \quad (7)$$

4 Experimental Results

The traditional metrics such as likelihood ratio, chi-square test, and wavelet-based method as shown in Eqs. (1), (2) and (4) have been tested on given datasets.

For example: In Pearl Harbor-I movie clip,

- Number of frames = 95
- Breaks obtained manually (ground truth) = 32, 73, 94
- Break using likelihood ratio = 21, 43, 73, 94
- Break using chi-square test = 2
- Break using wavelet-based method = 32, 43, 94.

Here actual shot boundaries are detected at 32, 73, 94th frame. From Fig. 2, it is observed that the threshold values are selected in such a way that few of the false cuts can be avoided. Here, threshold value is selected above 2 so that false positive is avoided but it results in missed shot boundaries.

Thresholding is an adaptive value which draws a boundary line below and above threshold.

$$\text{Threshold} = \mu + \alpha \times \sigma \quad (8)$$

where μ and σ are the mean and standard deviation and α is experimentally selected based on metrics. A transition is declared if sum exceeds the selected threshold. From Eqs. (1), (2), and (4), we have calculated the likelihood ratio, chi-square test, and wavelet-based method in terms of performance parameter, i.e., recall, precision, and F1 measure (Fig. 3).

Fig. 2 Wavelet-decomposition level

LL3	LH3	LH2		LH1
HL3	HH3	HL2		HH2
HL1		HH1		
HL1				HH1

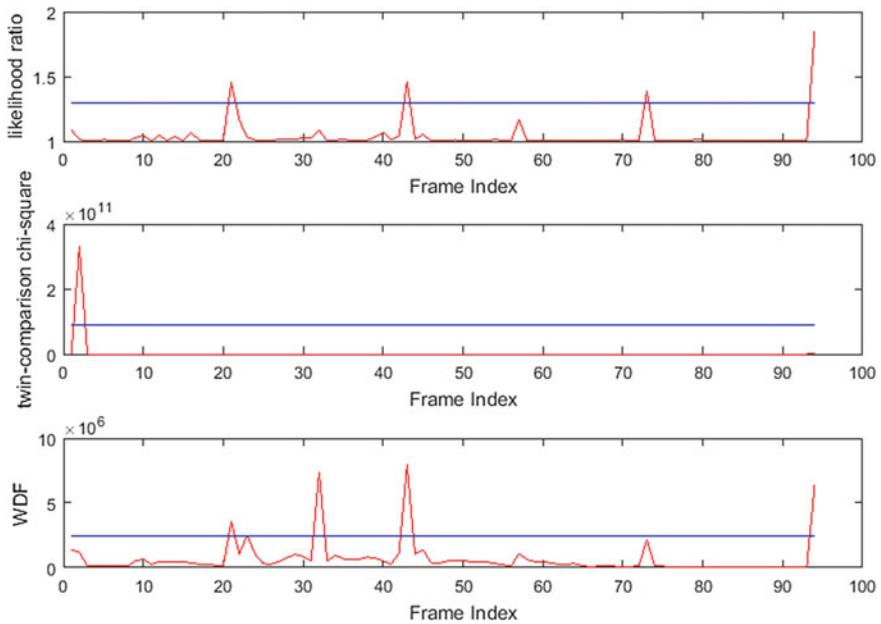


Fig. 3 Shot boundary detection using traditional metrics

The results for comparing the metrics are shown in Table 2.

Overall it has been observed that wavelet-based method shows better results as compared to likelihood ratio and chi-square test. But in case of movie 'The marine,' the results are very poor due to fire flicker and explosion and fast camera motion.

Table 2 Comparison of traditional metrics

Metric	Evaluation parameter	Pearl Harbor-I (%)	Pearl Harbor-II (%)	The Marine (%)	Iron Man-III (%)	Independence Day (%)
Likelihood ratio	R	74.7	67.07	45.23	41.78	71.25
	P	65.6	61.07	63.28	56.37	81.11
	F1	67.6	59.42	47.51	45.39	71.55
Chi-square test	R	43.3	28.14	16.45	28.60	56.57
	P	50	41.57	46.66	37.30	57.34
	F1	44.9	34.8	22.57	28.83	51.75
Wavelet-based method	R	91.2	85	54.58	55.44	84.86
	P	85.2	97.57	74.54	63.58	92.07
	F1	86	89.35	59.81	47.94	84.5

5 Conclusion

Shot boundary detection under flashlight, camera, and object motion is considered to be a challenging than regular scene change. In this paper, we have evaluated the different metrics proposed by researchers which produce false positives and missed shots. It has been observed that wavelet-based method shows comparatively better results and other metrics give poor results caused by camera and fire flicker. An algorithm to remove the false cuts and missed cuts from a scene caused by camera motion, fast object motion, flashlight, and fire flicker is required for proper video segmentation.

References

1. Kar T, Kanungo P (2018) Motion and illumination defiant cut detection based on weber features. *IET Image Proc* 12(1):1903–1912
2. Kavitha J, Jansi Rani PA, Sowmyayani S (2017) Wavelet-based feature vector for shot boundary detection. *Int J Image Graphics* 17(1):1750002
3. Zhang HJ, Kankanhalli A, Smoliar S (1993) Automatic partitioning of full-motion video. *Multimedia Syst*. 1(1):10–28
4. Mounika BR, Khare A (2017) Shot boundary detection using second order statistics of gray level co-occurrence matrix. *Res J Comput Inf Technol Sci* 5:1–7
5. SenGupta A, Singh KM, Roy S (2015) Video shot boundary detection: a review
6. Wu Z, Xu P (2013) Shot boundary detection in video retrieval. In: IEEE 4th international conference on information and emerging communication
7. Gargi U, Kasturi R, Strayer S (2000) Performance characterization of video-shot-change detection methods. *IEEE Trans Circ Syst Video Technol* 10(1):1–13
8. Jain R, Kasturi R, Schunck B (1995) Machine vision. McGraw-Hill, New York, pp 406–415
9. Nagasaka A, Tanka Y (1992) Automatic video indexing and full video search for object appearance. In: Visual database systems II. Elsevier Science Publishers, pp 113–127

Chapter 21

Suspicious Activity Detection Using Live Video Analysis



**Asmita Gorave, Srinibas Misra, Omkar Padir, Anirudha Patil
and Kshitij Ladole**

1 Introduction

A video surveillance system can be generally described as a system which enables us to get a constant view of multiple locations within the same area using multiple cameras. The surveillance system is primarily utilized for the detection of unusual and abnormal activities. Activity detection is a very crucial component of a video surveillance system. However, of the majority of current video surveillance systems in place, the task of activity detection and recognition is left for humans, i.e., personnel are appointed whose job is to constantly monitor the systems and check whether any suspicious activity is occurring. Along with this, many organizations do not have the budget to employ a security staff or a surveillance team to constantly monitor the video streams. Thus, in their case the surveillance footage is analyzed after a crime has occurred.

Our paper proposes a system which performs the task of activity detection and suspicious activity recognition in real time and simultaneously notifies and updates the concerned authorities. It automates the process of analyzing the video stream, thus not requiring a set of personnel to constantly monitor the camera streams. It

A. Gorave · S. Misra · O. Padir · A. Patil (✉) · K. Ladole

Department of Computer Engineering, MIT College of Engineering, Pune, India

e-mail: anirudhapatil1998@gmail.com

A. Gorave

e-mail: asmita.gorave@mitcoe.edu.in

S. Misra

e-mail: srinibas.misra97@gmail.com

O. Padir

e-mail: omkar.padir@gmail.com

K. Ladole

e-mail: kshitijladole@gmail.com

uses computer vision techniques to perform object detection, person detection, and activity analysis.

The environment of our system has been currently set to a shop. The algorithm in use can be extended for a large convenience store or a large shop. The expert video-surveillance increases the efficiency of suspicious activity detection in comparison with traditional methods, allowing human operators to manage a higher number of cameras and their corresponding risk situations.

The rest of the paper is organized as follows: Section 2 describes the related work. Section 3 describes the proposed approach for suspicious activity detection. Section 4 describes the results and findings of our system. Section 5 describes the conclusions found from our project.

2 Related Work

A general method for analysis of human behavior (suspicious or normal) in videos involves steps such as motion detection with the help of background modeling and foreground segmentation, object classification, motion tracking, and activity recognition [1].

For human identification or person detection from a surveillance video [2], earlier systems have utilized several different methods. MID-based foreground segmentation and HOG-based head and shoulder detection have been used to detect humans. Though this method provides an accurate detection of humans, it has a shortcoming. When the number of humans is greater in the observed area, this method fails to detect the humans efficiently, i.e., the method fails when the crowd density is high.

Current detection systems utilize classifiers to perform the detection. To detect an object, they use the classifier of that particular object at different locations and scales in a test image. A sliding window approach is used in systems like deformable parts model (DPM) [3], where the classifier is run at evenly spaced locations over the entire image.

Even more recent approaches utilize deep neural networks and convolutional neural networks such as R-CNN [4] to go region by region and generate potential bounding boxes in an image and then run a classifier on these bounding boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and re-score and re-evaluate the boxes based on the other objects in the scene. This method is slow and extremely difficult to optimize because each individual component must be trained individually.

Viola and Jones introduced the Haar features approach [5]. This approach can be used for performing face detection. The main advantage of this method is that it has a low false positive rate, i.e., the probability of identifying non-face regions as face regions is very low. It has a very high accuracy in detection of faces and, when coupled with a classification algorithm such as AdaBoost, it gives the best performance and hence has an extra overhead attached with it.

The task of performing human face recognition is very challenging. The variations of facial expressions, personal appearances, the different poses, and different illumination add to the complexity of face recognition. In addition to these factors, the variability of lighting density and direction, the number of light sources in the scene and the orientation of the camera add to the existing complexity and challenges of performing face recognition in a real-time scenario. The object recognition system requires an enormous volume for the computing process, and thus, the classification processing time can be reduced by reducing the image dimensions. LBPH [6] is one of the most popular traditional and conventional methods. It is used for robust data representations, as well as histograms, for feature reduction.

There are various techniques for detection of suspicious behavior of moving people within the cameras field of view. The techniques utilize computing the motion vector of the moving object as presented in [7]. The displacement vector is represented as the mean motion vector of the object's interest points. Scale Invariant Feature Transform (SIFT) [8] descriptor can be chosen for object recognition between consecutive frames. It provides high performance, and suitability for video vision application with a slow acquisition video frames where the displacement vector is recomputed for each received frame in the video sequence.

3 Classification Models

3.1 YOLOv3 (*You Only Look Once*)

The main detection model utilized is YOLO (You Only Look Once) for performing all the classification and detection of objects and humans. The YOLO model utilizes a single network and performs a single evaluation to predict the bounding boxes and class probabilities. The simplicity of YOLO algorithm provides real-time speed in comparison with other neural network utilizing algorithms such as R-CNN and Fast R-CNN.

The steps of YOLO are:

1. It divides the image into an $S \times S$ grid.
2. Each cell of the grid predicts B bounding boxes with a confidence score. The confidence is basically the probability to detect the object multiplied by the IoU (Intersection over Union = Area of Overlap/Area of Union) between the predicted and the ground truth boxes (Fig. 1).

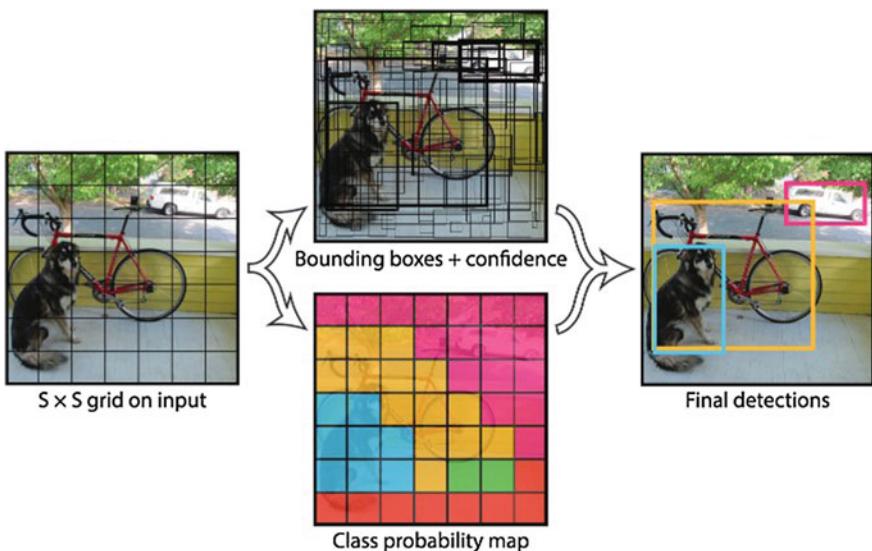


Fig. 1 YOLO prediction model

3.2 LBPH (Local Binary Pattern Histogram)

Along with YOLO, another classifier used is the LBPH classifier. LBPH classifier is used at scenarios where there needs to be an implementation of dynamic classification, that is, a person detected needs to be added to a classifier model so that it can be recognized immediately. And this entire process occurs over 15–30 frames. Local binary pattern (LBP) is a simple algorithm which utilizes a texture operator which is used for image classification. A normal training dataset is passed to it along with the labels for the different objects present in the dataset (Fig. 2).

The steps for enhancing are:

1. First selects 3×3 grid of pixels..
2. Sets the value of the middle pixel of the grid as the threshold value.

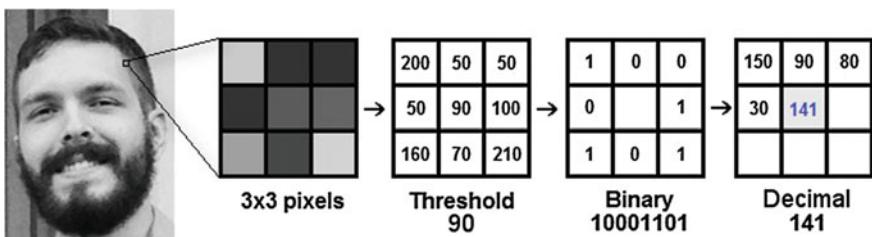


Fig. 2 Working of LBPH

3. Converts the 3×3 grid of pixels into corresponding binary values. This is done by comparing all the values of the pixels in the grid with the threshold value. If the value is greater than the threshold value, then it is set as 1, otherwise it is set as 0. Thereby creating a binary pixel grid.
4. After this, the binary number is converted to its equivalent decimal number, and the value of the middle pixel is set that decimal number.
5. This process is done for every pixel that can be a middle pixel in a grid.

4 Proposed System

The scenario for which the project aims to develop an algorithm for is set in a small shop containing an entrance, a counter where the objects are placed, and a billing counter where the objects can be bought.

There are three cameras present in the shop, and they are arranged in the following way:

1. Camera 1 is located at the front gate, looking outward as the customers enter. The camera is used to recognize and add the customers into the database, and also it is used to check if any object is taken out of the shop at any instance.
2. Camera 2 is located at the counter where the objects are placed for the customers to see. The camera is pointed at the customers to get a clear view of the customers interacting with the objects present. The use of the camera 2 is to identify which person is interacting with the objects and to identify which person is responsible if an object goes missing.
3. Camera 3 is located at the billing counter, again pointed toward the customers. This camera is used to identify which customer is intending to buy the object and which person has some other malicious intent.

4.1 Algorithm Prerequisites

1. **MIN_MOVEMENT_RANGE**: A range of movement which is allowed.
2. **MIN_OBJECT_MISSING**: A preset value, which when crossed, the object would be termed as missing.
3. **MIN_PERSON_OBJECT_MISSING**: A preset value, which when crossed, would mean that the person and the object are missing. And the person would be classed as suspicious.
4. **MIN_OBJECT_RETURN**: A preset value, which when crossed, would mean that the object has been returned to its location.
5. **MIN_PERSON_OBJECT_RETURN**: A preset value, which when crossed, would mean that the person has returned the object back to its assigned location.

`MIN_PERSON_OBJECT_MISSING` should be greater than `MIN_OBJECT_MISSING`, because the object should be classed as missing before the person is classed as suspicious. Similarly, `MIN_OBJECT_RETURN` should be greater than `MIN_PERSON_OBJECT_RETURN`, because the object should be classed as not missing before the person is classed as not suspicious.

4.2 Algorithm Steps

To Detect Shoplifting

General working algorithm for shoplifting (at camera 2):

1. Calculate the centroids of the person interacting with the object and the object itself. And also track the movement of the object by calculating the distance between the current centroid and the original location centroid.
2. If the object is moved out of the `MIN_MOVEMENT_RANGE`, then:
 - (a) Send a message saying unusual movement detected.
 - (b) Store frames where unusual movement has been detected.
 - (c) If the object is not detected, count the number of frames where it is missing consecutively. If the frame count exceeds the `MIN_OBJECT_MISSING`:
 - (i) Label the object as missing.
 - (ii) Label the person as suspicious.
 - (iii) Send a message saying that the object is missing and the person is suspicious.
 - (iv) Store the frames labeled as suspicious activity.
3. If the object is detected again:
 - (a) Count the number of consecutive frames where the object is detected
 - (i) If the frame count exceeds `MIN_OBJECT_RETURN`, then label the object as not missing and the person as not suspicious.
 - (ii) Send a message saying that the object is found.
4. If the person and the object are not detected, start counting the frames where the person and object are missing, or the object is missing. If the count exceeds `MIN_OBJECT_MISSING` or `MIN_PERSON_OBJECT_MISSING`:
 - (a) If the count exceeds `MIN_OBJECT_MISSING`:
 - (i) Label the object as missing.
 - (ii) Label the person as suspicious.
 - (iii) Send a message saying that the object is missing and the person is suspicious.
 - (iv) Store the frames labeled as suspicious activity.
 - (b) If the count exceeds `MIN_PERSON_OBJECT_MISSING`:
 - (i) Label the object as missing.

- (ii) Label the person as suspicious.
 - (iii) Send a message saying that the object is missing, and the person is suspicious.
5. If object is detected again:
 - (a) Count the number of consecutive frames where object is detected.
 - (b) If the frame count exceeds MIN_OBJECT_RETURN, label the object back to not missing.
 - (c) Send a message saying that the object is found.
 - (d) If person marked as criminal, change it to suspicious.
 6. If person and object are detected again:
 - (a) Count the number of consecutive frames where missing object and person are detected.
 - (b) If the frame count exceeds MIN_PERSON_OBJECT_RETURN, then label the object as not missing and the person as not suspicious.
 - (c) Send a message saying that the object and the person are found.
 - (d) If person marked as criminal or suspicious, change to normal.

At camera 1:

1. Detect and recognize the person and the object.
2. If the person is classified as suspicious:
 - (a) Send a message saying that the suspected person has been spotted at camera 1.
 - (b) Mark the person as criminal.
 - (c) Store frames of the person at camera 1.
3. If the object is classified as missing:
 - (a) Send a message saying that the missing object has been spotted at camera 1.
 - (b) Store the frames of person at camera 1.

At camera 3:

1. Detect the person and the object.
2. If person detected is suspicious and the missing object is not detected:
 - (a) Send a message saying that the suspected person has been spotted at camera 1.
 - (b) Mark the person as criminal.
 - (c) Store frames of the person at camera 3.
 - (d) If the person detected is suspicious and the missing object is also detected:
 - (i) Send a message saying that the suspicious person is spotted at camera 3.
 - (ii) Mark the object as not missing and the person as not suspicious.

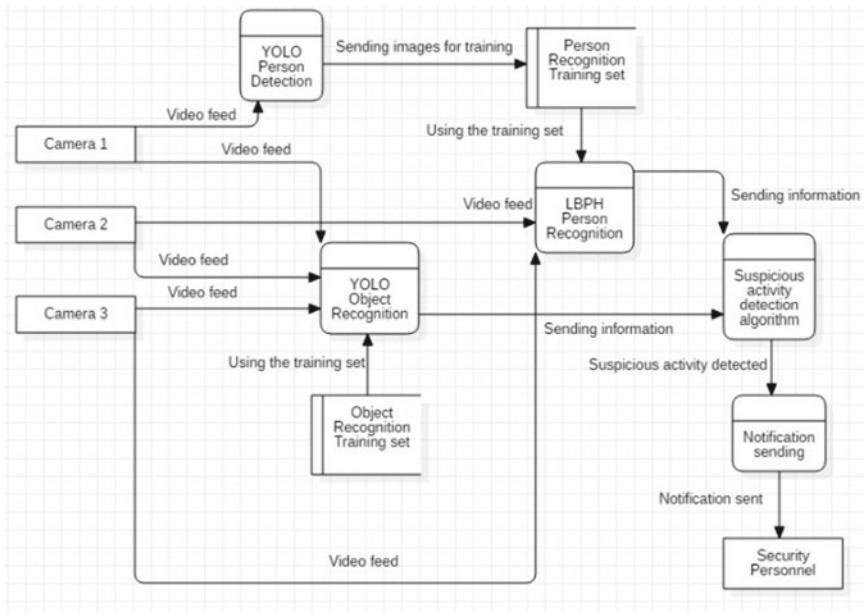


Fig. 3 Block diagram

To Detect Violent Activity (Pulling out a weapon) To detect violent activity, in the use case of pulling out a weapon such as a knife, or a gun, is relatively simple. Classifier models would be in place which would be able to detect gun or any such weapon in the frame. All the three cameras would constantly check if they detect a weapon of any sorts, as soon as the weapon is detected, the suspicious activity alarm would be raised (Fig. 3).

4.3 Actual Working Using YOLOv3 and LBPH for Person Identification at Camera 1

1. First task would be to detect a person. All the three cameras are running a YOLOv3 classifier model, which is capable of detecting individual people, and provide the coordinates of the detected person.
2. As soon as a person is detected, a unique ID is provided to the person. And consecutive 15 frames of the person (only the person) are stored to form a training set, so that we can identify and recognize the person later on.
3. The training set is used to develop a LBPH classifier. A YOLO classifier is not developed for person identification because training a YOLO classifier is very re-source intensive.

4. This classifier is used to recognize the person at all the three cameras. And this entire process is redone whenever a new person is detected.
5. For person identification, the YOLO classifier would be able to detect the person in the frame, also providing a small rectangle around the person. This rectangular section of the frame can be passed to the LBPH classifier for further classification, thus providing with an ID of the person.
6. If the ID is not found, then this person has not been added to the database yet, and is new. Thus, the training set is created and the process repeats.

For Object Identification at Cameras 1, 2, and 3

1. For object identification, a custom YOLO classifier model needs to be created. This is done beforehand.
2. A training set of images for each individual objects are created, and these are used to train a YOLO classifier. This is done beforehand, as the process of training a custom classifier using YOLO is very resource intensive.
3. This custom model along with the normal straight YOLO classifier model would be used to classify objects as well as people in the frame.

For Weapon Identification at Cameras 1, 2, and 3

1. For identifying weapons, the same process of object recognition using YOLO needs to be done.
2. A training set of weapons is created which is used to train the YOLO classifier. And then, this classifier is used to detect weapons at the three cameras.

5 Experimental Results and Analysis

The proposed suspicious activity detection system is evaluated on real time for validation of framework's procedural tasks. The format of live video feed is 940*640 pixels. The implementation and testing of a system were conducted in a well-lit environment with testing objects, tables, and other entities.

The workflow of the system is as depicted in the following figures.

5.1 Camera 1

Figure 4a depicts that a new person enters in the shop and comes in field of view of camera 1. In the next figure, the person is detected and assigned a particular ID. Multiple images of the person are stored to generate the training set for further recognition.

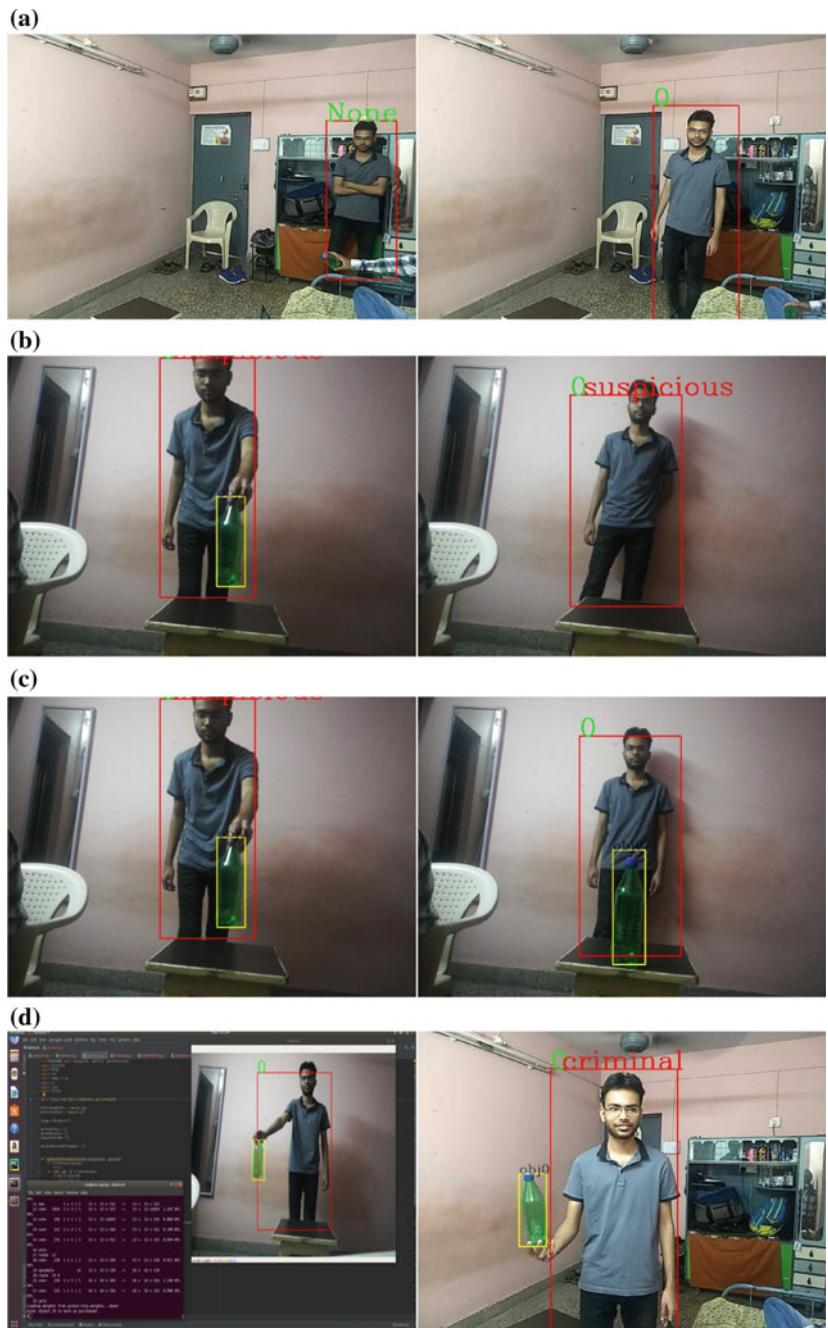


Fig. 4 **a** Entry of customer and assigning ID to him. **b** Person interacting with the object and marking him as suspicious when the object goes missing. **c** Person returning the object and marked safe again. **d** Person with the object at the billing desk and if he exits without billing

Table 1 Confusion matrix

	Predicted positives	Predicted negatives
Actual positives	TP = 45	FN = 15
Actual negatives	FP = 22	TN = 8

5.2 Camera 2

Figure 4b shows the interaction between the recognized person and an object. In the next figure, it can be seen that object has gone missing and, hence, the person is marked as suspicious. In Fig. 4c, the person has then placed the object again at its original location. Consequently, he is marked safe again.

5.3 Camera 3

In Fig. 4d, person has bought the object along with him at the billing desk with the intention of purchasing. If the customer has purchased the object, then the object along with all its interactions is removed from the database and the person is marked safe. If the person tries to escape without billing, he is marked as suspicious (Table 1).

The system is evaluated by precision and recall metrics. It includes true positives, true negatives, false positives, false negatives count, and precision and recall are counted.

Precision and recall are calculated as follows:

$$\text{Precision} = \text{TP} \div (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} \div (\text{TP} + \text{FN})$$

The value of precision is 0.750 and that of recall is 0.673.

6 Conclusion and Future Scope

This paper proposes a possible system for suspicious activity detection to improving the functionality of the current surveillance cameras to convert them to an intelligent device capable of detecting suspicious activities. The system is also capable of notifying the appropriate authorities about the suspicious activities occurring. The most important aspect about the proposed system is that it works real-time. It takes in live camera streams from the multiple cameras, performs activity detection, and calculates whether it is suspicious or not. And this occurs in real time, thus providing live and real-time updates of suspicious or criminal activities occurring in the specific area.

Initially, a customer dataset is created for person recognition and then the interactions between the object and the person are captured. Based on the nature of the interactions, the person is classified as safe, suspicious, or criminal. The algorithm works efficiently with an accuracy of 74% when experimented against real-time video feed captured through multiple cameras.

Our goal is to improve the system by improving our detection and notifying system by adding various other features such as expanding the dataset, experimenting in different locations. We aim to make our system scalable and improve efficiency in crowded places. Also, generalization for various stores is utmost importance. There is room to increase the accuracy of the system to reduce the false positives and false negatives and to investigate the category of activity and level of impact.

References

1. Sakaino H (2013) Video based tracking, learning and recognition method for multiple moving objects. *IEEE Trans Circ Syst Video Technol* 23(10):1661–1674
2. Gowsikha D, Abirami S (2012) Suspicious human activity detection from surveillance videos. *Int J Internet Distribut Comput Syst* 2(2)
3. Arroyo R, Yebes JJ, Bergasa LM (2015) Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls. *Expert Syst Appl* 42(21). Elsevier
4. Hariyono J, Jo KH (2016) Centroid based pose ratio for pedestrian action recognition. In: IEEE 25th International symposium industrial electronics (ISIE), pp 895–900
5. Pavithradevi K, Aruljothi S (2014) Detection of suspicious activities in public areas using staged matching technique. *Int J Adv Inf Comm Tech (IJAICT)* 1(1):140–144
6. Nguyen V-T, Le T-L, Tran T-H, Mullot R, Courboulay V (2014) Hand posture recognition using Kernel descriptor. In: 6th International conference on intelligent human computer interaction, IHCI 2014. Procedia Comput Sci 39:54–157. Elsevier Publications
7. Farooq J, Ali MB (2014) Real time hand gesture recognition for computer interaction. In: 2014 International conference on robotics and emerging allied technologies in engineering (iCREATE), IEEE Conf Publications, pp 73–77
8. Albukhary N, Mustafah YM (2017) Real time human activity recognition. In: 6th International conference on mechatronics, ICOM'17

Chapter 22

A Review on Using Dental Images as a Screening Tool for Osteoporosis



Insha Majeed Wani and Sakshi Arora

1 Introduction

Osteoporosis is the skeletal disease associated with the mineral content of the bones. Onset of osteoporosis can be identified by the decrease in the mineral density of bones results in porous bones leading to fractures, morbidity, mortality and increasing burden of cost [1]. Mineral density (BMD) decreases mostly after the age of 30 because the bone formation process decreases [2]. And, it is more common in women than men because of the low oestrogen level in women after the menopause [3, 4]. BMD is measured with the help of dual energy X-ray absorptiometry (gold standard) given by WHO by calculating the T-score (i.e. standard deviations around the mean BMD which is at or below 2.5 for osteoporosis) values of the different sites [2, 5]. Tooth is mainly composed of calcium as that of the bones of humans. With the decrease in bone mineral density, the tooth also is affected and undergoes various changes resulting in the decrease of minerals in tooth.

Detection of diseases with the help of medical images has led to the revolution in medical field. CAD systems based on image processing are less expensive and can be made readily available leading to the less economic burden and large accessibility to the screening of diseases. In case of osteoporosis, also these systems can be used as the early detection tools. Osteoporosis is detected using DEXA images [6], QCT [7], QUS [8], etc., but their limitations like areal measurements, limited access to whole body scanning, absence of strong empirical evidence, respectively, and less availability, high radiation exposure and high cost in common make them not so eminent. So, many CAD systems have been developed using medical images like

I. M. Wani · S. Arora (✉)

School of Computer Science Engineering, Shri Mata Vaishno Devi University,
Katra, Jammu and Kashmir, India
e-mail: sakshi@smvdu.ac.in

I. M. Wani
e-mail: insha333@gmail.com

Table 1 BMD (bone mineral density) of different skeletal sites

Bone site	Mean BMD	SD	Minimum BMD	Maximum BMD
Mandibular body	1.105	0.296	0.414	1.747
Mandibular ramus	0.708	0.188	0.254	1.187
Mandibular symphysis	1.658	0.43	0.8	2.492
Distal forearm	0.286	0.062	0.168	0.431
Proximal forearm	0.393	0.072	0.25	0.559
Lumbar vertebrae	1.058	0.169	0.728	1.415
Femoral neck	0.831	0.144	0.614	1.17

SD standard deviation

All BMD values are in g cm⁻²

X-ray and MRI of different body sites like hip, hand, vertebrae, pelvis, knee and tooth. In this paper, we will review different CAD systems based on tooth images and will show that tooth images can be used as less expensive and easily available screening tool for early diagnosis of osteoporosis.

Study to assess the bone density in dental radiographs could be dated back to the 1960s. Many techniques had been investigated like radiometric classification using grey level histograms [9], pixel intensity [10], subtraction radiography [11, 12], greyscale analysis [13], micro-densitometry [14, 15], fractal dimension analysis [16] and cortical bone thickness [17, 18], etc. For using tooth for detection of osteoporosis, the bone mass in the jaws should relate to other skeletal sites where the osteoporosis occurs. Horner et al. in 1996 [19] showed that the DXA measurements of mandible have significant relation with other sites where DXA measurements for detection osteoporosis are taken. The BMD measurements taken in their study are shown in Table 1. They showed through various evaluation methods like Pearson correlation coefficient method, sensitivity and specificity that mandible can be used to assess the risk of osteoporosis.

Southard et al. [20] used dental radiographs of alveolar bones to detect the osteoporosis. They compared the texture features and found out that the fractal dimension, mean intensity, Law's texture energy measures and gradient performed better in identifying the osteoporosis. Texture features like fractal dimension (FD), micro-densitometry, pixel intensity (PI) and panoramic analysis of cortical thickness were evaluated by Law et al. [21] from dental radiographs for early signs of osteoporosis. But the problem was that the images available were not of good quality due to which the features extracted were not clear. In our survey, we have covered the main researches of 2000s as of now, the image processing technology has improved, and also medical image database has grown many a fold and is portable and of high quality.

2 Osteoporosis and Dental Images

In this section, we present the review of the major researches about the detection of osteoporosis from tooth images. The key objective is to recognize the strengths and limitations of using tooth images as cheap and readily available alternative tool for early diagnosis of osteoporosis.

Bollen et al. [21] conducted the case control study on the mandibular cortical bone and self-reported osteoporotic fractures with the help of dental panoramic radiographs (DPR) of the elderly patients. They showed that mandible's radiographic appearance is related to the osteoporotic fractures. They divided the subjects into two groups: One is the control group who had no previous fractures and their appearance of mandibular cortex was not eroded or porous. Second group was the group with previous osteoporotic fractures and their mandibular cortex appearance on radiographs was eroded or porous. They found that the subjects with porous or eroded mandibular cortex reported twofold to eightfold odds for osteoporotic fractures.

For postmenopausal women, DPRs were shown to be a gauge of bone turnover and spinal bone mineral density in [22]. Mandibular cortical measurements like degree of mandibular cortical erosion and mandibular cortical width (MCW) were taken from mandibular inferior cortical shape. Inferior cortical shape of mandible could be classified into three phases: normal, mildly to moderately eroded and severely eroded as shown in Fig. 1. This system showed that the findings of mandibular cortical erosion could identify low BMD of spine up to 83% times correctly. And

Fig. 1 Classification of mandibular inferior cortical shape

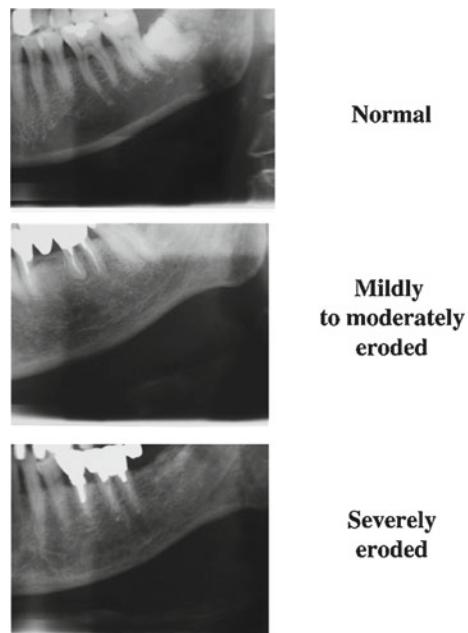
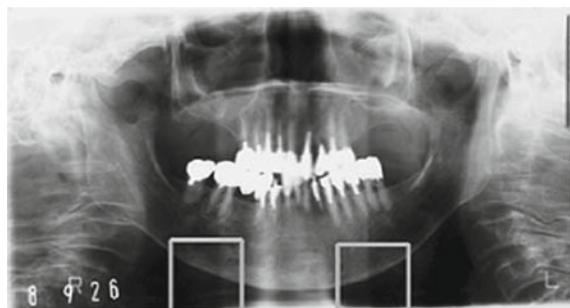


Fig. 2 Two boxes correspond to the area below the mental foramen on the left and the right sides of the mandible on digitized dental panoramic radiographs



normal dental radiograph identified the women with normal BMD 60% of the time. This suggests that dentists could identify the signs of osteoporosis in postmenopausal women.

Panoramic radiographs were also used by Klemetti et al. [23] for assessment of the osteoporotic fluctuations in the cortical area of mandible. They named the index as Klemetti index used for taking measurements. Different classifications for mandible cortical appearance described were even and sharp endosteal margin of cortex, endosteal cortical residues could be seen or semilunar defects present in endosteal margin and the cortical layer with clear pores and forming heavy residues of endosteal cortex.

Arifin et al. [24] developed a computer-aided system (CAD) for measuring cortical width from DPRs and identified the postmenopausal women with low BMD. They determined mental foramen manually and used computer-aided system for measuring the MCW below the mental foramen in which the area of interest was identified as shown in Fig. 2, image was enhanced, inner and outer cortex margins were determined, and then the appropriate point was selected where width could be measured. These measurements were compared with the BMD at femoral neck and lumbar spine, and a good correlation was found indicating that cortical width is useful for identifying the women with the risk of osteoporosis.

Allen et al. [25] suggested the automatic model, i.e. active shape model [26] for measuring the width of the inferior mandibular cortex (IMC) from dental radiographic tomograms. As we have seen that width measurements of IMC are correlated with BMD, they showed that the correlation is highest with the BMD of hip and spine in the lateral region of the mandible. Receiver operator characteristic (ROC) with area under curve (AUC) for osteoporosis analysis showed the value of 0.71 compared to AUC of 0.66 with manual measurements.

Arifin et al. in 2007 [27] built a computer-aided system by combining the novel fuzzy thresholding with fuzzy inference system fused through multilayer neural network perceptron for identifying the cortical width and shape from the DPRs of postmenopausal women. They compared the BMDs of spine and femur and were able to identify the osteoporotic women with sensitivity and specificity of 94.4% and 64.0% at spine and 90.09% and 64.7% at femur, respectively.

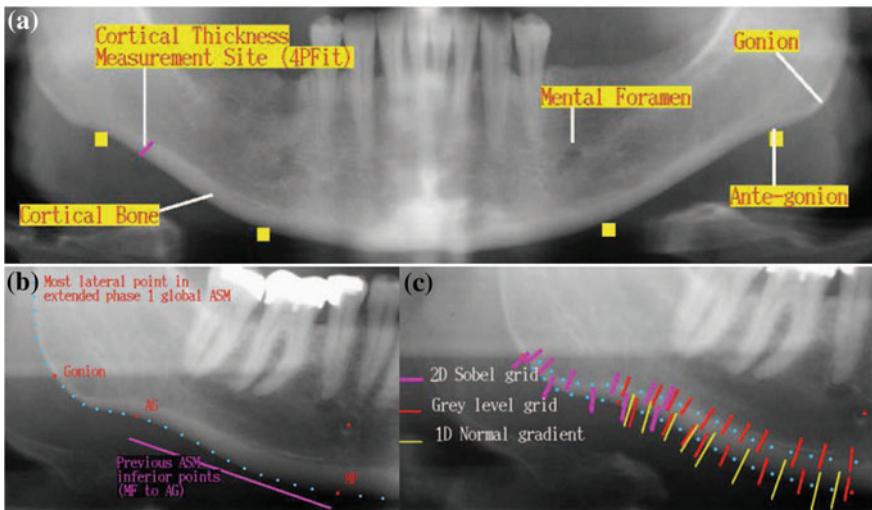


Fig. 3 **a** Typical dental panoramic tomogram, with labelled anatomical points. The rectangles indicate the lateral position of the four point initialization. **b** The phase 1. ASM shape extent (patient right side). **c** The regions of the AAM texture model sample types are indicated

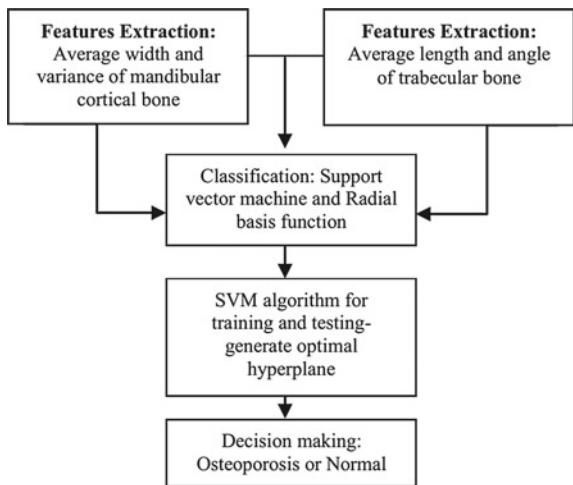
Herumurti et al. [28] used the mandibular trabecular patterns analysed from the dental panoramic radiographs to analyse the women having low BMD resulting in osteoporosis. They used computer-aided diagnostic system build using weighted fuzzy ARTMAP [29] and got the accuracy of 87.88%.

An improved model for automatically measuring the inferior mandibular cortex proposed in [25] was given by Roberts et al. [30] using an active appearance model (AAM) to overcome the limitation of poorly defining the superior border of mandible. This study gave the superior accuracy than that obtained in [25]. Figure 3 shows the dental panoramic tomogram and incorporation of ASM and AAM to locate the edges of mandible.

Another CAD system was proposed by Kavitha et al. [31] to automatically measure the mandibular cortical width (MCW) from DPRs. They used image processing paradigm in their CAD system. The X-ray images were first enhanced, cortical boundaries were determined, and then finally all the distances between the upper and lower mandibles were evaluated to calculate the width. In postmenopausal women for evaluating the correlation between the BMD of lumbar spine and that of femoral neck and the mean mandibular cortical width calculated with CAD system, Pearson's correlation coefficient was used. And it showed a good correlation between them.

With the help of support vector machine (SVM) classifier, Kavitha et al. [32] showed that the cortical measurements are more appropriate for screening of subjects with low BMD than the trabecular measurements. This was a two-class classifier with one-class predicting subjects with osteoporosis and other class predicting the normal subjects. For cortical bone, average width (AW) and variance were measured, and for trabecular bone, distribution of average length (AL) and angle of trabecular bone

Fig. 4 Flowchart of SVM classifier



(ATB) segments on DPRs were measured. Flowchart used in SVM classifier is shown in Fig. 4.

In [33], Kavitha et al. gave improved CAD system combining SVM with the histogram-based clustering algorithm for diagnosing osteoporosis by analysing the dental panoramic radiographs and showed the improved accuracy of detecting low BMD or osteoporotic postmenopausal women.

Correlation between dental panoramic and micro-CT images to find mineral density of bones was found with the help of image analysis in [34] for Indonesia. And, they showed that the dental panoramic radiographs have the potential to act as screening means for the detection of osteoporosis.

As we have seen in the above literatures, the DPRs are strongly correlated with BMD calculations. Roberts et al. in [35] showed that this correlation can be improved by calculating the image texture features. They measured the texture features on the basis of fractal dimension and co-occurrence matrix in cortex as well as in the superior basal above the cortex. They used random forest classifier [36] to identify the femoral neck osteoporosis, total hip and lumbar spine.

Association between texture features and MCW was also evaluated by [37] for screening of the osteoporosis. K-nearest neighbour, Naïve Bayes and SVM classifiers were used to describe the relations and were found that for femoral neck BMD the accuracy was high with individual with features of MCW and FD combined and was low for lumbar spine. Better values were found for the combination of texture features (GLCM and FD) and MCW at the lumbar spine and femoral BMD. So, the combination of texture features and MCW shows better results.

Hybrid genetic swarm fuzzy (GSF) classifier was prosed in 2016 [38] for automatic diagnosis of osteoporosis from digital DPRs. The input attributes were portioned to form member functions (MF) and a rule set (RS). Fuzzy inference system was used for classification purpose, and genetic swarm algorithm was used for optimizing the member functions and a rule set. The evaluation of the GSF in identifying

Fig. 5 Measurement sites on mandible. CRB probe in position at parasymphysis site



the osteoporosis or low BMD was done for the lumbar spine site and the femoral neck site. The diagnostic performance of the system was 0.986 at femoral neck BMD.

Beattie et al. [39] proposed the use of quantitative ultrasound (QUS) of mandible as a diagnostic tool for osteoporosis. The axial transmission quantitative ultrasound device was used. Speed of the sound (SOS) measurements are the basis of this study and were taken for mandible. Figure 5 shows the device by which SOS measurements of parasymphysis site of mandible were taken.

BMD of hip and SOS of mandible showed a moderate correlation showing that osteoporosis could be diagnosed by such means this investigation on easily accessible skeletal site with no exposure to harmful X-ray radiations. It could be used for primary care by dentists for identifying early signs of osteoporosis.

In computer-aided diagnostic system based on deep convolutional neural network, DCNN [40] was evaluated to diagnose the osteoporosis with the help of DPRs [41]. Comparison was made with the diagnosis made by maxillofacial and oral radiologists having experience greater than ten years. The diagnosis was based on the erosion in cortical of the mandibular inferior cortex. Three systems were used as follows: SC-DCNN that is single-column DCNN, SC-DCNN Augment that is single-column with data augmentation DCNN, and MC-DCNN that is multicolumn DCNN. The values for area under the curve (AUC) were 0.9763, 0.9991 and 0.9987 for SC-DCNN, SC-DCNN (Augment) and MC-DCNN, respectively. These systems showed higher accuracy and better performance than the experienced radiologist and can even extract the features which could have been missed by human observer.

3 Conclusion

From Table 2, we noticed that dental panoramic radiographs are mostly used than those of micro-CT images of QUS by the researchers in the detection of osteoporosis. Secondly, for the region of interest, cortical bone has shown good results than trabecular bone. Thirdly, among the features, mandibular cortical width is seen to be more suitable for diagnosing osteoporosis from dental radiographs.

From the literature review, we can conclude that the tooth radiographs can be used as a diagnostic tool for the early detection of osteoporosis. And dental panoramas

Table 2 Image types, ROIs and features used in the different literature

S. no.	Author	Image type	ROI	Features	Reference no./year
1	Bollen et al.	DPRs	Mandibular cortex	Appearance: eroded or normal	21/2000
2	Taguchi et al.	DPRs	Mandibular cortex	Cortical erosion and MCW	22/2003
3	Klemetti et al.	DPRs	Mandibular cortex	Cortical residues	23/1994
4	Arifin et al.	DPRs	Cortical bone	MCW	24/2006
5	Allen et al.	DRTs	Cortical bone	MCW	26/2007
6	Arifin et al.	DPRs	Cortical bone	Cortical width and shape	27/2007
7	Herumurti et al.	DPRs	Mandibular trabeculae	MTP	29/2010
8	Roberts et al.	PDTs	Cortical bone	MCW	30/2010
9	Kavitha et al.	DPRs	Cortical bone	MCW	31/2012
10	Kavitha et al.	DPRs	Cortical and trabecular bone	AW, variance, AL and ATB	32/2012
11	Prafiadi et al.	DPRs arid micro-CT	–	Image features	34/2013
12	Roberts et al.	DPRs	Cortex and superior basal	Texture features	36/2013
13	Kavitha et al.	Digital DPRs	Mandibular cortex	Member fns and rule set	38/2016
14	Beatti et al.	QUS	Mandibular cortex	SOS	39/2018
15	Lee et al.	DPRs	Mandibular inferior cortex	Cortical erosion and MCW	41/2019

with MCW as main feature could best be suited for the purpose. And computer-aided diagnostic systems built in the last decade have shown better efficiency for osteoporotic detection, so more CAD systems with improved accuracy could be built. DCNN has shown many good results in identifying the objects in medical field, so in osteoporosis detection more work is to be done in this field. Finally, we suggest the use of dental images as a tool for diagnosing osteoporosis.

References

1. The worldwide problem of osteoporosis: insights afforded by epidemiology
2. NIH (2001) Consensus development panel on osteoporosis prevention, diagnosis, and therapy. *Osteoporosis prevention, diagnosis, and therapy*. JAMA 285(6):785–795
3. Kaplan FS (1985) Osteoporosis. *Women's health* 10(2/3):95–114
4. Graham BA, Gleit CJ (1984) Osteoporosis: a major health problem in postmenopausal women. *Orthop Nurs* 3(6):19–26
5. Tu KN, Lie JD, Wan CKV, Cameron M, Austel AG, Nguyen JK, Van K, Hyun D (2018) Osteoporosis: a review of treatment options. *Pharm Ther* 43(2):92
6. Rosholm A, Hyldstrup L, Baeksgaard L, Grunkin M, Thodberg HH (2001) Estimation of bone mineral density by digital X-ray radiogrammetry: theoretical background and clinical testing. *Osteoporos Int* 12(11):961–969
7. Brett AD, Brown JK (2015) Quantitative computed tomography and opportunistic bone density screening by dual use of computed tomography scans. *J Orthop Transl* 3(4):178–184
8. Guglielmi G, de Terlizzi F (2009) Quantitative ultrasound in the assessment of osteoporosis. *Eur J Radiol* 71(3):425–431
9. Hildebolt CF, Zerbolio DJ Jr, Shrout MK, Ritzi S, Gravier MJ (1992) Radiometric classification of alveolar bone health. *J Dent Res* 71(9):1594–1597
10. Southard KA, Southard TE (1992) Quantitative features of digitized radiographic bone profiles. *Oral Surg Oral Med Oral Pathol* 73(6):751–759
11. Lurie AG, Greenberg RJ, Kornman KS (1983) Subtraction radiology demonstrates crestal bone loss in experimentally induced marginal periodontitis. *Oral Surg Oral Med Oral Pathol* 55(5):537–541
12. Hausmann E, Christersson L, Dunford R, Wikesjo U, Phylo J, Genco RJ (1985) Usefulness of subtraction radiography in the evaluation of periodontal therapy. *J Periodontol* 56:4–7
13. Shrout MK, Hildebolt CF, Vannier MW (1993) Effects of region of interest outline variations on gray-scale frequency distributions for alveolar bone. *Oral Surg Oral Med Oral Pathol* 75(5):638–644
14. Isenberg G, Goldman HM, Spira J, Parsons FG, Street PN (1968) Radiograph analysis by two-dimensional microdensitometry. *J Am Dent Assoc* 77(5):1069–1073
15. Kribbs PJ, Smith DE, Chesnut CH (1983) Oral findings in osteoporosis. Part I: measurement of mandibular bone density. *J Prosthet Dent* 50(4):576–579
16. Ruttimann UE, Webber RL, Hazelrig JB (1992) Fractal dimension from radiographs of peri-dental alveolar bone: a possible diagnostic indicator of osteoporosis. *Oral Surg Oral Med Oral Pathol* 74(1):98–110
17. Bras J, Van Ooij CP, Abraham-Inpijn L, Kusen GJ, Wilmink JM (1982) Radiographic interpretation of the mandibular angular cortex: a diagnostic tool in metabolic bone loss. Part I: normal state. *Oral Surg Oral Med Oral Pathol* 53:541–545
18. Benson BW, Prihoda TJ, Glass BJ (1991) Variations in adult cortical bone mass as measured by a panoramic mandibular index. *Oral Surg Oral Med Oral Pathol* 71(3):349–356
19. Horner K, Devlin H, Alsop CW, Hodgkinson IM, Adams JE (1996) Mandibular bone mineral density as a predictor of skeletal osteoporosis. *Br J Radiol* 69(827):1019–1025

20. Southard TE, Southard KA (1996) Detection of simulated osteoporosis in maxillae using radiographic texture analysis. *IEEE Trans Biomed Eng* 43(2):123–132
21. Law AN, Bollen AM, Chen SK (1996) Detecting osteoporosis using dental radiographs: a comparison of four methods. *J Am Dent Assoc* 127(12):1734–1742
22. Taguchi A, Sanada M, Krall E, Nakamoto T, Ohtsuka M, Suei Y, Tanimoto K, Kodama I, Tsuda M, Ohama K (2003) Relationship between dental panoramic radiographic findings and biochemical markers of bone turnover. *J Bone Miner Res* 18(9):1689–1694
23. Klemetti E, Kolmakov S, Kröger H (1994) Pantomography in assessment of the osteoporosis risk group. *Eur J Oral Sci* 102(1):68–72
24. Arifin AZ, Asano A, Taguchi A, Nakamoto T, Ohtsuka M, Tsuda M, Kudo Y, Tanimoto K (2006) Computer-aided system for measuring the mandibular cortical width on dental panoramic radiographs in identifying postmenopausal women with low bone mineral density. *Osteoporos Int* 17(5):753–759
25. Allen PD, Graham J, Farnell DJ, Harrison EJ, Jacobs R, Nicopolou-Karayianni K, Lindh C, van der Stelt PF, Horner K, Devlin H (2007) Detecting reduced bone mineral density from dental radiographs using statistical shape models. *IEEE Trans Inf Technol Biomed* 11(6):601–610
26. Castro-Mateos I, Pozo JM, Cootes TF, Wilkinson JM, Eastell R, Frangi AF (2014) Statistical shape and appearance models in osteoporosis. *Curr Osteoporos Rep* 12(2):163–173
27. Arifin AZ, Asano A, Taguchi A, Nakamoto T, Ohtsuka M, Tsuda M, Kudo Y, Tanimoto K (2007) Developing computer-aided osteoporosis diagnosis system using fuzzy neural network. *JACIII* 11:1049–1058
28. Herumurti D, Arifin AZ, Sulaiman R, Asano A, Taguchi A, Nakamoto T, Uchimura K (2010) Weighted fuzzy ARTMAP for osteoporosis detection. *J Inst Electron Eng Korea*, pp 89–95
29. Kasuba T (1993) Simplified fuzzy ARTMAP. *AI Expert* 8(11):18–25
30. Roberts MG, Graham J, Devlin H (2010) Improving the detection of osteoporosis from dental radiographs using active appearance models. In: 2010 IEEE international symposium on biomedical imaging: from nano to macro. IEEE, pp 440–443
31. Kavitha MS, Samopa F, Asano A, Taguchi A, Sanada M (2012) Computer-aided measurement of mandibular cortical width on dental panoramic radiographs for identifying osteoporosis. *J Invest Clin Dent* 3(1):36–44
32. Kavitha MS, Kurita T, Asano A, Taguchi A (2012) Automatic assessment of mandibular bone using support vector machine for the diagnosis of osteoporosis. In: 2012 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, pp 214–219
33. Kavitha MS, Asano A, Taguchi A, Heo MS (2013) The combination of a histogram-based clustering algorithm and support vector machine for the diagnosis of osteoporosis. *Imaging Sci Dent* 43(3):153–161
34. Prafiadi H, Putra NK (2013) Image analysis for correlation between dental panoramic and MicroCT to measure bone density. In: 2013 3rd international conference on instrumentation, communications, information technology and biomedical engineering (ICICI-BME). IEEE, pp 359–362
35. Roberts MG, Graham J, Devlin H (2013) Image texture in dental panoramic radiographs as a potential biomarker of osteoporosis. *IEEE Trans Biomed Eng* 60(9):2384–2392
36. Loh WY (2011) Classification and regression trees. *Wiley Interdisc Rev Data Min Knowl Discovery* 1(1):14–23
37. Kavitha MS, An SY, An CH, Huh KH, Yi WJ, Heo MS, Lee SS, Choi SC (2015) Texture analysis of mandibular cortical bone on digital dental panoramic radiographs for the diagnosis of osteoporosis in Korean women. *Oral Surg Oral Med Oral Pathol Oral Radiol* 119(3):346–356
38. Kavitha MS, Ganesh Kumar P, Park SY, Huh KH, Heo MS, Kurita T, Asano A, An SY, Chien SI (2016) Automatic detection of osteoporosis based on hybrid genetic swarm fuzzy classifier approaches. *Dentomaxillofacial Radiol* 45(7):20160076
39. Beattie A, Courcane S, Finucane C, Walsh JB, Stassen LF (2018) Quantitative ultrasound of the mandible as a novel screening approach for osteoporosis. *J Clin Densitometry* 21(1):110–118

40. Shen D, Wu G, Suk HI (2017) Deep learning in medical image analysis. *Annu Rev Biomed Eng* 19:221–248
41. Lee JS, Adhikari S, Liu L, Jeong HG, Kim H, Yoon SJ (2019) Osteoporosis detection in panoramic radiographs using a deep convolutional neural network-based computer-assisted diagnosis system: a preliminary study. *Dentomaxillofacial Radiol* 48(1):20170344

Chapter 23

An Expert Diagnosis System for Parkinson’s Disease Using Bagging-Based Ensemble of Polynomial Kernel SVMs with Improved GA-SVM Features Selection



Vinod J. Kadam, Atharv A. Kurdukar and Shivajirao M. Jadhav

1 Introduction

Neurodegeneration diseases are incurable, age-dependent, and multifactorial debilitating disorders of the nervous system. The persistent loss of certain dopamine-secreting neural tissues is the main cause of neurodegeneration diseases [1]. Parkinson’s disease (PD) is a chronic and progressive motoric neurodegeneration disease that predominantly affects older adults. PD is a worldwide problem and the second most frequent (common) long-term human neurodegeneration disease following Alzheimer’s disease (AD) [1]. The main causes (pathogenesis) of PD are not clear (unknown), and powerful treatments are lacking. Recognition and detection of PD at early stages are very important as it can provide an opportunity for an early medication that may be essential to prevent further progress of the disorder, relieve the signs/symptoms, and improve the quality of life for patients. According to many studies available in the literature, changes in voice and quality of a person’s voice is potentially the most vital biomarker and nearly 90% of the PD patients exhibit vocal impairment symptoms (dysphonia) [2]. Recently, many researchers have proposed different machine learning techniques for classification of PD cases through proper representations from speech dataset (dysphonic measures). These machine learning techniques include support vector machine [3–5], kernel methods [3, 6],

V. J. Kadam (✉) · A. A. Kurdukar · S. M. Jadhav

Department of Information Technology, Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, Maharashtra, India

e-mail: vjkadam@dbatu.ac.in

A. A. Kurdukar

e-mail: 3atharvkurdukar@gmail.com

S. M. Jadhav

e-mail: smjadhav@dbatu.ac.in

multiclass multi-kernel relevance vector machines [7], Dirichlet process mixtures [8], multinomial logit models, decision trees [9], neural networks [9, 10], expectation–maximization algorithm [11], similarity classifier [12], rotation forest ensemble [13], optimum-path forest classifier, KNN [14, 15], fuzzy KNN [16], Naive Bayes, deep neural network classifier [17], e.g., stacked sparse autoencoders [2, 15, 18] and softmax classifiers [2, 18], etc., feature selection and extraction approaches like PCA [16], mutual information-based feature selection [4], CFS [13], and evolutionary algorithms, e.g., genetic algorithm [11], PSO [19], harmony memory, gravitational search algorithm, etc. In this paper, an attempt is made to investigate the cubic (degree 3) polynomial kernel-based SVM ensemble classifier with GA-SVM selected features in constructing an automatic classification model for diagnosis of Parkinson’s disease based on vocal measurements. The proposed model includes genetic algorithm with 10-fold CV SVM fitness function for feature selection [20], bootstrap aggregating method [21] to construct an ensemble of classifiers, and double-layer hierarchical combining to make a collective prediction.

2 Proposed Approach

The proposed method has two consecutive phases: feature selection and classification. In the first phase, we applied a novel GA-SVM approach to select relevant features, and in the second phase, we constructed an ensemble of polynomial kernel SVM classifiers using bagging (bootstrap aggregating) approach. The proposed approach is explained in Fig. 1.

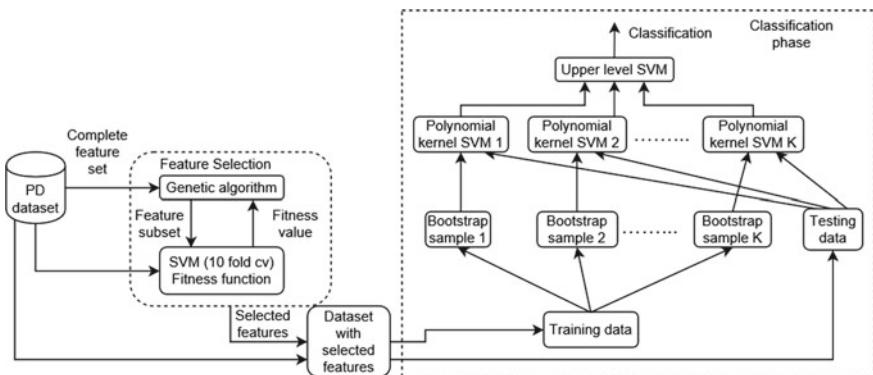


Fig. 1 Proposed model: bagging-based ensemble of polynomial kernel SVMs with improved GA-SVM features selection

2.1 GA-SVM Feature Selection

The dataset may contain many irrelevant and non-informative features which drastically reduce the overall true accuracy of the base classifier. It is very essential to remove these features and select more relevant features (having more discriminative power) from the given dataset before the classification task. Here, we used a GA as an optimal feature selector. Genetic algorithms (GA) are a stochastic, adaptive, and domain-independent general search method, optimization strategy, and population-based evolutionary algorithm. We also proposed a novel fitness function based on the classification error rate achieved by the 10-fold cross-validated SVM classifier. The GA minimizes this classification error. In other words, the objective of the genetic algorithm was to choose the best feature subset which yielded the lowest 10-fold CV classification error rates when using the SVM as a classification approach. Chromosomes (feature subset) are represented as F bit binary string of 1s and 0s with a 1 showing the presence of a feature and a 0 its absence. F is a total number of features available in the given dataset [20]. We also used elitist selection strategy in the proposed genetic algorithm.

2.2 Polynomial Kernel SVMs Ensemble Using Bagging

The support vector machines (SVM) are powerful, promising, and very effective supervised learning classifiers and established as one of the standard tools a binary classification (linear as well as nonlinear classification) due to its outstanding generalization capability and better empirical performance. The SVM classifier formally learns how to separate different classes by forming separating hyperplanes. SVM is mainly a discriminative classifier approach that does classification by obtaining the optimal (or best) separating hyperplane that provides the maximum margin distance between the nearest points of the two classes [22, 20]. This appears simple, but in real life, not all data are linearly separable. This is where kernel trick plays an important role. SVM is a member of a group of algorithms called kernel methods (kernel machines). The kernel trick involves transforming feature set into another dimension (high-dimensional feature space) that has precise separating margin between given classes. Simply, kernel trick provides a modular way to learn nonlinear features using linear models. Kernels functions applied with support vector machines are linear, radial basis function (also called the Gaussian kernel), polynomial, and sigmoid. Here, we used the polynomial kernel function. The format of the polynomial kernel is given below [23].

$$K(x, y) = (x^T y + 1)^d \quad (1)$$

Here, x and y are vectors (data points) in the given input space and d is degrees of the polynomial. One of the advantages of the polynomial kernel is that it looks

not only at the provided features of input data to determine their similarity, but also combinations of these features. In many cases, it provides a better result than linear SVM classifier. In this study, we used cubic (degree 3) polynomial kernel SVMs. Here, we proposed the ensemble of polynomial kernel SVMs. Feature selected using genetic algorithm- SVM wrapper were supplied to this ensemble classifier. Each individual polynomial kernel SVM was trained separately using randomly chosen samples through a bootstrap aggregating (bagging) method. Then, to combine (aggregate) these trained individual classifiers and to make a collective prediction, we used another upper-layer linear kernel SVM. This method is called double-layer hierarchical combining.

3 Experimentation and Results

3.1 Dataset

To validate the performance of our classifier for Parkinson's disease detection, we used Oxford PD detection dataset (created and donated by Little et al.) of biomedical voice measurements [24]. It contains 197 total samples (voice recordings) and a total of 22 vocal features (statistical parameters) [25].

3.2 Experimentation

We coded both feature selection module and ensemble classifier module in the MATLAB 2016b environment. The parameter configuration used for GA is shown in Table 1. We implemented the SVM-based fitness function using three MATLAB functions: ‘fitcsvm’ (with default parameters setting except standardization flag was set true), crossval, and kFoldLoss. The GA-SVM feature subset selector phase gave 12 optimal features. These 12 optimal features were applied to ensemble classifier.

Table 1 Parameter configuration for GA

Parameter	Value	Parameter	Value
Population size	500	Genome length	279
Population type	Bitstrings	Number of generations	100
Crossover	Arithmetic	Crossover probability	0.8
Mutation	Uniform	Mutation probability	0.2
Selection scheme	Tournament of size 2	Elite count	50

Table 2 Performance of proposed ensemble of cubic (degree 3) polynomial kernel SVMs using different numbers of SVMs without and with GA-SVM feature selection

No. of classifiers	With all (22) feature			With GA-SVM selected (12) features		
	Accuracy	Specificity	Sensitivity	Accuracy	Specificity	Sensitivity
11	90.77	93.88	81.25	94.88	97.28	87.5
22	90.77	93.2	83.33	94.36	97.28	85.42
33	90.26	93.2	81.25	94.36	96.6	87.5
44	90.26	93.88	79.17	94.88	97.96	85.42
55	92.31	94.56	85.42	95.9	97.28	91.67
66	93.33	95.24	87.5	96.42	97.96	91.67
77	91.28	93.88	83.33	95.39	96.6	91.67
88	92.31	94.56	85.42	94.88	97.28	87.5
99	91.79	95.92	79.17	95.39	97.96	87.5

Every single cubic (degree 3) polynomial kernel-based support vector machine is trained separately using the randomly picked training samples via a bootstrap resampling method. Then, to combine (aggregate) these trained individual classifiers and to make a collective prediction, we used another upper-layer linear SVM. Ten-fold CV method adopted to compare the performance of the classifier. To code each individual SVM classifier, we used the MATLAB function ‘fitcsvm.’ The optimal value of parameters ‘KernelScale’ flag and ‘Box Constraint’ flag C obtained using the grid search method. We also varied the number of SVMs in the ensemble to study its effect. The performances of the proposed classifier with and without feature selection are provided in Table 2 and Fig. 2. We achieved the best accuracy 93.33% and 96.42% (using 66 classifiers) without and with GA-SVM features, respectively. With the proposed cubic (degree 3) polynomial kernel SVM ensemble classifier and feature selection method, we obtained a promising classification accuracy 96.42%, sensitivity 97.96%, and specificity 91.67% on UCI Oxford PD dataset. We also compared our approach with other approaches available in the literature (Table 3). The statistical analysis of Table 3 indicates that the proposed approach is also comparable with other approaches.

4 Conclusion

The proposed method has two consecutive phases: feature selection and classification. In the first phase, we applied a novel GA-SVM approach to select relevant features, and in the second phase, we constructed an ensemble of polynomial kernel SVM classifiers using bagging (bootstrap aggregating) approach. To evaluate and compare the performance of the proposed approach, the Oxford Parkinson's disease detection dataset was used. The GA-SVM feature selection phase helped

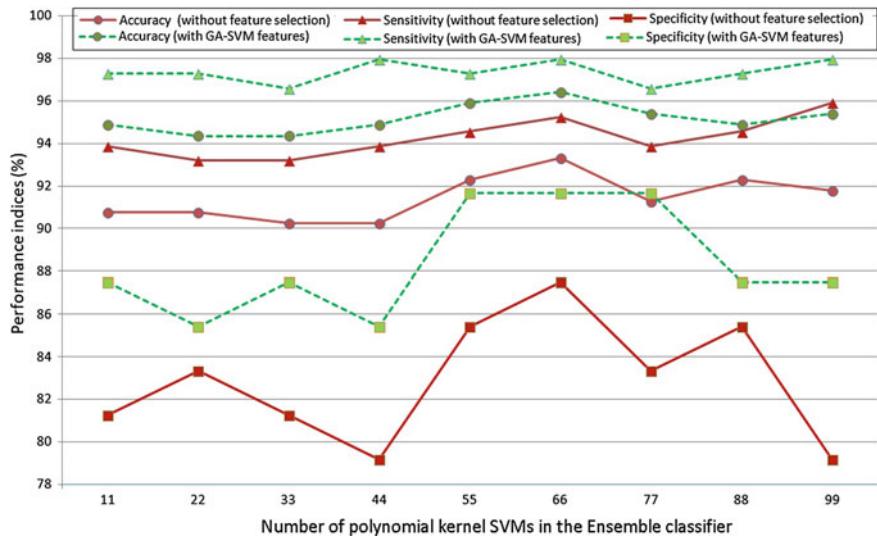


Fig. 2 Prediction results of proposed ensemble classifier using different numbers of SVMs (without and with GA-SVM feature selection)

Table 3 Comparison with other approaches

Year	Study (Ref.)	Acc.%
2009	Kernel SVM (bootstrap with 50 replicates) [3]	91.4
2009	Dirichlet process mixtures (5-fold CV) [8]	87.7
2010	Multiclass relevance vector machines (10-fold CV) [7]	89.47
2010	Genetic Prog. and the expectation–maximization algo. [11]	93.1
2010	Mutual information + SVM (bootstrap with 50 replicates) [4]	92.75
2010	ANN (hold-out) [9]	92.9
2011	CFS + rotation forest (RF) ensemble (10-fold CV) [13]	87.1
2011	Fuzzy-based nonlinear transformation + SVM (hold-out) [5]	93.47
2011	Parallel neural network (hold-out) [10]	91.2
2013	PCA-fuzzy k-nearest neighbor (10-fold CV) [16]	96.07
2013	SVM with chi-square distance kernel (hold-out) [6]	91.2
2017	SAE and KNN classifier (hold-out) [15]	94–98
2017	Stacked autoencoder + softmax classifier (10-fold CV) [18]	93.79
2019	Stacked sparse autoencoders and softmax (10-fold CV) [2]	92.19
2019	Feature ensemble using sparse autoencoders (10-fold CV) [2]	93.84
2019	[This study] bagging-based ensemble of cubic polynomial kernel SVMs (10-fold CV)	93.33
2019	[This study] bagging-based ensemble of polynomial kernel SVMs with improved GA-SVM features selection (10-fold CV)	96.42

us to reduce over-fitting, improve the accuracy of the classifier, and reduce training time by selecting the best feature subset. The experimental outcome and statistical analyses indicate that the proposed ensemble classifier is very efficient and practical model for Parkinson disease diagnosis. High true accuracy obtained by this approach also ensures that this approach is comparable to the best approaches available in the related literature.

References

1. Ahmad K et al (2017) Commonalities in biological pathways, genetics, and cellular mechanism between Alzheimer disease and other neurodegenerative diseases: in silico-updated overview. *Curr Alzheimer Res* 14(11):1190–1197. <https://doi.org/10.2174/1567205014666170203141151>
2. Kadam VJ, Jadhav SM (2019) Feature ensemble learning based on sparse autoencoders for diagnosis of Parkinson's disease. In: Iyer B, Nalbalwar S, Pathak N (eds) Computing, communication and signal processing. Advances in intelligent systems and computing, vol 810. Springer, Singapore
3. Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO (2009) Suitability of dysphonia measurements for telemonitoring of Parkinson's Disease. *IEEE Trans Biomed Eng* 56(4):1015–1022. <https://doi.org/10.1109/TBME.2008.2005954>
4. Sakar CO, Kursun Olcay (2010) Telediagnosis of Parkinson's disease using measurements of dysphonia. *J Med Syst* 34:591–599. <https://doi.org/10.1007/s10916-009-9272-y>
5. Li D-C, Liu C-W, Hu S (2011) A fuzzy-based data transformation for feature extraction to increase classification performance with small medical data sets. *Artif Intell Med* 52:45–52. <https://doi.org/10.1016/j.artmed.2011.02.001>
6. Daliri MR (2013) Chi-square distance kernel of the gaits for the diagnosis of Parkinson's disease. *Biomed Sig Process Control* 8:66–70
7. Psorakis I, Damoulas T, Girolami MA (2010) Multiclass relevance vector machines: sparsity and accuracy. *IEEE Trans Neural Netw* 21(10):1588–1598. <https://doi.org/10.1109/TNN.2010.2064787>
8. Shahbaba, B, Neal R (2009) Nonlinear models using Dirichlet process mixtures. *J Mach Learn Res*, 1829–1850
9. Das R (2010) A comparison of multiple classification methods for diagnosis of Parkinson disease. *Expert Syst Appl* 37:1568–1572
10. Åström F, Koker R (2011) A parallel neural network approach to prediction of Parkinson's Disease. *Expert Syst Appl* 38(10):12470–12474
11. Guo PF, Bhattacharya P, Kharma N (2010) Advances in detecting Parkinson's disease. In: Zhang D, Sonka M (eds) Medical biometrics. ICMB 2010. Lecture notes in computer science, vol 6165. Springer, Berlin, Heidelberg
12. Luukka P (2011) Feature selection using fuzzy entropy measures with similarity classifier. *Expert Syst Appl* 38(4):4600–4607. <https://doi.org/10.1016/j.eswa.20; https://doi.org/10.09.133>
13. Ozciift A, Gulten A (2011) Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. *Comput Methods Programs Biomed* 104(3):443–451
14. Polat K (2012) Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering. *Int J Syst Sci* 43(4):597–609
15. Zhang YN (2017) Can a smartphone diagnose Parkinson Disease? A deep neural network method and telediagnosis system implementation. Parkinson's disease, Article ID 6209703, 11 pages. <https://doi.org/10.1155/2017/6209703>

16. Chen H-L et al (2013) An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach. *Expert Syst Appl* 40(1):263–271
17. Kadam VJ, Jadhav SM, Vijayakumar K (2019) Breast cancer diagnosis using feature ensemble learning based on stacked sparse autoencoders and softmax regression. *J Med Syst* 43:263. <https://doi.org/10.1007/s10916-019-1397-z>
18. Caliskan A et al (2017) Diagnosis of the Parkinson disease by using deep neural network classifier. *Istanbul Univ-J Electr Electron Eng* 17(2):3311–3318
19. Zuo W-L, Wang Z-Y, Liu T, Chen H-L (2013) Effective detection of Parkinson's disease using an adaptive fuzzy k-nearest neighbor approach. *Biomed Sig Process Control* 8(4):364–373
20. Kadam VJ, Yadav SS, Jadhav SM (2020) Soft-margin SVM incorporating feature selection using improved elitist GA for Arrhythmia classification. In: Abraham A, Cherukuri A, Melin P, Gandhi N (eds) Intelligent systems design and applications. ISDA 2018. Advances in intelligent systems and computing, vol 941. Springer, Cham
21. Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140
22. Kadam V, Jadhav S, Yadav S (2019) Bagging based ensemble of support vector machines with improved elitist GA-SVM features selection for cardiac arrhythmia classification. *Int J Hybrid Intell Syst*, 1–9. <https://doi.org/10.3233/his-190276>
23. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press
24. Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM (2007) Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMed Eng* 6:23
25. Dua D, Graff C (2019) UCI machine learning repository <http://archive.ics.uci.edu/ml>. University of California, School of Information and Computer Science, Irvine

Part III

Communication and Networks

Chapter 24

Case Study: Use of AWS Lambda for Building a Serverless Chat Application



Brijesh Choudhary, Chinmay Pophale, Aditya Gutte, Ankit Dani
and S. S. Sonawani

1 Introduction

Organizations such as WhatsApp, Instagram and Facebook have long been promoting conversational user interface (UI) based applications that are based on either text or voice or both. This has recently resulted in growing interest regarding how relevant technologies could be used to improve business in wide industry domains.

The latest trend in cloud computing constructs is the use of serverless architecture [1]. This fast-developing cloud model includes the provision of a server and its managerial operations by the cloud provider itself which eliminates the need for the user to maintain, monitor and schedule servers. This greatly simplifies the methodology of deploying the code into production process. Serverless code is employed in conjunction with code deployed in generic code designs, like micro-services. Majority of the serverless applications run in stateless compute blocks that are triggered by events. The pricing depends on the number of executions rather than already purchased number of compute capacity servers.

This paper reports a month-long case study where a chat application was built based on Amazon Web Services' Lambda framework which, as mentioned above, is

B. Choudhary (✉) · C. Pophale · A. Gutte · A. Dani · S. S. Sonawani
Department of Computer Engineering, Maharashtra Institute of Technology, Pune, India
e-mail: brijesh.choudhary7@gmail.com

C. Pophale
e-mail: chinmay2997@gmail.com

A. Gutte
e-mail: adityagutte@gmail.com

A. Dani
e-mail: ankitdani1997@gmail.com

S. S. Sonawani
e-mail: shilpa.sonawani@mitpune.edu.in

a rapidly evolving serverless computing [2] platform. The paper also covers explorations about various services and components of Amazon Web Services and their usage [3].

2 Literature Survey

For security and authentication, numerous techniques provided by AWS were discovered including the use of a single sign-on session. OTP-based authentication was found to be more reliable and secure over HTTP-based methods [4]. Cloud implications relating to infrastructure elasticity, load balancing, provisioning variation, infrastructure and memory reservation size [5] were studied. This study helps to implement abstraction for serverless architecture which enables efficient cross-server data management, resource allocation and isolation among other things [6]. Different frameworks that cater to certain criteria of real-life applications [7] were also studied. The comparison between various cloud platforms was carried out [8]. Interactive uses of various Amazon cloud services were explored [9]. The evaluation of the viability of the services offered by AWS was checked. The major reason for the shift of paradigm to serverless cloud computing applications was the infrastructure cost comparison of running a web application on AWS Lambda. This paper [3] rightly compared the whereabouts. Till now, the major cloud service models were Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS), but with the coming of serverless computing, a new cloud service model is introduced, i.e., Function as a Service (FaaS) [10].

3 Amazon Web Services

AWS is a cloud service platform, providing a plethora of services for the development of software packages and varied applications. The developer is not responsible and receives hassle-free solutions for accessibility, availability, scalability, security, redundancy, virtualization, networking or computing as Amazon handles them automatically looking into the requirements [11]. The below sections further emphasize the various services employed in the development of the chat application.

3.1 Simple Storage Service (S3)

The Amazon S3 in simple terms is built as a storage for the web. It is tailored to enable easier web-scaled computing for the developing community. It has a simple interface that is used for storage and retrieval of any amount of data, at any time, universally on the web.

Fundamentally, it is a key blog store. Each created S3 object has data, a key and metadata. The key name uniquely identifies the object in a bucket. Object metadata is a set of name-value pairs and is set at the time it is uploaded. Blogs are associated with unique key names which make them easier to sort and access. They may contain information such as metadata (e-tag), creation time information, etc. Amazon S3 is extremely durable.

3.2 *Lambda (λ)*

AWS Lambda is a trigger-driven computing cloud platform that allows programmers to code functions on a pay-as-you-go basis without having to define storage or compute resources. One of the most prominent advantages of AWS Lambda is that it uses abstraction and segregates server management from end user. With its use, Amazon itself handles the servers, which enables a programmer to focus more on writing code.

Lambda follows the Function as a Service (FaaS) model and allows any arbitrary function to run on AWS in a wide range of programming languages including Node.js, Python, Java and C#. Users can also utilize code compiler tools, such as Gradle or Maven, and packages to build functions on a serverless platform.

Why Lambda over other services?

Lambda enables the creation of a new execution of functions in milliseconds. The time taken to execute a function, reduces in Lambda. Comparing with other services for instance Elastic Compute Cloud (EC2) wherein a server boots up the OS before doing any work, Lambda bills a user for just the time you took rounded up to the next 100 ms and bill is generated for just the power computed. On the other hand, EC2 bills by the second. Servers are billed till the time they are shut down by the user. The user also might be forced to pay for the computing power he/she might not be using.

For Lambda, all you need to do is code, build and upload your function. From there, triggers will execute it however it is required. In EC2, you have to pay for the server's OS boot time also. For example, if a user has a fast Lambda function, the bill is generated for 100 ms depending on how fast the Lambda function is. But for the same function, EC2 can charge for the minute. For scaling, Lambda can scale up to a thousand parallel executions. In EC2, a user can scale only by adding more servers. A user might wind up paying for the CPU time they might not use. It is just not as granular at how well you can scale the servers which means you end up wasting money. In Lambda, AWS automatically keeps the server up to date. In EC2, you can restart the system with the updated image but that still takes time and has to be done explicitly. For all the above discussed reasons, Lambda is chosen as an integral framework for the application (Table 1).

Table 1 Comparison of AWS Lambda and EC2

Feature	Lambda	EC2
Time to execute	Milliseconds	Seconds to minutes
Billing increments	100 ms	1 s
Configuration	Function	Operating system
Scaling unit	Parallel function executions	Instances
Maintenance	AWS	AWS and user

3.3 Identity Access Management (IAM)

IAM manages users and groups for the user. It defines “roles” which allows a user to define pre-packaged access sets. There is a functionality to assign roles to users, and we also needed to assign roles to our Lambda function and your API Gateway as we built our serverless application [12]. Policies in IAM are the actual access control descriptions. A developer adds a policy to the S3 bucket to actually provide public access to that bucket. This is where the details actually live. Policies cannot do anything on their own. They need to be assigned to a role or a user or a group in order to do something. We need to attach the policy to that entity in order to underline how we interact with that thing. It also manages identity providers and user account settings. Encryption keys are something that AWS can manage through IAM, and it will generate and provide access to encryption keys.

3.4 DynamoDB

DynamoDB is a NoSQL database provided by AWS. It is basically a key-value storage system. It does not typically provide atomicity, consistency, isolation, durability (ACID). There is a really good chance that data might be written but may be not immediately.

For retrieving items from DynamoDB:

- Get Item: It is the easiest way to retrieve an item. User just has to provide a key to use it. The result will be a single item, or if the item does not exist, it will throw an error.
- Query: Queries require a hash key. If the user does not have a sort key, there is no need to do a query. User can do a “get item” with a hash key. The client can also put a constraint on sort key for retrieving a particular set of data.
- Scan: If you do not know a key, you can use a scan. It is a brute force approach in which the entire dataset is scanned. A filter can be applied to retrieve particular information from the dataset.

In DynamoDB, a user needs to set different capacities for read and write functionalities. DynamoDB gives an option of how much consistency is needed, but the prices get increased gradually.

- 1 read unit = 1 consistent read up to 4 kb/s.
- 1 write unit = 1 write up to 1 kb/s.

3.5 API Gateway

Amazon API Gateway is a service offering that allows a developer to deploy non-AWS applications to AWS back-end resources, such as servers. This allows two or more programs to communicate with each other to achieve greater efficiency. A user creates and manages APIs within API Gateway, which receives and processes concurrent calls. A developer can connect to other services such as EC2 instances, AWS Elastic Beanstalk and code from AWS Lambda. In order to create an API, a developer defines its name, an HTTP function, how the API integrates with services and how requests and transfers are handled. It accepts all payloads including JavaScript Object Notation (JSON) and Extensible Markup Language (XML). An AWS user can monitor API calls on a metrics dashboard in Amazon API Gateway and can retrieve error, access and debug logs from Amazon's CloudWatch. The service also allows an AWS user to maintain different versions of an API concurrently.

3.6 Cognito

Amazon Cognito is a User Management System used to manage user pools that controls user authentication and access for mobile applications. It saves and synchronizes client data, which enables an application programmer to focus on writing code instead of building and managing the back-end infrastructure.

Amazon Cognito collects a user's profile attributes into directories referred as user pools and associates information sets with identities and saves encrypted data as key-value pairs within Amazon Cognito sync storage.

3.7 CloudWatch and CloudFront

CloudWatch is used to provide data insights, graphical representation of resources used by the application. The dashboards provided by AWS for CloudWatch have umpteen options for data crunching. CloudWatch is essential for resource monitoring and management for developers, system operators, site reliability engineers and IT managers.

Amazon CloudFront is a middle-ware which resides in between a user request and the S3 data center in a specific region. CloudFront is used for low latency distribution of static and dynamic web content from S3 to the user. Amazon CloudFront is a Content Delivery Network (CDN) proxies and caches web data at edge locations as close to users as possible.

4 Case Study: Serverless Chat Application

This case study involves building a serverless chat application to demonstrate serverless computing through the use of AWS. The application puts to use several of the functionalities and advantages of the serverless computing platform that have been mentioned above. Following are the features of the web-based chat application (Fig. 1):

- User authentication and cloud-based security: Determines the authenticity of the user by verifying the e-mail provided by one-time password (OTP) generation.
- Real-time messaging: Instant delivery of messages to the chat users without any delay.
- Session monitoring: Allows the users to observe the time when a particular message was sent or received.
- Intuitive user interface: Provides the user with an easy to use chat interface.
- Platform independent: Can function on disparate devices such as both personal computers and mobile phones.
- Serverless: Does not involve management and maintenance of any type of servers.
- Scalable: The number of users can be augmented as required.

S3 and DynamoDB: A bucket is created in the AWS Simple Storage Service that consists of the source code of the web-based chat application. The S3 bucket has been given the public access so that other Amazon cloud services are able to utilize

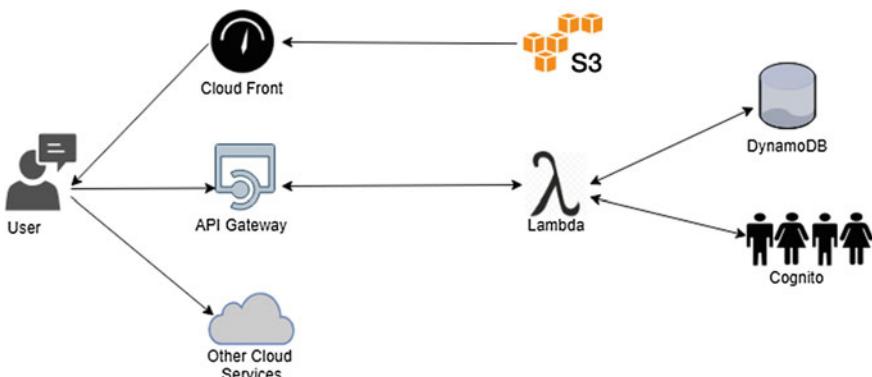


Fig. 1 Model for AWS-based serverless chat application

the code. To maintain the content of the chats and the user sessions, tables have been created in the DynamoDB.

AWS Lambda: AWS Lambda contains the JavaScript methods that are required to run the web application. It is the link between the S3, DynamoDB and AWS Cognito that allows for the execution of the chat application program and resource management for the same. The AWS function used here is `Dynamo.query()` that has the following attributes: table name, projection expression (which acts like a select statement), expression attribute names (`timestamp`) and key conversation expression (used for retrieving conversation id).

AWS Cognito and API Gateways: Amazon Resource Name (ARN) is a naming system used to identify a resource. Several functionalities of API Gateway are Models, Stages and Resources that are utilized for the application.

1. **Models:** It contains the JSON data of the structure of the conversation between the two users.
2. **Stages:** A stage is a named reference to a deployment, which is a snapshot of the API. You use a Stage to manage and test a particular deployment. We have used stages to configure logging, define stage variables and attach a canary release for testing.
3. **Resources:** It consists of the GET and POST methods that are used to request and send data to specific resources.

5 Conclusion and Future Scope

This paper has proposed the use of AWS Lambda for serverless applications and merits of serverless computations. The research is supported by a case study of a serverless chat application built using AWS Lambda and other AWS cloud services. The architecture of the messenger application requires very less management and is cost-efficient. Implementing the messenger application using the above-mentioned technologies helped in grasping the usefulness and relevance of serverless computing and “FaaS” in hosting dynamic applications with lots of user interaction. The future work of this messenger application includes creation of personalized groups and advanced encryption methods to enhance cyber-security. Since Lambda cost model is based on the time required for the code execution, future scope also includes cost and pricing optimizations. Use of other NoSQL databases instead of DynamoDB may be tested in the future implementations of the application. Cross-platform integration of AWS Lambda with other prominent serverless technology providers like Google Cloud and Microsoft Azure Functions should be explored. Serverless computing with the help of AWS and other similar cloud services and their decreasing costs for the resources is here to stay for a long time changing the way we develop, test, deploy and maintain our applications.

References

1. Sewak M, Singh S (2018) Winning in the era of serverless computing and function as a service. In: 2018 3rd international conference for convergence in technology (I2CT), Pune, pp 1–5
2. Lynn T, Rosati P, Lejeune A, Emeakaroha V (2017) A preliminary review of enterprise serverless cloud computing (Function-as-a-service) platforms. In: 2017 IEEE international conference on cloud computing technology and science (CloudCom), Hong Kong, pp 162–169
3. Villamizar M et al (2016) Infrastructure cost comparison of running web applications in the cloud using AWS lambda and monolithic and microservice architectures. In: 2016 16th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid), Cartagena, pp 179–182
4. Swedha K, Dubey T (2018) Analysis of web authentication methods using Amazon web services. In: 2018 9th international conference on computing, communication and networking technologies (ICCCNT), Bangalore, pp 1–6
5. Lloyd W, Ramesh S, Chinthalapati S, Ly L, Pallickara S (2018) Serverless computing: an investigation of factors influencing microservice performance. In: 2018 IEEE international conference on cloud engineering, Orlando, FL, pp 159–169
6. Al-Ali Z et al (2018) Making serverless computing more serverless. In: 2018 IEEE 11th international conference on cloud computing, San Francisco, CA, pp 456–459
7. Kritikos K, Skrzypek P (2018) A review of serverless frameworks. In: 2018 IEEE/ACM international conference on utility and cloud computing companion, Zurich, pp 161–168
8. Kotas C, Naughton T, Imam N (2018) A comparison of Amazon web services and microsoft azure cloud platforms for high performance computing. In: 2018 international conference on consumer electronics, Las Vegas, NV, pp 1–4
9. Yoon H, Gavrilovska A, Schwan K, Donahue J (2012) Interactive use of cloud services: Amazon SQS and S3. In: 2012 12th IEEE/ACM international symposium on cluster, cloud and grid computing, Ottawa, ON, pp 523–530
10. García López P, Sánchez-Artigas M, París G, Barcelona Pons D, Ruiz Ollobarren Á, Arroyo Pinto D (2018) Comparison of FaaS orchestration systems. In: 2018 IEEE/ACM international conference on utility and cloud computing companion (UCC Companion), Zurich, pp 148–153
11. Narula S, Jain A, Prachi (2015) Cloud computing security: Amazon web service. In: 2015 fifth international conference on advanced computing and communication technologies, Haryana, pp 501–505
12. McGrath G, Brenner PR (2017) Serverless computing: design, implementation, and performance. In: 2017 IEEE 37th international conference on distributed computing systems workshops (ICDCSW), Atlanta, GA, pp 405–410

Chapter 25

Detection and Classification of Diabetic Retinopathy Using AlexNet Architecture of Convolutional Neural Networks



Udayan Birajdar, Sanket Gadhave, Shreyas Chikodikar, Shubham Dadhich and Shwetambari Chiwhane

1 Introduction

It is calculable that 415 million people live with polygenic disorder (diabetes) within the world which is estimated to be one in eleven of the world's adult population. Diabetes is a disease in which the body's ability to supply or reply to the endocrine hypoglycemic agent is impaired, leading to abnormal metabolism of carbohydrates and elevated levels of aldohexose or glucose in the blood. One of the foremost areas wherever we are able to observe the impact of diabetes is the human eye. This development is termed as diabetic retinopathy. If left untreated diabetic retinopathy will cause permanent vision defect, then early detection plays a vital role in treating diabetic retinopathy. Traditional methods to observe and classify diabetic retinopathy are complex and include heap of efforts on the part of patient and doctors where the proposed system comes into play. The proposed system works on convolutional neural networks and uses the AlexNet in which a database of human eyes affected with diabetic retinopathy is used for training. The database considered is Kaggle database which consists of over two thousand images which are used for training the model. The trained model detects the presence of diabetic retinopathy and further

U. Birajdar (✉) · S. Gadhave · S. Chikodikar · S. Dadhich · S. Chiwhane
Computer Engineering, NBN Sinhgad School of Engineering, Pune, India
e-mail: udayanbirajdar@gmail.com

S. Gadhave
e-mail: sanketgadhave09@gmail.com

S. Chikodikar
e-mail: chikodikarshreyas98@gmail.com

S. Dadhich
e-mail: s.a.dadhich@gmail.com

S. Chiwhane
e-mail: shwetambari.chiwhane@sinhgad.edu

classifies it based on severity. The classification is based on three major categories which include no diabetic retinopathy which indicates that the eye is clean and has no disease, medium level diabetic retinopathy which indicates that diabetic retinopathy is detected; however, it is treatable and high DR which indicates that immediate attention is required [1]. The programming language used for training the model is Python along with Google's Tensorflow library and Keras image library coupled with Python Image Loading Library. The final model gives accuracy of up to 88%.

2 Literature Review

In 'Detection of diabetic Retinopathy and Cataract by vessel extraction' [2], retinal segmentation is used to extract various features like hemorrhages that help to detect disease and do treatment on them. The segmentation used in this particular paper consists of green filter and then using of CLAHE algorithm to enhance the features. Furthermore, Support Vector Machines are used to detect the diabetic retinopathy and also Feature Vector Machine is used. The drawback of this research paper is that it only gives binary output that is whether diabetic retinopathy is present or absent. No information or detailed report is presented indicating the amount of DR or the severity of diabetic retinopathy.

In 'Fundus Image Texture Features Analysis in Diabetic Retinopathy Diagnosis' [3], the paper investigates texture feature capabilities from fundus images to differentiate between DR, age-related macular degeneration (AMD), and normal. Four experiments are designed for two types of databases namely DIARETDB0 and STARE. The classifiers used are Naïve Bayes, SVM, and KNN. In the case of multiclass classification for images, distinguishing AMD and DR has been a challenge and therefore gives lower accuracy.

In 'Diabetic Retinopathy Screening Based on CNN' [4], a convolutional neural network is used as a classifier. Initially, the database used is MESSIDOR which contains images of diabetic retinopathy affected eyes. These images are preprocessed. In preprocessing, different techniques are used and finally, the results of each technique are calculated. A few image processing techniques used include: AHE, Gauss Noise, Grayscale Images, Green Channel, and Mixed Transformation. The best result obtained is from mixed transformation. However, the accuracy obtained is near to 80% which is quite less. Also, multiclass classification is not given in this paper which is also a drawback as only binary output is present.

In 'Red Lesion detection using dynamic shape features for diabetic retinopathy' [5], a novel method for automatic detection of both microaneurysms and hemorrhages in color fundus images is described and validated. The main contribution is a new set of shape features, called dynamic shape features that do not require precise segmentation of the regions to be classified. These features represent the evolution of the shape during image flooding and allow to discriminate between lesions and vessel segments. The method is validated per-lesion and per image using six databases, four of which are publicly available. On the Messidor database, when detecting images

with diabetic retinopathy, the proposed method achieves an area under the ROC curve of 0.899, comparable to the score of human experts, and it outperforms state-of-the-art approaches. However, less precision is available in this system and automatic DR grading is also not present causing it to fail multiclass classification as well.

3 Convolution Neural Networks Usage in the Model

Convolution neural networks or CNN is a part of the detection of different images and recognition of other images which the computer has never seen. CNN plays an important role in image recognition as it is more accurate than any other system yet developed. The working of CNN takes place through four layers. Similar to neural networks, there are deep CNN learning models. The three layers of CNN include convolution, rectified linear unit or ReLu, pooling and finally, fully connected layer artificial neural network layer [6].

An image is nothing but an array of pixels. In a normal computer system what happens is when an image is given for comparison and recognition, then a normal computer compares the pixel values and checks for any similarities. If there are similarities, then it is given as a match or not a match. However, it may happen that in one image, the required feature is present in different part of the different image causing problems in comparing and detecting.

The functioning of convolution layer is totally different than normal working of computer. In convolution layer, if a particular data set of images is given to train, then the CNN first multiplies the pixel values in image with the selected stride as the hyperparameter. This layer extracts the features from the images by using multiple features which further help to find patterns and recognize them. The next two layers that are rectified linear unit and pooling layer further help to convert negative pixels to positive ones and shrink the image size to lower levels (Fig. 1).

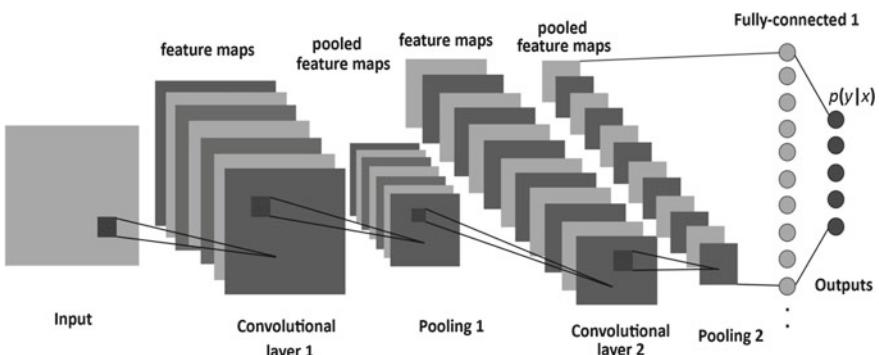


Fig. 1 Feature extraction using convolutional neural networks

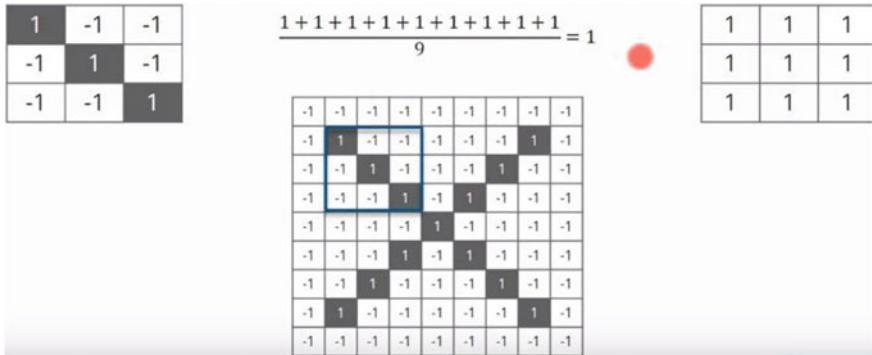


Fig. 2 Convolved image

a. Convolution Layer

This is the first layer of CNN, the convolution layer. In convolution layer, the features are extracted by the system. A particular stride is selected as hyperparameter. In our case, the stride selected is of 4×4 and the number of filters used is 96. The stride is passed on to the whole image and the numbers in the filters that are matrices are multiplied by the numbers present in the pixel values of the images of eyes. This step happens across the whole image and finally, a new table containing numbers with features extracted is formed. This new table obtained is containing values of pixels with the features extracted which helps for further functioning. Consider Fig. 2. The necessary features are extracted then the values of pixels are compared with each pixel, and the pixels are added and then divided by a total number of pixels. This gives the output of convolution layer. This process is done with the help of a 3×3 stride. Such a process helps to reduce the image without loss of any features.

b. Rectified Linear Unit Layer

The ReLu layer is rectified linear unit layer. The working of this layer is simple. After the feature extraction, the ReLu layer deactivates the pixels which are not necessary and only keeps the pixels which are important. From the output of convolution layer, we get positive as well as negative values as the filters are randomly initialized resulting to possible negative values as well. The positive values include successful finding of features and negative values include that those areas are not much important.

The ReLu layer takes the data of the pixels and if the value is positive, then the value is kept as it is and if it is negative, then it is converted to a zero. Thus, the rectified linear unit is a function which returns the value as it is if it is positive and returns zero if the value is negative. Figure 3 shows the exact working. If the value of x is greater than 0, then it retains the value and if it is below 0, then it returns 0.

c. Pooling Layer

The pooling layer does a simple job of reducing the input size. For the pooling layer, generally, a stride is selected. The size of stride is usually 2×2 or 3×3 . That stride

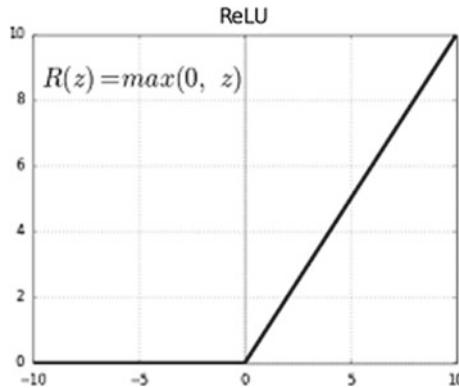


Fig. 3 ReLu layer

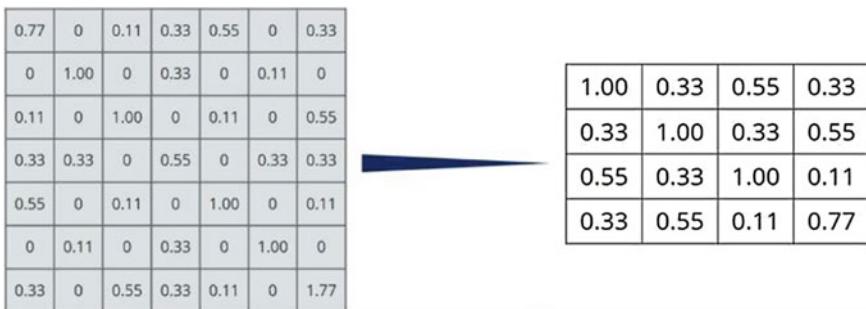


Fig. 4 Max pooling

is walked over the image and the maximum value from the image is extracted and stored in a different table. This helps to reduce the size of the input data a lot. The pooling can be max pooling or min pooling. The max pooling takes the maximum value of the stride, whereas min pooling takes minimum value. In our design, we have used max pooling. This step helps to shrink the features into smaller sized matrices resulting in fast optimization. Figure 5 shows max pooling layer (Fig. 4).

d. Layer Stacking

Finally, this process of convolution ReLu and pooling is carried out once more or any number of times as long as the input becomes small enough for processing by the neural network. This is carried out by stacking the layers one over the other. After stacking the layers, the output obtained is in numerical form. This output is given as input to the neural network and predictions are made with the help of it. This is how convolution neural networks work. In our case, we have used 26-layered convolutional neural network along with the input to the neural network having 9216 neurons in the first layer followed by 4096, 1000, and later two neurons for classification [7].

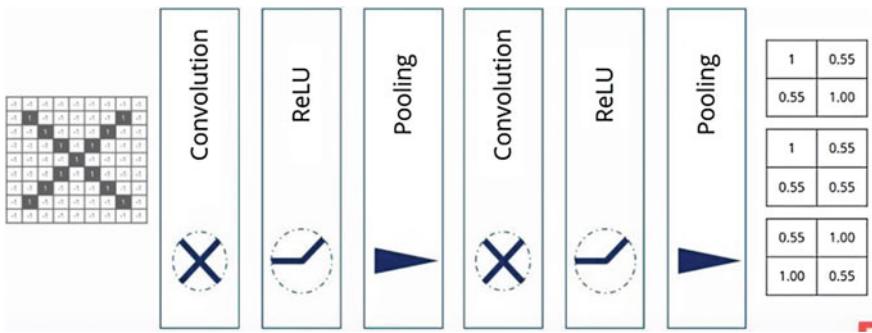


Fig. 5 Layer stacking

e. Fully Connected Layer

This is the final layer in convolutional neural network. It takes the input as numbers from previous layers and feeds the data to the neural network. The fully connected layer consists of multiple neurons and a connection of those neurons to each other. The output of fully connected layer consists of number of classifications. It consists of feedforward neural network and error backpropagation to reduce the errors and get better accuracy. It helps to fine tune the weights and biases.

4 Proposed Algorithm

- i. Getting datasets is one of the tricky jobs. In the proposed paper, the dataset is gathered from various free databases like direct dB 1 and kaggle and also from some doctors of India. Some of the entities in the dataset have been changed or reversed so as to have a variety in the dataset (Fig. 6).
- ii. Image processing forms a huge part of CNN. As clear and fine images need to be provided to the neural network. The images are cleaned and they are resized to fit the AlexNet architecture properly. The valid image size required for AlexNet is 227×227 pixels. RGB images are used as it is for training the model in order to extract the features. RGB images although are complex to process and required a lot of time, more number of features can be extracted from such images.
- iii. Initially, the whole image of eye is broken into smaller pieces of 227×227 . This step converts one whole image into multiple smaller images. This process is carried out using Python programming language and pillow image processing library. From the obtained small pieces, the convolutional neural network is trained.
- iv. The network is trained only of two categories that are a clean eye having no disease and an infected eye containing diseases. This helps the network to identify whether a given piece of image of an eye contains disease or not.

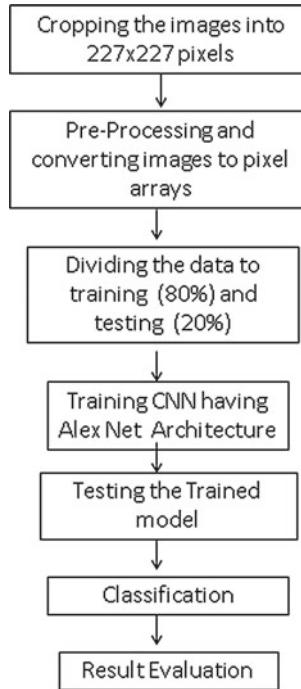


Fig. 6 Flowchart of the system

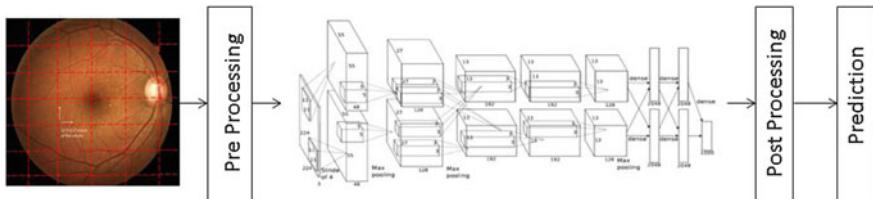
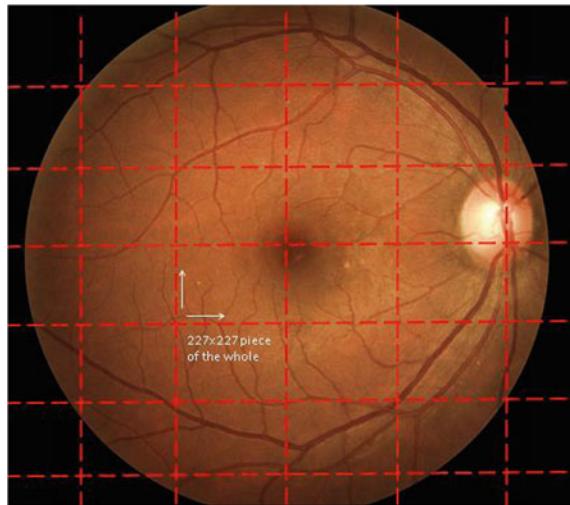


Fig. 7 Detail system architecture

- v. Finally, after training the network, the post-processing takes place. In case of post-processing a new eye image is used, this will be broken down to multiple snapshots of 227×227 images. Each part of the image is identified and each part is given a specific weight. The more the weight, the higher the priority if the disease is detected in that specific part of the eye. Higher weights are given to the area near the optic disk and the center of the eye and the weights are decreased as we move further away from this part.
- vi. In order to extract these features from the images, the proposed model uses CNN which automatically extracts the features by using convolution.
- vii. Multiple layers are stacked for the CNN the layers are shown in Fig. 8.

Fig. 8 Eye image cropping and processing



- viii. The optimizer which is used to train the network is AdaDelta.
- ix. The neural network is trained for 40 epochs that is 40 cycles and the training accuracy obtained is around 88% (Fig. 7).

5 Experiment and Result

- i. The neural network is trained on 2000 images obtained from the dataset.
- ii. The testing of the system is further done on another 20 images.
- iii. In each case, one image of eye is broken into multiple pieces each having 227×227 pixels. Each piece of the image is given a specific weight where the area near the optic disk and the center of the eye having highest weight.
- iv. Higher the weight indicates more intensity of the disease. If multiple diseases are detected in the area having high weight, then the eye can be categorized into high-level diabetic retinopathy.
- v. The trained model is loaded using a Python script and each of the pieces of the eye is given as input to the model.
- vi. Furthermore, the output of the model is analyzed. It can be zero which indicates that no diabetic retinopathy and it can be one indicating diabetic retinopathy is present.
- vii. If category is one, then the weights of the pieces are referred and more the weights having category 1 higher is the probability of medium and high diabetic retinopathy (Fig. 8).

6 Conclusion

This paper represents a new efficient way for detection and classification of diabetic retinopathy for the human eye. With the help of CNN using the AlexNet architecture, proposed system gives accuracy for up to 88%. Also, this system acts supplementary for the doctors and does not focus on replacing the doctors. Furthermore, it can also perform multiclass classification and gives report about the intensity of diabetic retinopathy.

7 Future Scope

As this system if proposed only for laptops, our prime future scope is to implement such a system on android devices. If the classification and detection are done on handheld smartphones, it will be even more beneficial and effective. Another future scope is to detect other diseases related to DR other than just PDR or NPDR.

References

1. Kharabe S, Nalini C (2018) An efficient study on various biometric methods. *Int J Civil Eng Technol*
2. Chorage SS Prof. Dr., Khot SS (2017) Detection of diabetic retinopathy and cataract by vessel extraction from fundus images. In: International conference on electronics, communication and aerospace technology
3. Sarwinda D, Bustamam A, Arymurthy AM (2017) Fundus image texture features analysis in diabetic retinopathy diagnosis. In: International conference on sensing technology. IEEE
4. Seoud L, Hurtut T, Chelbi J, Cheriet F, Pierre Langlois JM (2015) Red lesion detection using diabetic retinopathy screening. In: IEEE transactions on medical Imaging
5. Andonova M, Pavlovipova J, Kajan S, Oravec M, Kurilova V (2017) Diabetic retinopathy scanning based on CNN. In: 5th international symposium
6. Kharabe S, Nalini C (2018) Using adaptive thresholding extraction—robust ROI localization based finger vein authentication. *J Adv Res Dyn Control Syst*
7. Kharabe S, Nalini C (2018) Survey on finger-vein segmentation and authentication. In: *Int J Eng Technol (UAE)*

Chapter 26

Contextual Recommendation and Summary of Enterprise Communication



Anuja Watpade, Nikita Kokitkar, Parth Kulkarni, Vikas Kodag,
Mukta Takalikar and Harshad Saykhedkar

1 Introduction

Professionals and employees are often subscribed to hundreds of public mailing lists or channels to keep themselves updated with their projects. This implies a deluge of mails or messages in their inboxes every day. Any professional subscribed to such public mailing list/channel needs to either sift through thousands of messages or run the risk of missing out on important conversations.

Employees on average spend 13 of their working hours each week in their email inbox. Average number of mails received by an office worker is 121 emails per day.

Professionals often spend a lot of time crafting elaborate filtering rules to keep up with the information overload. However, these filtering rules can get quickly outdated, especially in dynamically evolving projects. We aim to reduce the cognitive overload of dealing with large number of messages. We try to filter messages for a given user based on their past interactions on the channels/mailing lists. There is

A. Watpade · N. Kokitkar · P. Kulkarni (✉) · V. Kodag · M. Takalikar

Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

e-mail: parthmaheshkulkarni@gmail.com

A. Watpade

e-mail: anujanwatpade@gmail.com

N. Kokitkar

e-mail: nikkikokitkar@gmail.com

V. Kodag

e-mail: vikaskodag2@gmail.com

M. Takalikar

e-mail: mstakalikar@pict.edu

H. Saykhedkar

Slack Technologies, Bavdhan, Pune, India

e-mail: harshad.say@gmail.com

no such generalized automated alternative using machine learning approach to filter messages to the best of our knowledge.

2 Previous Work

Google's Priority Inbox [1] ranks mail by the probability that the user will perform an action on that mail. Because importance is highly personal, they try to predict it by learning a peruser statistical model, updated as frequently as possible. It focuses on all the mails received by users. Social features are based on the degree of interaction, e.g., the percentage of a sender's mail read by the recipient. Content features identify headers, presence of a recent term in the subject. Thread features note if a user began a thread. Google keeps track of this data for each of its user which is not possible in our case. We only have the content of the mails posted and the social information based on the frequency of replying.

Dabbish et al. [2] used regression techniques to predict a message's perceived importance from characteristics of the recipient, the sender and the message itself. It identified key message content elements, adapted from the categories which were reminders, social content, requests and responses for action, information, scheduling and status updates. Also, the predictions about how these types should relate to the actions taken on a message were considered.

An intelligent semantic email content technique is proposed [3] that will assist in establishing relations between email messages and conversation threads, help email users handle, organize their email conversation messages and automatically learn as the email contents vary. It works by capturing the sender's addresses, subject held, time that each email conversation threads are sent.

Ayodele et al. [4] predict if an email received requires a reply as well as grouping email based on user's activities and summarizing the emails. Email field headers (CC and BCC) are checked. Email message content and subject field for special words and the presence of question marks are checked. Topics of discussion change over time, and extracting special words and grouping are not the best approach.

Yang et al. [5] propose that email reply behaviors are influenced by various factors such as time, email content, metadata, historical interaction, email addresses, email recipients and attachments. Supervised machine learning models are developed to predict whether a recipient will reply to an email and how long it will take to do so.

3 Problem Statement

To design and implement recommendation system for enterprise communication based on context which includes the content of the message and the social structure of communication. To experiment on different deep learning architectures and other parameters to model the behavior of communication through text. The summarization

task involves providing a short summary of the recommended messages; however, this is not the subject of this paper, but included for completeness, as this paper describes a potential product.

4 Dataset

The dataset for the proposed system has been derived from Debian mailing list (<https://lists.debian.org/>). For the project purpose, multiple types of mailing lists, namely Developers, Dpkg, Release, Cloud and many more were scraped. The dataset after scraping contains a total of 100,812 numbers of mails exchanged between 3094 users from January 2016 to December 2018.

Debian is developed through distributed development all around the world. Therefore, email is the preferred way to discuss various items. Much of the conversation between Debian developers and users is managed through several mailing lists. For the purpose of our project, we kept the threads which had number of mails between 4 and 25. The threads containing less than four mails were discarded as they would not contribute much to the system, and the context of the threads having more than 25 mails often changes, contributing to its dynamic nature. Thus, total of 42,195 mails among 1403 users contributed to the dataset of which 37,362 mails before September 2018 were used for training, and the rest were used for testing (Fig. 2).

Figure 1a shows the traffic of mails per month in the dataset. Figure 1a shows the distribution of thread lengths in the dataset. In Fig. 1b, x-axis represents the frequency of participation, and y-axis represents the number of users for that frequency.

Other datasets which we considered for the proposed system were Gonuts dataset and Linux Kernel mailing list. Both these datasets had large number of mails, but they were not suitable for the project because of the noise present in them. The noise includes code segments from multiple languages and large number of terminal traces in mails.

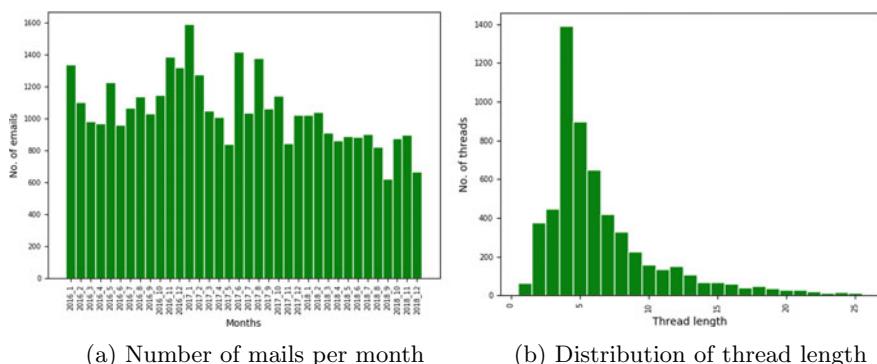


Fig. 1 Statistics of Debian dataset

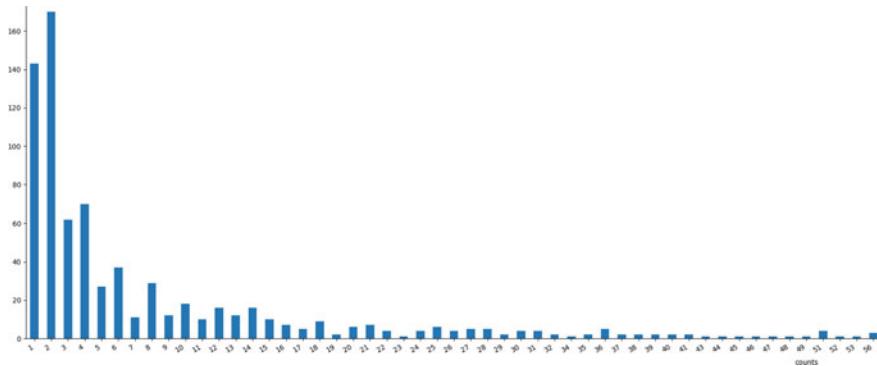


Fig. 2 Frequency of user participation

5 Scraping and Threading

The scraping module scraped the fields: ‘To,’ ‘From,’ ‘Subject,’ ‘Date,’ ‘Message-id,’ ‘In-reply-to’ and ‘References’ from the mailing list and saved them in a file. Threading involves grouping scraped mails into threads. All the mail files were combined and threaded according to the field ‘References’ instead of ‘In-reply-to’ field as there was a hierarchy for replying. The threaded mails were then sorted according to the date to get their relative order.

6 Preprocessing

Preprocessing contains the following four steps:

1. The first step was to remove the reply text from previous messages.
2. The dataset was then divided into three categories, dataset with no short messages (min three sentences or 50 words), dataset with no long messages (not more than 10 lines) and the dataset with no short and long message. This was done to study the effect of length of the message on the accuracy.
3. The lemmatization module was then used to normalize the mail content. NLP Spacy module was used for this purpose.
4. The dataset was then cleaned using standard regex which included removing specific tokens like days, names, special characters and URLs with generalized tokens and removal of null lines.

7 Method/Technical Approach and Models

A neural network consisting of input layer where concatenated word embeddings and user vectors are given as input. Refer Fig. 3. The embeddings are vector representations of the mail body. The next layer is the hidden layer where intermediate processing is done. The output layer has an activation function which gives a softmax vector giving likeliness of each user to replying to the mail.

A user will be interested in a mail based on the topic of the mail. In order to model the contextual information contained in a mail, we use embeddings of the mail body to represent the topic.

The tendency of a user to participate in a conversation also depends on the previous users present in the thread. If thread has participants with whom a user frequently communicates, there is a good chance of him joining the discussion. So, the recommendation model has to consider the social aspect as a feature.

By analyzing the threads, a social graph can be created. This graph will capture the hierarchy and the social relationships of the users. Python networkx library was used to generate the graphs. Refer Fig. 4 for sample graph.

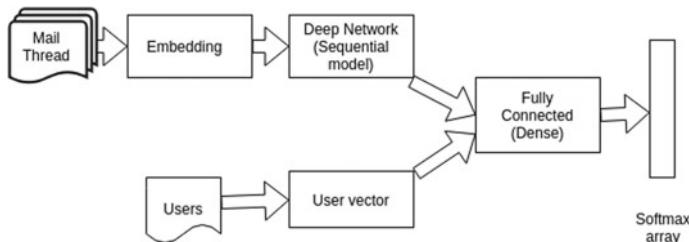


Fig. 3 System architecture

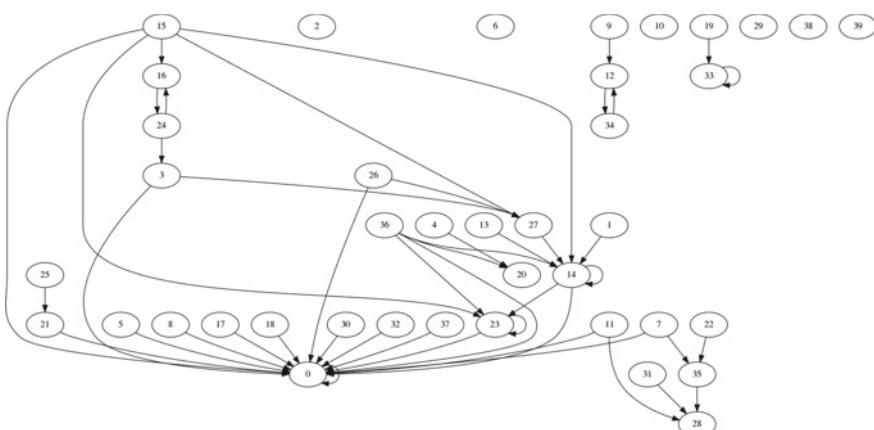


Fig. 4 Social graph

First, a user interaction matrix was created by considering the frequency of communication between the users. The frequency is used as the edge weight, and all the users are nodes. This graph captures the social nature of communication. Density of links gives an idea of the hierarchical nature of communication.

Interaction matrix:

$$\text{mat}[i][j] = m$$

i = id of user to whom the mail is sent

j = id of sender from whom the mail is sent

m = no. of mails sent by i th user to j th user.

We use an extractive summarization technique to provide a summary of the feed. A sentence vector for each sentence is created using the dictionary of all the words in the document. Then, a similarity matrix is created using the cosine similarity between pairs of sentences. Using the similarity matrix, a graph is generated, on which PageRank algorithm is applied, and scores for each sentence are calculated. The sentences are ranked according to their scores, and the top-N sentences are selected and given as the summary of the document. This is a commonly used method, and this paper does not focus on the same.

Metric–Recall@N Recommender systems generating predictions are evaluated such that they propose a list of N items that a user is expected to find interesting (top- N recommendation) and check whether the correct item is present in the list or not. We use this metric such that our model predicts N number of users that are most probable to reply to the thread or mail. If the user who actually replied is present in the list, then it is a hit, and if he is not, then it is a miss. The accuracy is calculated by taking the ratio of the number of hits to the number of total test cases. Around 50% accuracy is viable in our case, as it means that there is 50% chance that the next replier will be among the N (3, 5, 10, 15, 20) users our model is predicting as opposed to the remaining (total— N) users who have the remaining 50% chance of replying.

8 Experiments

8.1 Experiment 1

The results of this experiment can be found in Table 1.

- (a) **One-hot user vector:** A vector of size length equal to the total number of users present in the mailing list is created. The users who have participated in the thread under consideration either by initiating or by replying are set to 1.
- (b) **One-hot user vector with decay:** Many times, a thread becomes too long, and the topic of the thread diverges from the initial topic. The concerned users also

Table 1 Experiments on user vector (figures in %)

No.	Details	Recall@3 (%)	Recall@5 (%)	Recall@10 (%)	Recall@15 (%)	Recall@20 (%)
1	One hot	32.1	38.12	45.8	51.03	54.86
2	One hot with decay	32.49	38.17	45.95	50.50	53.72
3	Personalized PageRank	23.7	30.69	39.45	44.14	48.59

change. There is much greater probability of a user replying based on the recent repliers instead of the initial repliers. By applying a decay function, we decrease the weight of the old repliers. Consequently, the weight of the recent repliers is more, and this ensures that the predictions made by the model are affected more by the recent users. This is done by applying the decay function to the one-hot vectors of the previous step.

- (c) **Personalized PageRank:** Personalized PageRank is a modified algorithm of PageRank, where PageRank values are not initialized as $1/N$. A user node that we want to find the highest rank for it has a rank value of 1. All the other nodes have rank values of 0. Personalized PageRank scores are used in place of one-hot vectors.

8.2 *Experiment 2*

The results of this experiment can be found in Table 2.

- (a) **Text size—removal of short mails:** Mails which contain only one sentence reply (e.g., Thanks, I will look into it.) are also present in the dataset. Such mails do not contain enough context for the model to learn from.
- (b) **Text size—removal of long mails:** Some mails have very long bodies, and these might contain redundant and repetitive information. The topic of discussion of

Table 2 Experiments on text size (figures in %)

No.	Details	Recall@3 (%)	Recall@5 (%)	Recall@10 (%)	Recall@15 (%)	Recall@20 (%)
1	Removing short mails	28.53	34.17	41.87	47.94	52.00
2	Removing long mails	32.63	37.54	46.28	51.38	55.16
3	Long and short	32.45	37.57	44.98	49.90	53.48

Table 3 Experiments on embeddings (figures in %)

No.	Details	Recall@3 (%)	Recall@5 (%)	Recall@10 (%)	Recall@15 (%)	Recall@20 (%)
1	Doc2Vec	32.1	38.12	45.8	51.03	54.86
2	TF-IDF	28.18	33.74	41.556	46.55	50.41

these types of mails can be captured approximately by sampling few sentences (10–15) from the text. The sentences can be sampled by extracting the top- N , last- N or random- N sentences.

- (c) **Text size—removal of long and short mails:** Effect of removing both short and long mails present in the dataset is studied by combining the above two strategies.

8.3 *Experiment 3*

The results of this experiment can be found in Table 3.

- (a) **Google Word2vec:** It has pretrained vectors for 3 million words and phrases trained on Google news dataset (100 billion words) gives embeddings 300-dimensional vectors.
- (b) **TF-IDF:** TF-IDF vectorization allows for differential weighting for words based on how commonly they occur in the corpus.

9 Future Scope

Explicit actions such as marked important, moved to trash or archive, replied immediately, seen but not replied and not seen can be input parameters to the model. Attachments of mails can also be used to extract context. Experiments can be carried out on the deep learning model architecture to increase accuracy.

10 Conclusion

To conclude, we have used various deep Learning and NLP techniques in order to recommend emails to a user based on the context of the mail and the social relationships of communication. We also carried out experiments on the processing of the input to the model, to evaluate which method contributes toward increasing the accuracy of recommendation.

References

1. Google Priority Inbox (Aberdeen, Douglas Pacovsky, Ondrej Slater, Andrew (2019) The learning behind gmail priority inbox.)
2. Dabbish LA, Kraut RE, Fussell S, Kiesler S (2005) Understanding email use: predicting action on a message. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI '05). ACM, New York, USA, pp 691–700. doi: <https://doi.org/10.1145/1054972.1055068>
3. Ayodele T, Akmayeva G, Shoniregun CA (2012) Machine learning approach towards email management. World congress on internet security (WorldCIS-2012), Guelph, ON, pp 106–109
4. Ayodele T, Zhou S (2008) Applying machine learning algorithms for email management. In: 2008 third international conference on pervasive computing and applications, Alexandria, pp 339–344. <https://doi.org/10.1109/icpca.2008.4783606>
5. Yang L, Dumais ST, Bennett PN, Ahmed Hassan P (2017) Characterizing and predicting enterprise email reply behavior, 235–244. <https://doi.org/10.1145/3077136.3080782>

Chapter 27

Cybersecurity and Communication Performance Improvement of Industrial-IoT Network Toward Success of Machine Visioned IR 4.0 Technology



Sachin Sen and Chandimal Jayawardena

1 Introduction

Communication between cyber- and physical systems has become paramount, especially for the Internet of Things (IoT), cyber-physical systems (CPS), and cloud computing. These technologies together leverage the smart industrial processes which include Industrial-IoT (IIoT) as well as Industrie 4.0. The IIoT combines those great technologies CPS, IoT, and cloud computing which have been internetworked to maximize the productivity of the Industry 4.0. The Industrial-IoT has been evolved by combining information technologies, cloud and edge computing platforms, smart networked objects, and CPS technologies; the purpose is to enable intelligent, real-time, and autonomous access, analysis, collection, communication and exchange of service, process, and product information within the industrial environment [1].

The Industrial-IoT exploits machine vision to perform a very critical function in the automation, by intervening fast and effectively in the Industry 4.0 to identify and flag defective products understanding their deficiencies [2]. The advancement of the Industrial-IoT leveraged its integration to the machine visioned Industry 4.0 by empowering industrial processes to ease of intelligent operation [3]. In order to create intelligent, self-optimizing industrial equipment and facilities, the Industrial-IoT is functioning the most important role and is the key element of smart industrial processes or Industry 4.0; its automation trend brings modern cloud computing, machine learning/machine vision, and computational intelligence together [4]. To

S. Sen (✉) · C. Jayawardena

Department of Computer Science, Unitec Institute of Technology, Auckland, New Zealand
e-mail: ssen@unitec.ac.nz

C. Jayawardena

e-mail: cjayawardena@unitec.ac.nz

fulfill the purpose of the IR 4.0 (Industry Revolution 4.0) or the successful operation of Industrial-IoT applications, a smart industrial process needs to be integrated with a TCP/IP network and cloud computing facilities. A sample use case with a SME network controlled automated industrial process has been shown in Fig. 1, where the industrial process is being operated with a wireless sensor network and communicating through to the Internet cloud through the SME network.

The Industrial-IoT network setup in Fig. 1 contains four sections, i.e., manufacturing network zone (MNZ), demilitarized zone (DMZ), local area network (LAN), and the enterprise network zone (ENZ) that includes the virtualized cloud computing networks. The MNZ includes wireless networks with a gateway, cluster heads, and the sensor nodes. The cluster heads collecting information from the sensors and transporting through to the cyber-network via the gateway.

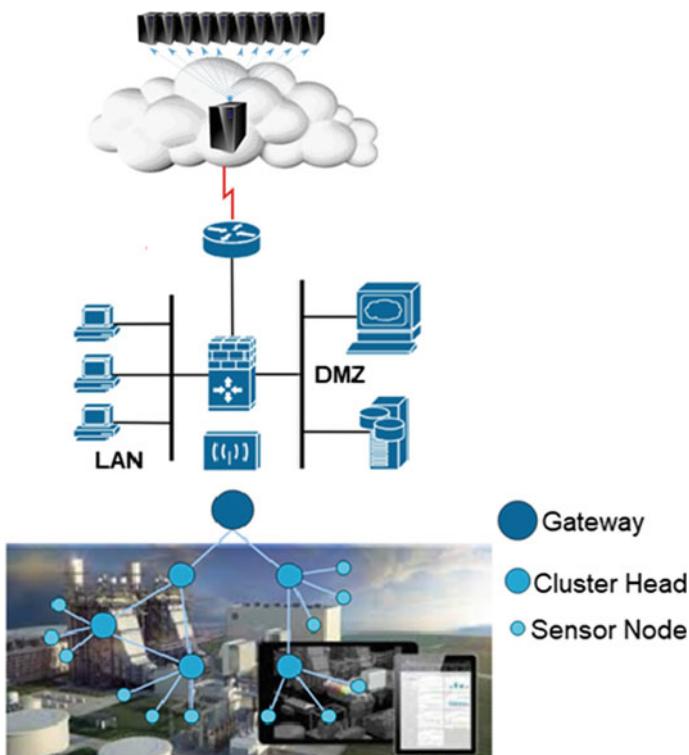


Fig. 1 SME network controlling an automated industrial process

2 Background and Context

The current Internet architecture allows network engineers and application developers to use the technologies independently and also allows boundless innovation as protocols can change without affecting the network [5]. Internet is the most successful communication network, due to the open standard and architecture, OSI and TCP/IP model. According to [5], these open standards made the communication protocols narrowed down to the IPs, hides physical interfaces from applications, packet forwarding is enabled by address schemes and eliminates protocol gateways. But in the cyber-physical communication, the Ethernet or the physical interface is exposed to the cyber-network and being communicated through the low-power wireless sensor networks. This has put the physical systems in a great challenge with respect to performance and security.

In CPS, a physical layer transports data from the physical dynamics to the cyber-layer through the communication network, and the cyber-layer transmits instructions using man-machine interface or actuators to the physical layer [6]. The wireless sensors and controllers will cause bottlenecks of performance and security due to the communication latency between the sensors/actuators and controllers as well as due to security vulnerabilities [7]. A cyber-physical communication infrastructure applicable to the Industrial-IoT has been shown in Fig. 2, which has reflected how the network performance and quality will be evaluated and analyzed.

Due to the machine-to-machine (M2M) connections in the CPS/IoT, the physical layer or the Ethernet is exposed to cyber-network, which can make critical services

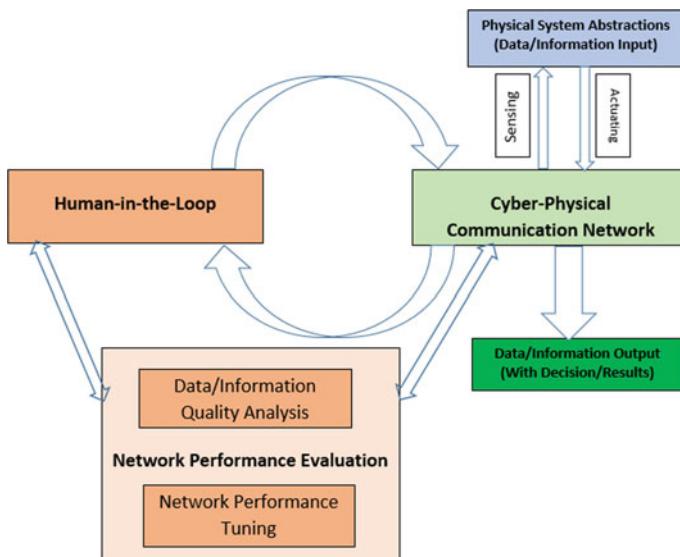


Fig. 2 Cyber-physical communication infrastructure

prone to the cybersecurity. This makes a critical issue in the cyber-physical communication systems, in addition to the network performance due to the communication latency of wireless sensor networks. The unexpected delays between controllers of the physical systems and the wireless sensors could make the distributed control systems substantially worsen [7]. The smart industrial system applications or the IIoT applications, which could also include power system industry, oil/gas, and manufacturing industries, may be posed to critical challenges in their control systems operations, due to the communication latency and cybersecurity issues like denial of service attacks [7].

3 Performance Improvement of Cyber-Physical Communication Network

The Industrial-IoT integrated technologies and applications, like CPS, IoT, and cloud computing, all require a real-time and assured communication process in order to transport data packets between sensors, computational units, and actuators that can provide guarantees on both the throughput and delays of packet flows. But as the current infrastructure of cyber-physical communication doesn't guarantee the reliability, it has become necessary of improving the communication performance. Following subsections will shade some lights on this requirement.

3.1 Performance Improvement by Intelligent Cluster-Head Selection in Wireless Sensor Networks (WSN)

In cluster-based implementation of low-power wireless sensors, clusters are organized with a cluster head which collects and aggregates data/information from its member nodes and sends to the gateway for further processing, in order for getting ready for decision making by the system. For wider implementations depending on business needs, there might be number of clusters; therefore, number of cluster heads will require to communicate in between before reaching to the target wireless gateway.

Therefore, first step is to select cluster heads from among the sensor nodes and then need to generate a routing path; both the steps need to be designed very carefully by intelligent use of the sensor inherent properties. Some wireless network protocols, like radio frequency/free space optical (RF/FSO) based protocols, select cluster heads at the beginning of the network lifetime and continue till the exhaustion of the sensor lifetime of energy [8]. On the other hand, some protocols select cluster heads on rotation basis among all the nodes; this helps the lifetime of a cluster head divided into several rounds. An example is a RF-only based routing protocol low-energy adapting clustering hierarchy (LEACH), in which the lifetime is divided into N/R ,

where N is the number of nodes in the system and R is the number of nodes selected as cluster heads [8].

This way, all the deployed nodes have a chance to become cluster head. Hence, the node energy will be used intelligently as all the nodes are expected to serve the cluster-head roles at least once after N/R rounds. The cluster-head selection probability has been chosen in [8], in which the j th node will become cluster head during the round r according to the following formula:

$$\mathbb{P}_i = \begin{cases} \frac{R}{N-R \times (r \bmod \frac{N}{R})}; & \mathbb{C}_i = 1 \\ 0, & \text{otherwise; } \mathbb{C}_i = 0 \end{cases}$$

where \mathbb{C}_i determines if a node serves as cluster head recently in $(r \bmod \frac{N}{R})$ rounds.

In LEACH, cluster heads are chosen on rounds until node energy is exhausted and even HEED or LEACH networks do not consider the close proximity of the gateway or base station while select cluster heads [8–10]. Here, we have proposed a new protocol named reliable and efficient adaptation of cluster techniques (REACT) with an expectation of performing better than some available ones. In this protocol, clusters will be chosen on the basis of the objective function of the available residual energies of the sensors and average distance of neighbor sensor nodes.

Let us assume that the sensor nodes are positioned as (x_i, y_j) . While selecting cluster heads (C_i^H) from among the member nodes, the node forms following formula by estimating its distance d_j from neighbor cluster nodes.

$$\text{Minimized, } d_j = \alpha \times \mu + \beta \times \sigma, \text{ subject to, } \alpha, \beta \geq \frac{1}{2}\alpha + \beta = 1$$

Here, μ and δ are mean and standard deviations, respectively, from node i . The nodes will be chosen whose average distance from other nodes and comparatively less standard deviations of distances; the less the standard deviations, the balance among distances. On the other hand, the less the mean, the node is better to choose as cluster head. For example, if i_1 has some neighbor cluster-head nodes, $j_1, j_2, j_3, \dots, j_{n-1}, j_n$ and distances from node i_1 are $d_j^1, d_j^2, d_j^3, \dots, d_j^{n-1}, d_j^n$ and r_i is the residual energy of node i , then the average distance and standard deviations are as follows:

$$\text{Average Distance, } \mu = \frac{\sum_{k=1}^n d_j^k}{n}, \text{ and Standard Deviation, } \sigma = \sqrt{\frac{1}{n} \sum_{k=1}^n d_j^k - \mu}$$

Then the objective function (S_i) will be formed considering the residual energy (r_i), distances, and reputation/trust (t_i) for node i .

$$\begin{aligned} \text{Minimized } S_i = & 2 \times x_1 \times r_i \\ & + x_2 \times \frac{1}{d_i} + x_3 \times t_i, \quad \text{when } x_1, x_3 \geq \Upsilon_{1,3}, x_2 \geq \Upsilon_2 \\ & + x_3 \times t_i, \quad \text{when } x_1, x_3 \geq \Upsilon_{1,3}, \\ & x_2 \geq \Upsilon_2 \text{ and } x_1 + x_2 + x_3 \leq \Upsilon_{1,2,3} \end{aligned}$$

Similarly, the objective function (S_{i+1}) for node $i + 1$ can be defined as below:

$$\text{Minimized } S_{i+1} = 2 \times x_1 \times r_{i+1}$$

$$+ x_2 \times d_{i+1} + x_3 \times t_i, \quad \text{when } x_1, x_3 \geq \Upsilon_{1,3}, x_2 \geq \Upsilon_2 \text{ and } x_1 + x_2 + x_3 \leq \Upsilon_{1,2,3}$$

Here, $\Upsilon_{1,3}$, Υ_2 , $\Upsilon_{1,2,3}$ are constant or threshold values. The cluster node with highest objective function value will be nominated as cluster head (C^H).

$$\therefore \text{Cluster head, } C^H = \max(S_1, S_2, \dots, S_i, S_{i+1})$$

Algorithm 1: CHS – Cluster Head Selection

Result: Cluster Head (C^H)
Input : Sensor Position Value, $d_{i,j}(x_i, y_j)$, Residual Energy, r_s
Output: Objective Function, S_i

```

1   $x \leftarrow x_0$ 
2   $y \leftarrow y_0$ 
3  while  $CHS = true$  do
4    foreach node  $i = 1$  to  $m$  do
5      foreach node  $j = 1$  to  $n$  do
6         $| d_j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ 
7      end
8      For  $\alpha, \beta \geq \frac{1}{2}$ ,  $\alpha + \beta = 1$ 
9      Minimized,  $d_i = \alpha \times \mu + \beta \times \sigma$ 
10     Average Distance,  $\mu = \frac{\sum_{k=1}^n d_j^k}{n}$ 
11     Standard Deviation,  $\sigma = \sqrt{\frac{1}{n} \sum_{k=1}^n d_j^k - \mu}$ 
12     Minimized  $d_{i,j} = \mu \times x_1 + \sigma \times x_2$ 
13     Minimized  $S_i = 2 \times x_1 \times r_i + x_2 \times \frac{1}{d_i} + x_3 \times t_i$ 
14     when  $x_1, x_3 \geq \Upsilon_{1,3}$ ,  $x_2 \geq \Upsilon_2$ 
15     and  $x_1 + x_2 + X_3 \leq \Upsilon_{1,2,3}$ 
16   end
17   Cluster Head ( $C^H$ ) =  $\max(S_1, S_2, \dots, S_i, S_{i+1})$ 
18 end

```

The cluster head, C^H , can be selected using algorithms developed above. The cluster head collects data/information from all respective cluster members and aggregated data transported to the neighboring cluster head. This necessitates to develop another algorithm which will generate routing path within the sensor network domain.

3.2 Energy-Balanced Routing in Low-Power Sensor Networks for Operational Performance Improvement

Now, in order to select next cluster head by the source cluster node and repeating the same procedure to finally transport data/information packets to the gateway node or the base station, we shall develop an algorithm utilizing \mathcal{A}^* search methods. For this, we shall use the following equation:

$$f(n) = g(n) + h(n) \quad (1)$$

where n is the next node on the path, $g(n)$ is the cost of the path from the start node to n , and $h(n))$ is a heuristic function that estimates the cost of the cheapest path from n to the goal.

Now, in our case, let us express $g(n)$ as a function of both residual energy and distance between the source cluster head and the next forwarding cluster head. Hence, $g(n)$ can be defined as, $g(n) = E(r)E(d)$, where $E(r) = \frac{r}{R}$ and $E(d) = \frac{d}{D}$.

Packets will be forwarded by a node if its residual energy is greater than threshold residual energy, which results in the probability of accepting/rejecting of packet that is 1/2. On the other hand, a forwarding cluster head might be selected if it is within the range of source cluster head; therefore, the forwarding cluster-head node selection probability also is 1/2.

Now, if we consider n as probable forwarding cluster head currently and packets are already transmitted by previous nodes $1, 2, 3, \dots, (n - 1)$, the expected probability will become as follows:

$$E(r) = E(r_0) \times \frac{1}{2} + E(r_1) \times \frac{1}{2} + E(r_3) \times \frac{1}{2} + \dots E(r_{n-1}) \times \frac{1}{2} + E(r_n) \times \frac{1}{2}.$$

and,

$$E(d) = E(d_0) \times \frac{1}{2} + E(d_1) \times \frac{1}{2} + E(d_3) \times \frac{1}{2} + \dots E(d_{n-1}) \times \frac{1}{2} + E(d_n) \times \frac{1}{2}.$$

Next step, let us consider the heuristic function, $h(n) = \frac{1}{1+e^{-d}}$, where d is the heuristic distance from the probable forwarding cluster-head node to the gateway or the base station.

Finally, considering i be the source cluster head and some probable forwarding cluster-head nodes are, $j_1, j_2, j_3, \dots, j_{n-1}, j_n$. The most possible next forwarding cluster-head node will be found as follows:

$$\mathcal{F}_{\text{next}} = \max(f_1, f_2, \dots, f_i, f_{i+1}).$$

Algorithm 2: Routing Path Generation

Result: Optimal Routing Path

Input : Residual Energy, r_s , Distance Between Cluster Head, d_{ij}

Output: Forwarding Cluster Head, C^H

```

1 while PacketForwardingStatus = true do
2   foreach neighbor j = 1 to k do
3     | g(n) = ½[(sumi(E(ri) + E(di)) × (E(rj) + E(dj))]
4     | where, sender node is i
5     | h(n) = 1/(1+e-d)
6     | f(n) = g(n) + h(n)
7   end
8   | Fnext = max(f1, f2, ..., fi, fi+1)
9 end

```

Figure 3 is showing the logical routing layout. Let us assume that C_3^H is the current source, which needs to send data packets to one of the neighbor cluster-head

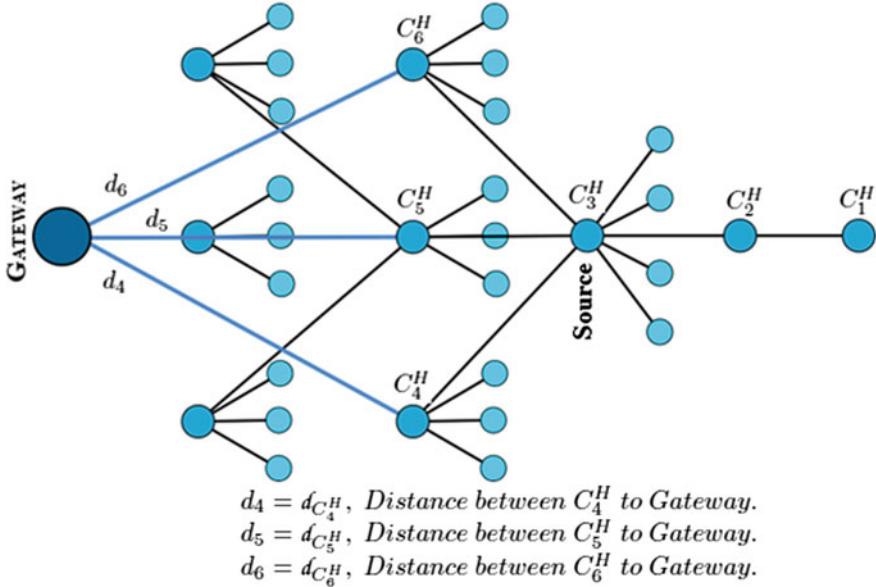


Fig. 3 Energy-balanced cluster-based route tree

node (it has been considered that C_1^H and C_2^H already transported data packets to C_3^H). Cluster heads C_4^H , C_5^H , and C_6^H , respectively, are neighbors of C_3^H , which will choose of those three neighbors to forward traffic using \mathcal{A}^* search algorithm.

Now C_3^H will estimate $g(C_4^H)$, $g(C_5^H)$, and $g(C_6^H)$, which is the summation of objective function from C_1^H to C_4^H , C_1^H to C_5^H and C_1^H to C_6^H , respectively.

The current source cluster head, C_3^H calculates, $g(C_4^H) = (\frac{1}{2}\mathcal{E}(r_{C_1^H}) + (\frac{1}{2}\mathcal{E}(r_{C_2^H}) + (\frac{1}{2}\mathcal{E}(r_{C_3^H}) + (\frac{1}{2}\mathcal{E}(r_{C_4^H})) \times (\frac{1}{2}\mathcal{E}(d_{C_1^H} + \frac{1}{2}\mathcal{E}(d_{C_2^H}) + \frac{1}{2}\mathcal{E}(d_{C_3^H}) + \frac{1}{2}\mathcal{E}(d_{C_4^H}))$, and

$g(C_5^H) = (\frac{1}{2}\mathcal{E}(r_{C_1^H}) + (\frac{1}{2}\mathcal{E}(r_{C_2^H}) + (\frac{1}{2}\mathcal{E}(r_{C_3^H}) + (\frac{1}{2}\mathcal{E}(r_{C_5^H})) \times (\frac{1}{2}\mathcal{E}(d_{C_1^H} + \frac{1}{2}\mathcal{E}(d_{C_2^H}) + \frac{1}{2}\mathcal{E}(d_{C_3^H}) + \frac{1}{2}\mathcal{E}(d_{C_5^H}))$.

Similarly, $g(C_6^H)$ will be calculated.

Finally, next forwarding cluster nodes will be calculated as,
 $f(C_4^H) = g(C_4^H) + h(C_4^H)$, where $h(C_4^H) = \frac{1}{1+e^{-x}}$. Similarly, estimation for $f(C_5^H)$ and $f(C_6^H)$ will be executed.

$\therefore F_{\text{Next}} = \max(f(C_4^H), f(C_5^H), f(C_6^H))$, which is the next forwarding cluster-head node.

3.3 Performance Analysis of the Proposed System for Operational Performance Improvement

The concept and the algorithm simulated in MATLAB. The simulation area spanned about 100×100 square fields with a single gateway/base station and N number of sensor nodes arbitrarily distributed in this area. Using the identical topology, around 50–60 experiments were done to establish result of each of the discussed experiments. The simulation parameters were chosen from [11–14] as per Table 1.

The simulation results compared performance of the proposed routing protocol REACT with two other protocols HEED and LEACH. The experiment performs the comparison with respect to (1) throughput, (2) network lifetime, and (3) energy remaining.

The network throughput is the utilization of network bandwidth within a specified time period; in this case, number of packets sent per second in certain period decided the throughput. In the proposed protocol REACT, by considering the balanced distance and the distance from the source to the destination cluster head decides how many packets will be transported per second. The comparison has been shown in Fig. 4, and this shows the throughput of the proposed protocol REACT is better than those other protocols used for comparison.

Next, the network lifetime of REACT protocol has been compared with HEED and LEACH in Fig. 5. REACT shows its lifetime higher in comparison of others in the experiment; the reason is more weight on residual energy assigned than that of chosen another parameter during cluster-head selection process. This results in richer energy of the proposed protocol, which facilitates to improve its network lifetime.

Table 1 Simulation parameter properties and values

Parameters	Values	Units
Initial node energy	3	Joules
Node electronics energy, E_{elec}	50	nJ/bit
Consumption loss for d^2 , ε_{fs}	10	nJ/bit/m ²
Consumption loss for d^4 , ε_{amp}	0.0013	pJ/bit/m ⁴
Optimum sensing radius, r_s	15	m
Data packet size, l	256	Bytes
Broadcast packet size, \tilde{N}_{bcast}	10	Bytes
Threshold distance, d_0	80	m
Cluster radius, r_{CH}	40	m
Sensor threshold energy, $\varepsilon_{threshold}$	0.1	Joules
Data cycles per round, L	5	—
Node numbers	(50, 100)	—
Sink position	100, 500	—
Network field	(0,0) (100,100)	—

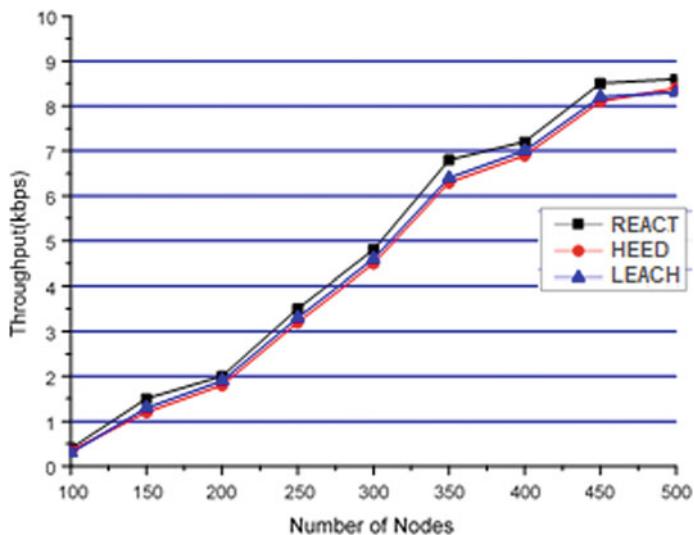


Fig. 4 Comparison of sensor network throughput

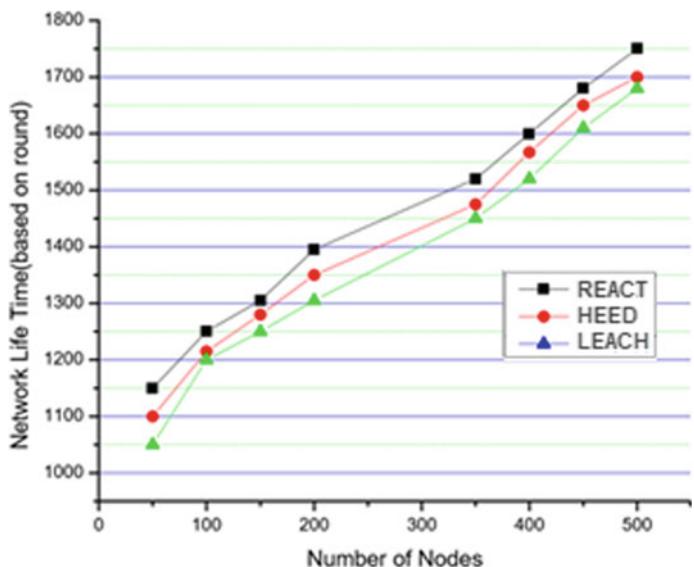


Fig. 5 Comparison of sensor network lifetime

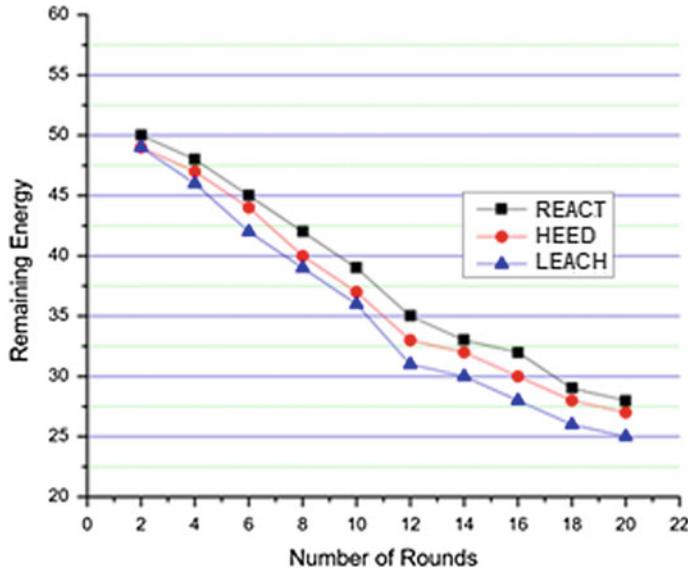


Fig. 6 Comparison of sensor network energy remaining

Then the cluster head of higher network lifetime dissipates less energy. Figure 6 shows the comparison of energy remaining of all three routing protocols; this figure depicts that the proposed REACT protocol balances the energy consumption. Therefore, the proposed protocol, REACT, has saved higher amount of energy compared to others in this experiment.

4 Security Improvement of IIoT Applications

Due to automation, different component objects of Industrial-IoT collect, analyze, and interpret real-time information/data for instant and intelligent decisions; these activities will be performed without human intervention [15]. Figure 1 integrates a digital power generation processes, where all the physical components are supposed to perform digital activities. Therefore, it is necessary to model a physical system and develop a control system to handle such automatic activities. This section will develop a physical control system and address how to stabilize possible control vulnerabilities of unexpected input; this might help improving security of IIoT applications.

4.1 System Modeling to Control Physical Systems in IIoT

Industrial-IoT uses industrial control systems, which collect information and transport for instant intelligent decision making, but there could be possibilities of losing such important information, due to lack of security systems [16]. Therefore, it is vital to develop a model for the control systems to address the security issues. Enormous advances of IIoT occurs such as industrial big data, cloud computing, and industrial wireless sensor networks; due to evolution of these techniques, industrial systems and intelligent devices are being supported to control industrial processes to meet the dynamic management and ubiquitous accessibility [17].

A system is stable when a storage function exists and the stored system energy is bounded by the supplied energy to the system [17]. The powerful tool for system analysis and control system design is the traditional passive systems theory [17]. Here, a system Σ has been considered with the following definition:

$$\dot{\psi}(t) = f(\psi, u) \quad (2)$$

$$y = g(\psi, u) \quad (3)$$

where $\psi \in \Psi \subset \Re^n$ is state of the process, $u \in U \subset \Re^m$ is the input to the control system, and $y \in Y \subset \Re^p$ is the control system output. The system Σ becomes passive or stable if a storage function $W(\psi) \geq 0$ exists; therefore, $\forall t_1 \geq t_0 \geq 0, \psi(t_0) \in \Psi$ and $u \in U$:

$$W(\psi(t_1)) - W(\psi(t_0)) \leq \int_{t_0}^{t_1} u^T(\tau) d(\tau) \quad (4)$$

Alternatively, if $W(\psi)$ is differentiable, then the above Eq. (3) can be written as,

$$\dot{W}(\psi) \leq u^T(t)y(t), \quad \forall t \geq 0 \quad (5)$$

Here, $W(\psi)$ is the initial energy or the energy content of the system, and $w = u^T(t)y(t)$ is the power fed to the system.

Definition IV.1. If a storage function exists and the initial or storage energy in the system is bounded above by the energy fed to it, the system can be defined as passive. Therefore, in the passive system, the storage function W satisfies $W(0) = 0$, if the system is dissipative with supply rate $w(u, y)$ [18]. This means that, in a passive system, $u^T(t)y(t) \geq \dot{W}(\psi)$.

A passive system provides the fundamental and inherent safety in building system infrastructures that are insensitive to implementation uncertainties [19].

Definition IV.2. A passive system can be defined as lossless, i.e., stable, if $u^T(t)y(t) = \dot{W}(\psi)$. Therefore, the system is stable when $u^T(t)y(t) = \dot{W}(\psi)$.

4.2 Securing Industrial-IoT Applications

According to Borg [20], company executives and key researchers are moving into the crosshairs of the cyber-hackers worse than ever before (this has never happened in the past), as hackers are increasingly targeting industrial equipment, particularly focusing on hardware, i.e., process control, including programmable logic controllers and local networks; this could hurt the affected company by resulting in a drop in the stock price due to possible failure of quality control. The cyber-hackers could earn more money than from a credit card fraud and could even advantage them further by taking position in the stock market [20]. Therefore, the smart industrial control processes will be in the list of critical target of cyber-hackers. Hence, it is necessary to develop a feedback control system to protect such logical control attack vulnerabilities.

Now, in order to design a controller to disestablish the attack on the industrial control systems, this section will review the adaptive. The stabilized system control developed here is similar to Lyapunov system of stability, $V(x) = A^T P A$.

Hence, in order to disestablish the attack vulnerability (nonlinear portion of Eq. (5), we shall use Lyapunov equation $\alpha^T \beta + \beta \alpha + \Theta = 0$, to develop a condition that will stabilize a nonlinear system.

$$\begin{aligned} \alpha^T \beta + \beta \alpha + \Theta &= 0 \\ \text{or, } \alpha^T \beta + \beta \alpha &= -\Theta \\ \text{or, } X^T (\alpha^T \beta + \beta \alpha) X &= -X^T \Theta X \\ \text{or, } \alpha^T X^T \beta X + X^T \beta \alpha X &= -X^T \Theta X \end{aligned} \quad (6)$$

Hence, using the state-space representation of nonlinear closed-loop control system concept above, Eq. (6), can be written as,

$$\begin{aligned} \dot{X}^T \beta X + X^T \beta \dot{X} &= -X^T \Theta X < 0 \\ [\because \Theta \text{ always is a positive definite}] \\ \therefore \frac{\delta}{\delta X} (X^T \beta X) &= -X^T \Theta X \end{aligned} \quad (7)$$

Equation (7), if compared with Lyapunov function, $V(x) = x^T P x$ can be rewritten as,

$$\dot{\Psi} = -X^T \Theta X \quad (8)$$

This proposition has been derived using the closed-loop stable state-space equation $\dot{X} = AX$; therefore, $\dot{\Psi} = -X^T \Theta X$ is a Lyapunov condition that will stabilize a nonlinear system.

5 Cyber-Physical Communication Security and Performance Improvement Issues

The success behind the Internet is established and well-known standards, OSI and TCP/IP model, which guide communication of each Internet component in layered and synchronous manner. But in the cyber-physical communication, the Ethernet is exposed to be interacted with the cyber-system; this makes a critical challenge to secure the physical systems without and until a well-understood standard in place. In smart power industries, the cyber-communication technology and the function of the network between the main station and substation control systems are totally different [21]. With the Internet of Things (IoT) in place, security risks are increasing; systems will be more vulnerable when more physical infrastructure connected to the Internet.

5.1 *Performance Improvement Issues to Address*

One of the most important vulnerabilities is the network performance; this includes both faster data transport through the network media and the performance of networked equipment. According to [22], the IIoT platform and the wireless sensor devices require highly available and the requirement of availability needs to be 99.999% uptime to provide scalable solutions. But this could be very difficult to maintain, as currently available low-power wireless sensors cannot guarantee of uninterrupted power.

As mentioned in [23], the focus of Industrial-IoT is to control and transport mission critical real-time information that heavily relies on M2M communications; a delay or an error of this transmission might result in catastrophic issues in some industrial applications. In the industrial network applications, the communication network advances quite a lot, but still the internetwork framework has limitations due to fixed bandwidth, communication latencies, and coverage, which results in poor adaptability of current network framework in the emerging Industrial-IoT technology [24].

As the IIoT advantages the IR 4.0, this distributed evolution will enhance industrial processes like advanced manufacturing processes by integrating IoT, networked control systems, cloud computing, and cyber-communication technologies. This complexity has put IIoT in a great challenge as in the manufacturing industrial process, different system levels require different standards to be communicated with each other, which does not exist yet and therefore resulted in one of the main challenges toward its success [25]. Industrial-IoT is the combination of IoT and CPS; therefore, alike IoT and CPS, IIoT uses wireless sensor and actuator network (WSAN) for sensing and actuating. But due to the critical latency, wireless control systems face serious challenges to meet the latency requirements of the feedback control systems [26]. The maximum bandwidth of WSAN is only 250 kbps according to IEEE 802.15.4 and the communication delays further increased over the mesh networks

of multi-hop communication, and also the latency is impacted due to the external interference and weather conditions [26].

5.2 *Security Improvement Issues to Address*

With the IoT in place, these security risks are increasing; systems will be more vulnerable when more physical infrastructure is connected to the Internet. The Chief Security Officer of PTC, a Massachusetts-based software firm, Corman, Josh, raised his concern about the vulnerability of IoT due to there being more physical systems and facilities connected to wireless networks, which will be difficult to tackle with traditional IT security methods [27].

As indicated by the concept of the IoT, lack of standards and interoperability results in huge barrier of adopting IoT technology in the industrial network applications [28]. The communication difficulty due to the latency and security in existing network technologies and IIoT applications has fallen in obvious challenges [29].

In the Industrial-IoT infrastructure, intelligent systems and devices, which are associated with sensors and actuators, collect, analyze, process, and transport the processed data to the cloud server storage for taking decisions and future use; therefore, those data require to be accessed by authorized users that necessitates proper security for confidential use of those important information [30]. Any vulnerability of security will cause business loss to a great extent.

The adaptation of IoT by the industrial application processing, the great technology IIoT, and the industrial revolution (IR 4.0) evolved, but due to massive exploded advancement of IoT is becoming a great concern of handling massive data securely [31]. This may cause great issues in achieving the goal of the fastest grown distributed technologies IIoT and IR 4.0.

The industrial process and technologies require an acceptable standard, in order to maintain secure and reliable network for business-critical Industrial-IoT applications [32].

6 Conclusions and Future Works

From current research, it is clear that the success of Industry 4.0 critically depends on the network communication performance and security. But as the current Internet does not assure guarantee of communication and security, low-power sensors also do not perform to the expectation. These necessitate to improve the Industrial-IoT network communication performance and security. This project aimed at contributing toward the solution of these issues toward useful operation of Industry 4.0 which is being leveraged by IIoT. The proposed routing protocol in this purpose has produced some better results, and the system security models discussed might also contribute toward the solution of cybersecurity issues. If the cybersecurity and performance

improvement bottlenecks mentioned can be resolved, more better results could be achieved.

There were limitations of choosing simulation parameters accurately, which we have planned to address in our future works. In future, we might also consider some use cases of Industrial-IoT applications for investigation using approaches of this project; this might contribute more toward the success of Industry 4.0 or the IR 4.0 technology.

References

1. Boyes H, Hallaq B, Cunningham J, Watson T (2018) The industrial internet of things (IIoT): an analysis framework. *Comput Ind* 101:1–12. <https://doi.org/10.1016/j.compind.2018.04.015>. ISSN 0166-3615
2. Industry 4.0 and Machine Vision, <https://www.cognex.com/what-is/industry-4-0-machine-vision>. Retrieved 06 Mar 2019
3. Koulali M, Koulali S, Tembine H, Kobbane A (2018) Industrial internet of things-based prognostic health management: a mean-field stochastic game approach. *IEEE Access* 6:54388–54395
4. Industrial Internet of Things Platform, <https://www.kaaproject.org/industrial-automation>. Retrieved 20 Mar 2019
5. Phillips A (2019) The industrial internet of things. ARM: the architecture for the digital world. www.hvm-uk.com/uploads/sgcp14phillips.pdf. Retrieved 03 Jan 2019
6. Yongfu L, Dihua S, Weining L, Xuebo Z (2012) A service oriented architecture for the transportation cyber physical systems. In: 2012 31st Chinese control conference (CCC), pages 7674768
7. Farraj A, Hammad E, Kundur D (2018) A cyber-physical control framework for transient stability in smart grids. *IEEE Trans Smart Grid* 9(2):1205–1215
8. Heinzelman WB, Chandrakasan AP, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans Wirel Commun* 1(4):660–670. <https://doi.org/10.1109/TWC.2002.804190>
9. Amjad M, Afzal MK, Umer T, Kim B-S (2017) Qos-aware and heterogeneously clustered routing protocol for wireless sensor networks. *IEEE Access* 5:10250–10262
10. Sivathanan S (2009) RF/FSO and LEACH wireless sensor networks: A case study comparing network performance. In: 2009 IFIP international conference on wireless and optical communications networks, Cairo, pp 1–4. <https://doi.org/10.1109/wocn.2009.5010512>
11. Yuan XH, Elhoseny M, El-Minir HK, Riad AM (2017) A genetic algorithm based, dynamic clustering method towards improved WSN longevity. *J Netw Syst Manag* 25:21–46
12. Liu X, He D (2014) Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks. *J Netw Comput Appl* 39:310–318
13. Taheri H, Neamatollahi P, Younis OM, Naghibzadeh S, Yaghmaee MH (2012) An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic. *Ad Hoc Netw* 10:1469–1481
14. Luna-Vazquez I (2006) Implementation and simulation of routing protocols for wireless sensor networks
15. Das AK, Wazid M, Kumar N, Vasilakos AV, Rodrigues JJPC (2018) Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment. *IEEE Internet Things J*
16. Choo KR, Gritzalis S, Park JH (2018) Cryptographic solutions for industrial internet-of-things: research challenges and opportunities. *IEEE Trans Ind Inf* 14(8):3567–3569

17. Qiu C, Yu FR, Yao H, Jiang C, Xu F, Zhao C (2018) Blockchain-based software-defined industrial internet of things: a dueling deep Q-learning approach. *IEEE Internet Things J*
18. Sen S, Pang P (2018) Architectural modeling and cybersecurity analysis of cyber-physical systems—a technical review. *Int Res J Eng Technol* 5
19. Ortega R, Spong MW (1988) Adaptive motion control of rigid robots: a tutorial. In: Proceedings of the 27th IEEE conference on decision and control, pp 1575–1584
20. Borg S (Director, U.S., Cyber Consequences Unit) (2017) To design better hardware, think like a cyber-criminal. At the MEMS and sensors technical congress held at Stanford University, California, USA, to an audience of 130 Chief Technical Officers, Engineering Directors and Key Researchers, IEEE Spectrum, Page 22
21. Liu W, Gong Q, Han H, Wang Z, Wang L (2018) Reliability modeling and evaluation of active cyber physical distribution system. *IEEE Trans Power Syst* 33(6):7096–7108
22. Miller D (2018) Blockchain and the internet of things in the industrial sector. *IT Prof* 20(3):15–18
23. Aazam M, Zeadally S, Harras KA (2018) Deploying fog computing in industrial internet of things and industry 4.0. *IEEE Trans Ind Inf* 14(10):4674–4682
24. Li X, Li D, Wan J, Liu C, Imran M (2018) Adaptive transmission optimization in SDN-based industrial internet of things with edge computing. *IEEE Internet Things J* 5(3):1351–1360
25. Nguyen N, Leu MC, Liu XF (2017) Real-time communication for manufacturing cyber-physical systems. In: 2017 IEEE 16th international symposium on network computing and applications (NCA). Cambridge, pp 1–4
26. Lu C et al (2016) Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proc IEEE* 104(5):1013–1024
27. Higginsbotham S (2018) Internet of everything: 6 ways IoT is vulnerable. *IEEE Spectrum*, p 21
28. Divya Darshini B, Paventhan A, Krishna H, Pahuja N (2016) Enabling real time requirements in industrial IoT through IETF 6TiSCH. In: 2016 international conference on internet of things and applications (IOTA), Pune, pp 121–124
29. Duan Y, Li W, Zhong Y, Fu X (2016) A multi-network control framework based on industrial internet of things. In: 2016 IEEE 13th international conference on networking, sensing, and control (ICNSC), Mexico City, pp 1–5
30. Karati A, Islam SH, Biswas GP, Bhuiyan MZA, Vijayakumar P, Karuppiah M (2018) Provably secure identity-based signcryption scheme for crowdsourced industrial internet of things environments. *IEEE Internet Things J* 5(4):2904–2914
31. Bloom G, Alsulami B, Nwafor E, Bertolotti IC (2018) Design patterns for the industrial internet of things. In: 2018 14th IEEE international workshop on factory communication systems (WFCS), Imperia, pp 1–10
32. Nwadiugwu WP, Kim D (2018) Energy-efficient sensors in data centers for industrial internet of things (IIoT). In: 2018 3rd international conference on internet of things: smart innovation and usages (IoT-SIU), Bhimtal, pp 1–6

Chapter 28

Dynamic Load Balancing in Software-Defined Networks Using Machine Learning



Kunal Rupani, Nikhil Punjabi, Mohnish Shamdasani and Sheetal Chaudhari

1 Introduction

In the SDN architecture, the data plane and the control plane are separated. The SDN architecture facilitates network virtualization. It is directly programmable, scalable and allows integration of services like load balancing, Firewall and Intrusion Detection System (IDS). The SDN architecture consists of three layers namely the application layer, the control layer and the data layer. The application layer consists of applications like load balancer, traffic monitoring, etc. The control layer is where the SDN controller resides. The data layer is responsible for forwarding the data from source to destination. In SDN, there has been limited research proposed on network load balancing. In previously proposed projects, the controller gets information from OpenFlow switches to analyze load on each link and accordingly modify the flow-tables by using a particular load balancing strategy. Since a routing plan which is dynamic and a static load balancing method is proposed in these strategies, these strategies fail to take advantage of SDN and do not make an efficient load balancing model. These algorithms work on SDN having multiple paths, but the strategy for routing is determined by considering the load condition of next-hop only while the property of global view in SDN is not leveraged. Therefore, such kind of strategies cannot determine the effective path in real time and hence they cannot achieve an ideal load balancing effect. To summarize, current research in load balancing in SDN reveals that the existing algorithms are too simple for a complex problem like load balancing and so they exhibit poor performance. It is also understood that in these algorithms data gathering is not done to the fullest due to which the desired accuracy is not achieved. The objective of the paper is to propose a system that load balances an SDN-based network in real time in order to make data transmission in SDN more efficient and reliable.

K. Rupani (✉) · N. Punjabi · M. Shamdasani · S. Chaudhari
Sardar Patel Institute of Technology, Mumbai, India
e-mail: krupani8@gmail.com

2 Related Work

In [2], the back-propagation artificial neural network is trained to find the integrated load condition of the network. Flow rules indicating the best path are then pushed into the switches by the controller. However, this technique lacks performance and does not consider node utilization as a part of load balancing a network. In [3], the algorithm can balance link load in a network quickly to resolve some congested path. Also minimized packet loss is achieved by changing paths of flows. But this technique consumes more time and does not consider node utilization. In [4], using the fuzzy synthetic evaluation model (FSEM) the paths can be dynamically adjusted by taking advantage of the global view of the network. However, this technique is not reliable as there is packet loss due to the time taken to detect link failure. In [5], as the load balancing algorithm keeps running, its performance improves over time but its initial performance is found to be low. The algorithm first finds the shortest path and then checks for link utilization. In [6], the fuzzy synthetic evaluation algorithm with dynamic weight (FSEADW) is used which supports dynamic weights to dynamically realize network status in real time to achieve better load balancing. However, this technique too ignores the overall network utilization as it does not consider node utilization.

3 Proposed System

The proposed system consists of two subsystems namely the simulation subsystem and the machine learning subsystem. The two subsystems are interconnected using the Django REST Framework using which path features are sent over from the simulation subsystem to the machine learning subsystem in JSON format using the HTTP protocol. Figure 1 shows the network architecture diagram for the proposed SDN load balancing system. The proposed system has minimal processor and memory requirements. Sending data in the form of JSON is again efficient as it requires less bandwidth.

3.1 *Simulation*

Mininet simulation software is used to simulate a DCN-based fat-tree topology network containing 8 hosts and then extended for network containing 16 hosts. The topology with 8 hosts is described. The hosts are labeled from h1 to h8. They are connected using switches which are arranged in the form of a tree structure. The topologies are created beforehand in python using the NetworkX library [7]. The Mininet simulator makes use of Floodlight Controller which acts as the SDN controller. The Mininet software provides support for python in the form of APIs. Using

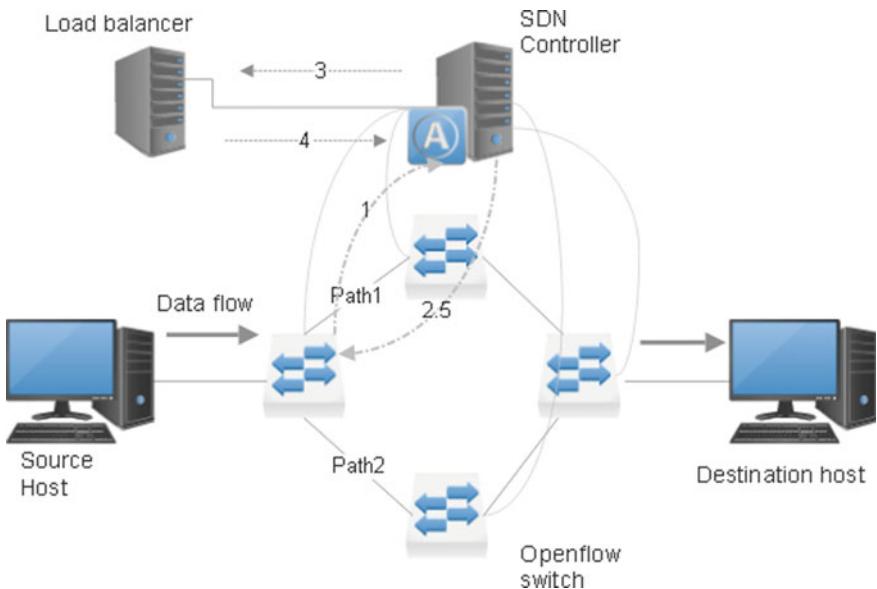


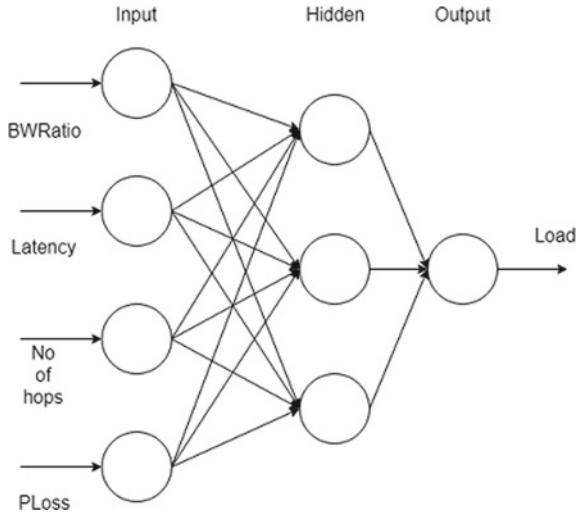
Fig. 1 Network architecture for proposed system

these predefined APIs, path features are extracted for the machine learning subsystem. Wireshark [8] is used to monitor the traffic while Mininet is running in parallel.

3.2 Artificial Neural Network

This subsystem is used to find the minimum loaded path between a source node and destination node in real time. Sequential model from keras library is used to train the dataset obtained from the simulation subsystem. The structure of the back-propagation artificial neural network is shown in Fig. 2. The predictors, i.e., the inputs are BW Ratio, latency, packet loss rate, the number of hops and node utilization from source to destination. These inputs form the input layer. Figure 2 shows that the hidden layer has three neurons. This number is varied from three to eleven using the formula given in (1) specified in [2] and the most accurate model is chosen. In (1), N is the number of neurons in the hidden layer, the number of neurons in the input layer is denoted by m , the number of neurons in the output layer is denoted by n and a is a constant between 1 and 10. The output layer gives the integrated load on the input path depending upon the path features. The activation function used is the popular ReLU activation function as it overcomes the problem of gradient descent. The learning rate is also adjusted to get an accurate model.

Fig. 2 Artificial neural network structure with three neurons in the hidden layer



The number of epochs value is set to 100 so that the mean squared error at the end of training is very close to zero. Also, initially, random weights are assigned. Setting the network parameters correctly is a must to get the desired accuracy of the model.

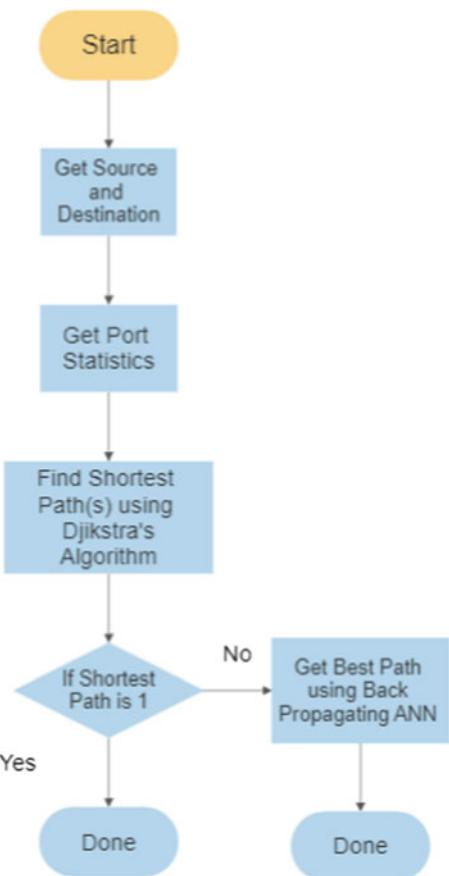
$$N^2 = m + n + a \quad (1)$$

4 Methodology

SDN controller has the property of global view of network that it possesses at any stage of the simulation. All the paths from one node to every other node are found by updating the topology information of the network. By leveraging the property of global view in SDN, the effective load condition of every path can be determined easily. The System Flow Diagram is shown in Fig. 3.

4.1 Network Simulation

Network simulation is first done for data collection and then for final testing. For testing the system, running the Floodlight Controller is the first step of network simulation. Running the topology using Mininet is the next step. Then the load balancer script is executed. Here, the source host is entered by the user. Then the least loaded host is selected as the destination. The selected hosts are activated and

Fig. 3 System flow diagram

traffic is generated between them. Wireshark [8] is used to monitor the traffic between these hosts. Port statistics are obtained using the in-built APIs of Mininet.

Then using Dijkstra's algorithm, shortest path(s) is/are determined between the source and destination. If multiple shortest paths are found, then the machine learning module comes into picture otherwise the shortest path obtained becomes the best path itself. To overcome link failure, the first step is to detect link failure and the other step is to choose an alternate path. Link failure is detected by the SDN controller when a switch is unable to send a packet-in message to the controller within a predefined timeout period [9]. The SDN controller then informs the load balancer about the link failure. The load balancer temporarily removes all the paths containing the failed link and finds the least loaded path as usual. Node failure is overcome by detecting the node failure and finding the backup path while the SDN controller performs failure recovery as specified in [10]. Node failure is detected by the SDN controller when the node stops sending packet-in message to the controller within a predefined timeout

period [9]. The load balancer utilizes the remaining backup paths for finding the best path between the source host and the destination host.

4.2 Data Collection and Preprocessing

The training dataset is obtained by running the Mininet simulator repeatedly after an interval of three minutes and recording all the path information using the APIs. The port statistics obtained are then used to calculate the path features using the formulae shown in the equation below. This method ensures that there is variety in the dataset which is usually the case in real-world networks. Such variety also ensures that the model is accurate. Such collection of data is only possible because of the global view architecture of SDN. This training data collected from the simulation subsystem is stored in the form of an excel sheet. This training dataset is first loaded into the program and then examined for errors. After cleaning, the data comes scaling of data after which the training data is ready. The testing data is obtained in real time in JSON format using Django. This data is converted into proper format after which it is ready for testing.

1. BWRatio: BWRatio is the subtraction of cumulative transmitted bytes $B(T)$ and previous cumulative transmitted bytes $B(T - 1)$ divided by the maximum bandwidth BWRatio(MAX).

$$\text{BWRatio} = B(T) - B(T - 1)/\text{BWRatio(MAX)} \quad (2)$$

2. Packet Loss Rate (P Loss): It is the ratio of the number of packets not received and the number of packets transmitted. Packet(T) is the number of packets transmitted while Packet(R) is the number of packets received.

$$P \text{ Loss} = \text{Packet}(T) - \text{Packet}(R)/\text{Packet}(T) \quad (3)$$

3. Transmission Latency: It is the ratio of bytes transmitted Byte and the transmission rate (trRate).

$$\text{Latency} = \text{Byte}/\text{trRate} \quad (4)$$

4. Total Node Utilization: It is the sum of node utilization values of all the switches along the path.
5. Transmission hops: This value is directly obtained by using the predefined API for hops.

4.3 Training and Testing Neural Network Model

The back-propagation artificial neural network is trained on the training data for 100 epochs to get the desired accuracy. Training is hardly time-consuming as the dataset has about 600 records collected at different times and the training time of a real-time system is less valuable than the testing time, i.e., the predicting time. The testing time needs to be negligible which is the case here. The model is evaluated with ‘Mean Squared Error’ as the loss function while ‘Mean Absolute Error’ is used as a metric to test the accuracy of the model. Now the shortest paths with their features are used to predict the integrated load on each one of them.

5 Results

5.1 Training Results of BPANN

The result after training the model is a graph of mean absolute error versus the number of epochs. The graph in Fig. 4 is for topology with 8 hosts. In this graph, the mean absolute error at the end of training the model with $n = 7$ is found to be closest to zero which means that the model is most accurate when $n = 7$. So, the value of n is set to 7. The graph in Fig. 5 is for topology with 16 hosts. In this graph, the mean absolute error at the end of training the model with $n = 9$ is found to be closest to

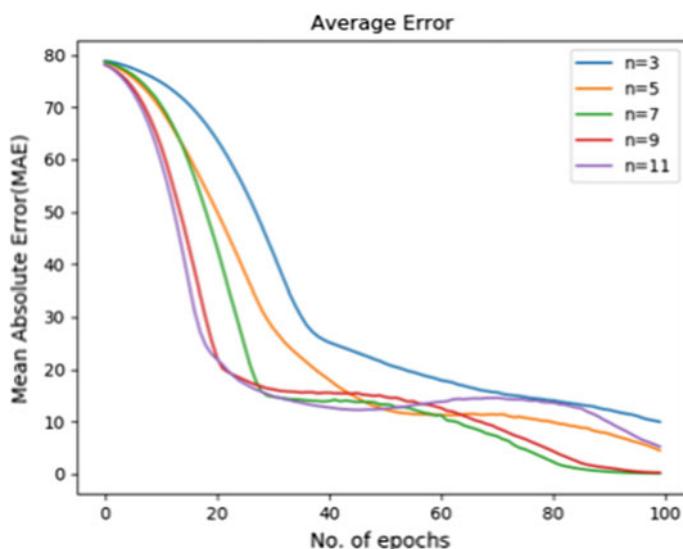


Fig. 4 Mean absolute error for topology with 8 hosts

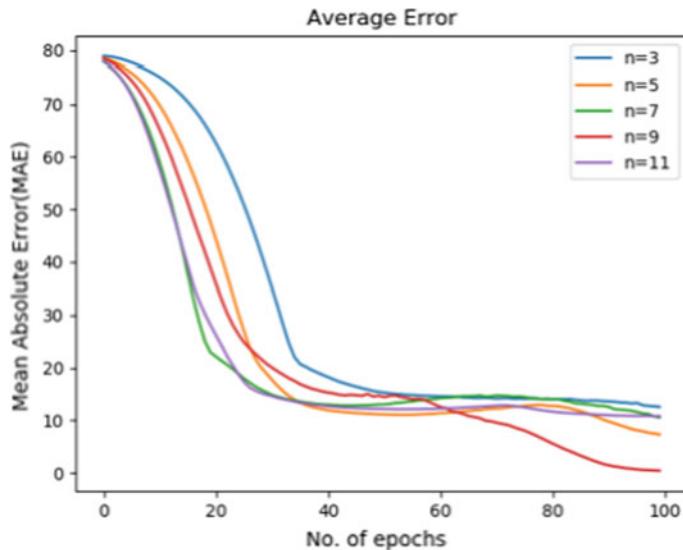


Fig. 5 Mean absolute error for topology with 16 hosts

zero which means that the model is most accurate when $n = 9$. So, the value of n is set to 9.

5.2 Comparing Latency Before and After Load Balancing

The iperf tool [11] is used to get the network statistics for analysis. The iperf command gives the throughput while the ping command is used to measure the latency. Figure 6 depicts the final result in the form of latency and throughput recorded before and after load balancing. In Table 1, the average latency between h1 and h8 decreases and the throughput increases. In Table 2, the average latency between h1 and h4 decreases and the throughput increases. Similar results are noted for a different pair of hosts in both topologies. This means that the load balancing is working.

6 Conclusion

Dynamic load balancing is achieved by predicting the effective load on all the shortest paths and selecting the path with minimum load in real time. The output of the system is nothing but this least loaded path. The model is trained successfully on topologies with 8 and 16 hosts. This means that the proposed system is scalable. The system can also handle link failure and node failure. This means that the system is reliable. The

Table 1: Results for Topology with 16 hosts recorded between h1 and h8			
Before Load Balancing		After Load Balancing	
Average Latency (ms)	Throughput (Gbits/sec)	Average Latency (ms)	Throughput (Gbits/sec)
0.2	3	0.15	3.67

Table 2: Results for Topology with 8 hosts recorded between h1 and h4			
Before Load Balancing		After Load Balancing	
Average Latency (ms)	Throughput (Gbits/sec)	Latency (ms)	Throughput (Gbits/sec)
0.159	3.96	0.152	3.98

Fig. 6 Result tables

proposed system can be used as a load balancing module in an SDN-based system to improve its performance.

References

1. <http://mininet.org/>
2. Chen-xiao C, Ya-bin X (2016) Research on load balance method in SDN. Int J Grid Distrib Comput 9(1):25–36. <http://dx.doi.org/10.14257/ijgdc.2016.9.1.03>
3. Lan Y-L, Wang K, Hsu Y-H (2016) Dynamic load-balanced path optimization in SDN-based data center networks. In: 2016 10th international symposium on communication systems, networks and digital signal processing (CSNDSP). <https://doi.org/10.1109/csnsp.2016.7573945>
4. Li J, Chang X, Ren Y, Zhang Z, Wang G (2014) An effective path load balancing mechanism based on SDN. In: 2014 IEEE 13th international conference on trust, security and privacy in computing and communications. <https://doi.org/10.1109/trustcom.2014.67>
5. Zakia U, Ben Yedder H (2017) Dynamic load balancing in SDN-based data center networks. In: 2017 8th IEEE annual information technology, electronics and mobile communication conference (IEMCON). <https://doi.org/10.1109/iemcon.2017.8117206>
6. Wang T, Guo X, Song M, Peng Y (2017) A fuzzy synthetic evaluation algorithm with dynamic weight for SDN. In: 2017 IEEE 2nd information technology, networking, electronic and automation control conference (ITNEC). <https://doi.org/10.1109/itnec.2017.8284896>
7. <https://networkx.github.io/>
8. <https://www.wireshark.org/>
9. Xu H, Yan L, Xing H, Cui Y, Li S (2017) Link failure detection in software defined networks: an active feedback mechanism. Electron Lett 53(11):722724. <https://doi.org/10.1049/el.2017.082>

10. Zhang S, Wang Y, He Q, Yu J, Guo S (2016) Backup-resource based failure recovery approach in SDN data plane. In: 2016 18th Asia-Pacific network operations and management symposium (APNOMS). <https://doi.org/10.1109/apnoms.2016.7737211>
11. <https://iperf.fr/>
12. <https://keras.io/getting-started/sequential-model-guide/>

Part IV

**Design and Application of Intelligent
Computing and Communication**

Chapter 29

Analysis and Comparison of Timbral Audio Descriptors with Traditional Audio Descriptors Used in Automatic Tabla Bol Identification of North Indian Classical Music



Shambhavi Shete and Saurabh Deshmukh

1 Introduction

Sound is a physical phenomenon of vibration that passes through air and reaches to human ear. Unstructured form of sound is called as noise, e.g., sounds of winds, trees, rivers, etc. Structured form of sound is called as music, which is pleasant to human ear. Music is a sole of life. Nowadays, music has made our life more comfortable. Therefore, it is consumed and produced at very high level. Music information retrieval (MIR) is sub-branch of sound information retrieval (SIR) that has variety of engineering applications in music industry.

Hindustani (North Indian) and Carnatic (South Indian) music are two classical music traditions in India. North Indian Classical Music (NICM) is one of the ancient and very rich forms of music, in terms of culture and technicality of performing arts that has yet not been technically explored to its full extent. NICM performances are vocal and instrumental. There are number of vocal forms like Ghazal, Khayal, Thumri, Dhrupad, Dhamar, Tarana, Bhajan and Dadara, etc. The performances of NICM are based on singer's and player's music compositions. These are based on some meter and produce many variations in it.

In North Indian Classical Music (NICM), Tabla is one of the important accompanying rhythm instruments. The Tabla instrument is used in Indian Classical Music in two ways. Either Tabla is played as solo musical (Rhythm) instrument, where an accompanying Tanpura and Rhythm keeping melodic instrument, such as Sarangi or Harmonium is used. The purpose of this accompanying musical instrument along

S. Shete (✉) · S. Deshmukh
CSE Department, MIT, Aurangabad, Maharashtra, India
e-mail: shambhavishete@gmail.com

S. Deshmukh
e-mail: saurabh.h.deshmukh@gmail.com

with Tabla solo performance is two folded. First, it gives pleasure and melodic bearing experience based on the raga and the musical notes being played repeatedly. Another use of these accompanying musical instruments is to keep track of the tempo of the Tabla solo performance.

Another way in which a Tabla instrument used is to give tempo and rhythm accompaniment to solo performance of another musical instrument or vocal, typically reciting Indian Classical Music Raga. The importance and usage of Tabla instrument are different for both performances. For the Tabla players and to the learners of the Tabla instrument, it is very important to know the various ways in which the Tabla is played.

Automatic Tabla Bol Identification is one of such useful applications, where the Tabla Bolts are identified. There exist many Tabla Bolts that are played during the performance. To identify the Tabla Bolt automatically, the presence of Tabla Bolt in the audio excerpt is to be located. In this paper, we have analyzed and compared various Timbral audio descriptors with traditional Mel-frequency cepstral coefficient (MFCC), linear predictive cepstral coefficients (LPCC) and spectral audio features used for automatic identification of Tabla Bolt.

Typically, sound has pitch, loudness, duration and Timbre. Timbre is undefined, perceptual, multi-dimensional and non-tangible entity. Human can separate the difference between two sounds similarly presented and having same pitch, loudness and duration. Each sound has its own unique audio attributes. The attributes or characteristic features of the audio segment which are used for analysis of sound are called as Audio Descriptors. The most common audio descriptors are traditional (MFCC and LPCC), Timbral, temporal, spectral and perceptual.

Mel-frequency cepstral coefficient (MFCC) and linear predictive cepstral coefficients (LPCC) are most common traditional audio descriptors for automatic speech, musical instrument or speaker identification. MFCC is used to describe spectral shape of sound, while LPCC is used to provide economical source-filter model for human auditory system. MFC coefficients are computed on a wrapped frequency scale based on known human auditory perception system, whereas LPC coefficients represent the human articulators system based on linear prediction [1].

2 Literature Survey

In North Indian Classical Music (NICM), Tabla is most important percussion instrument in recent times. It is made up of two drums called Tabla (Dayan/Right Drum) and Daga (Bayan/Left Drum). Tabla is made up of a hollow tapered cylindrical block of wood played by right hand. Daga is inverted dome-shaped pot made of copper played by left hand. Tabla instrument is used to produce pitched and un-pitched tones, which in turn creates rhythm and melody. Tabla (Dayan) produces open or closed sound. Tabla has high pitch than Daga. The Tabla players use hand pressure to change pitch and tone of this Tabla and Daga. Dissimilar to western rhythm

instruments, the Tabla Bols are created with single, alternate and simultaneous hits by hands on both the drums [2].

The basic strokes that are used during Tabla performance are called as Tabla Bols. Some basic Bols of Tabla are Na, Ta, Ka, Ti, Tita, Kata, Ga, Dha, Dhin, etc. Tabla Bols are classified as (a) Bols played on Tabla only, (b) Bols played on Dagga Only and (c) Bols played on both Tabla and Dagga simultaneously [3]. These Bols are then sequenced together to form different rhythm patterns called as Talas [4]. Rhythm patterns are composed with note, half note, quarter note and so on. Talas can have large variety of beats [5]. Talas are played in slow (Vilambit), medium (Madhya) and fast (Drut) tempo. Tempo (Lay) is used to represent rhythmic information of a Tala [2, 6].

Tabla produces both monophonic and homophonic sound. When only one drum is played, it produces monophonic sound, while some Bols are produced using both the drums, which produces homophonic sound. The sound signal analysis of homophonic texture is comparatively complex and difficult to analyze. There exist different types of signal features that are distinguished according to the steadiness or dynamicity, time extent, abstractness and extraction process of the audio feature. Nowadays, a large set of features are available mainly, including temporal shape, temporal features, energy features, spectral shape features, harmonic features, perceptual features and low-level audio descriptors. There are various global and instantaneous audio features. First, the preprocessing must be done for the extraction of audio descriptors that gives adequate signal representation. The preprocessing includes estimation of energy envelop, short-term Fourier transform (STFT), sinusoidal harmonic modeling and a cascade of process (like human earring process) [7].

Loudness, duration, pitch and timber are important attributes for musical rhythm instruments also. Timber depends upon overtones generated by rhythm instruments, but change in frequency and intensity causes change in Timber. There exists a lot of research on musical instrument identification with the help of Timbral audio descriptors. Shimmer, jitter, spectral spread, spectral centroid, inharmonicity, attack time, harmonic slope, MFCC, harmonic expansion, spectral flux, temporal centroid, ZCR are some important Timbral features that are majorly used to identify musical instrument. For automatic musical instrument, Timber recognition above features is extracted and tested to achieve robust feature vectors [8].

Attack time, attack slope, ZCR, roll-off, brightness, MFCC, roughness, irregularity are some Timbral feature functions which are available in MIR Toolbox 1.5. It is used to detect temporal duration, noisiness, high-frequency energy, spectral shape and peaks of sound [9].

MFCC is used to describe the spectral shape of sound. The parameters used for MFCC are window frame length, step between successive window frames, number of cepstral coefficients and frequency wrapping. The Mel-frequency cepstral coefficients (MFCCs) of a signal are the values of energy that are considered between 10 and 20 set of features. These features describe the overall shape of a spectral envelope. It is used to describe Timbre. MFCC features are mainly used in automatic

speech recognition [10], speaker recognition [11], automatic segmentation and recognition of Tabla strokes [12], automatic labeling of Tabla strokes [6], automatic drum transcription [13] and Mridangam transcription [14].

The linear predictive cepstral coefficients (LPCCs) extract feature using energy values of linearly arranged filter banks. It provides economical source-filter model of human auditory system. The parameters used for LPCC are hamming window frame, frames step size, number of cepstral coefficients and the frequency wrapping.

Zero crossing rate (ZCR) is a measure of noise. It is used to detect number of times the signal changes its sign. ZCR is directly proportional to noise. Roll-off is a measure of high-frequency energy of the signal. It is a cumulative sum of spectrum energy. It is one of the ways to estimate the amount of high frequency produced through the audio signal. The roll-off is another measure of spectral shape.

Brightness is a measure of energy above a cut of frequency $F_c = 1500$ Hz. It gives spectral energy corresponding to frequencies higher than a given cut-off threshold. Brightness is calculated per frame of length 50 ms with half overlapping. Roughness is an estimation of sensory difference. Total roughness is average difference between all peaks of the spectrum of the signal.

The frequency components are supposed to remain sufficiently constant throughout each frame of each audio file. Irregularity is the degree of variation of successive peaks of the spectrum of the signal. Irregularity is a ratio given by square of the sum of the difference between successive peaks of the entire signal by square of the number of signal points.

Spectral centroid calculates the center of mass of magnitude of the spectrum. It is the geometric center (centroid) of the distribution. Spectral spread gives spectral spread of the sound signal around its mean value or standard deviation. It is the variance. Being the squared deviation of the random variable from its mean value, the variance is always positive and is a measure of the dispersion or spread of the distribution. Skewness returns the coefficient of skewness of data. It is the measure of symmetry of the distribution. Higher is the value of skewness, lesser is the signal symmetric. A symmetrical distribution has coefficient of skewness of zero. A positive coefficient of skewness often indicates that the distribution exhibits a concentration of mass toward the left and a long tail to the right, whereas the negative value generally indicates the opposite. Kurtosis is the sharpness of the peak of a frequency-distribution curve. It indicates whether the curve is ‘picked’ or ‘flat’ relative to the normal distribution. It is fourth standardized moment.

These audio features are then passed to the classification algorithm such as K-means. K-means clustering is unsupervised type of learning which is based on vector quantization method. It works on generating centroid which is equal to total number of classes. After training, the centroids are fixed. In testing, each test point is checked which is closer to the centroid. Based on Euclidian distance, the centroid which is closer to new test data is assigned to the same class.

K-nearest neighbor (KNN) is based on distance matrix. Distances are calculated by using Hamming, city block, Euclidian and Manhattan distances. Based on a policy of neighborhood, nearest neighbor, farthest neighbor classification is done. There is no training in KNN.

Support Vector Machines (SVM) classification uses supervised type of learning. It comprises a set of supervised learning methods which takes as an input a set of data and builds a model that assigns one of the two classes for the input. Multiclass SVM is also popularly used in musical instrument identification [15].

For identification of musical instrument, the audio signal is segmented by using onset detection method [16–18]. Acoustic vector is calculated by using MFCC feature extraction, and HMM is used to model the sequence of feature vector [5]. Other than MFCC, there exist many audio features which could be used with Gaussian model, feed-forward neural network, probabilistic neural network and tree classifiers which give the highest 83% accuracy [12]. Dimension reduction method of audio features such as principal component analysis (PCA) plays important role in the classification of Tabla Bols. PCA selects prominent audio features [19]. When 21,361 and 18,802 instances of two databases are used, multi-layer perceptron with PCA gives 98.68% accuracy [20].

Audio descriptors are used to characterize the sound. For the selection of audio descriptors, various methods are used. Hybrid selection algorithm gives accurate audio descriptors which give the highest classification accuracy among them. It is a wrapper approach where output of classification method is considered as feedback [21]. The Timber is multi-dimensional entity of sound. Timbre has been defined in various ways as per the requirement of audio analysis and applications [8, 9, 22].

Classification of Tabla Bols is done by making use of various classifiers. K-nearest neighbor (KNN) classifier is most commonly used by many researchers for the identification of musical instruments [23, 24]. Artificial neural networks (ANN) are useful machine learning classifiers that are used in many music information retrieval and classification applications [8, 12]. Hidden Markov model (HMM) with MFCC feature is used for automatic discovery of percussion patterns from audio recordings of Tabla solos. In this approach, the recordings of the Tabla solos are identified and transcribed into a sequence of syllables using an HMM model [4]. MFCC feature is used to model the Timbre of the syllables [2, 14, 25].

A non-negative matrix factorization (NMF) approach is used to analyze the strokes of the Mridangam (a South Indian Music Rhythm Instrument). Using NMF, a dictionary of spectral basis vectors is first created, and then, the composition of the strokes is analyzed. Onset is one of the prime audio features useful for automatic Tabla or Mridangam stroke detection. For onset detection, the peaks from the activation matrix are computed for the different modes. Onset detection is carried out using the group delay algorithms. Strokes are identified by using NMF and HMM. This study reported accuracy of Mridangam stroke detection as 88.40%. The modes of the Mridangam are analyzed by validating the relationship between strokes and modes of the instrument, and those modes are used for identification [26].

3 Proposed System

Automatic Tabla Bol (stroke) identification system uses audio excerpts of duration 1–3 s each, sampled at 22,050 Hz, pulse code modulation (PCM) wave representation, containing various basic Tabla Bol samples for training. The input audio database consists of 15 basic Tabla Bol samples and 20 samples per Tabla Bol. There exist seven Tabla Bol samples that are played only on right drum (Tabla), two Bol samples played only on left drum (Dagga), three Bol samples played simultaneously on both the drums and three Bol samples generated using alternate combinations of both the drums. Total 300 audio excerpts are used in this system. As a thumb rule, 70% of the total Tabla Bol samples are used for training, and 30% are used for testing.

As shown in Fig. 1, the Tabla Bol audio repository contains all the basic Tabla Bol samples along with its various samples. The combinations of the basic Tabla Bol produce the Tala (rhythm) in North Indian Classical Music. In order to not only identify a Tabla Bol, this system is also useful for identification of Tala in North Indian Classical Music. Traditionally, Mel-frequency cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCC) are most popular audio descriptors that emulate a human auditory system. The first-order derivative of these coefficients is purposely avoided in this study to analyze the features in their purest forms and to compare them with the Timbral audio descriptors. The audio features are classified into five categories namely (i) MFCC (ii) LPCC (iii) Timbral audio descriptors excluding MFCC (iv) Timbral audio descriptors with MFCC and (v) Spectral audio features. There are various ways to define a Timbre; however, in some taxonomy, MFCC is considered as part of Timbral audio descriptors [9]. This system not only identifies the Tabla Bol from North Indian Classical Music but also evaluates the contribution of MFCC in the definition of Timbral audio descriptors.

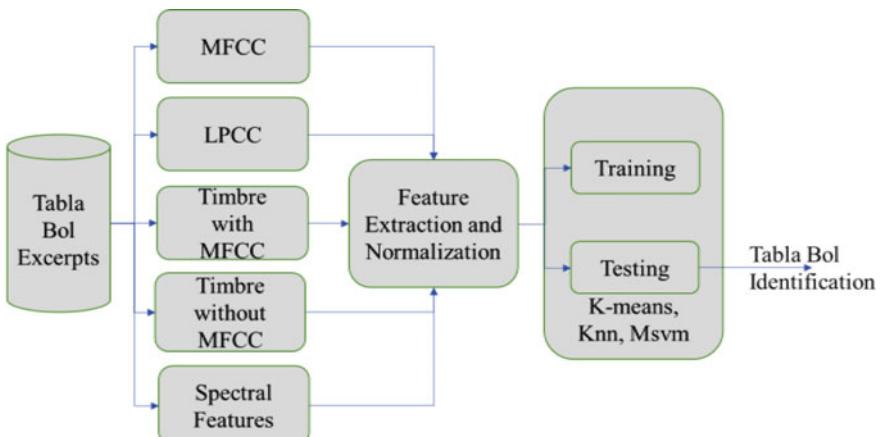


Fig. 1 Automatic Tabla Bol identification system

The system extracts the audio features from all the mentioned categories and passes these features to the classifiers. In order to deal with the range of values of these features and the negative coefficients of MFCC, a normalized version of these audio features is used for training and testing. The typical normalization of the audio features is used to normalize the audio descriptors between 0 and 1. A wrapper approach of hybrid selection algorithm could be applied here to choose from various sets of audio descriptors; however, due to less number of features being considered, we have used filter Approach in contrast to the wrapper approach of hybrid selection algorithm [21].

The normalized audio features are then passed to classifiers namely K-means (a vector quantized algorithm) and K-nearest neighbor (statistical classifier), multiclass Support Vector Machine (MSVM) for cross validation. After training, the test samples of Tabla Bol are tested to verify the classification accuracy. The performance of the proposed system is calibrated using parameters of percentage accuracy and using a confusion matrix. The percentage accuracy of Tabla Bol classification is obtained by the product of ratio of number of test samples correctly classified to the number of total test samples given for classification and 100.

We have used here multiclass SVM since the number of Tabla Bol is equal to number of classes in which Bolts are to be classified. The parameters precision, recall and sensitivity are used to correlate the audio samples that are misclassified into different classes.

4 Experiments and Results

4.1 Using MFCC

Experiments have been carried out using dataset mentioned above and MFCC as feature extractor. The result shows that MFCC gives the highest Tabla Bol recognition accuracy when used with SVM, as shown in Fig. 2. However, since MFCC accumulates high energy components in its first thirteen components, it does not address the Timbral sound produced by Tabla. As mentioned earlier, the Tabla is also a harmonic instrument that works on resonance. Therefore, the system gives high results for a smaller number of classes (Tabla Bol), and there is no significant change in the percentage accuracy obtained for Tabla Bol identification for classifier KNN for its variants 1NN, 2NN and 3NN.

4.2 Using LPCC

As shown in Fig. 3, when LPCC is used as feature extraction method, SVM shows promising results for percentage accuracy of Tabla Bol identification system. The

Fig. 2 Accuracies obtained for all classifiers using MFCC feature

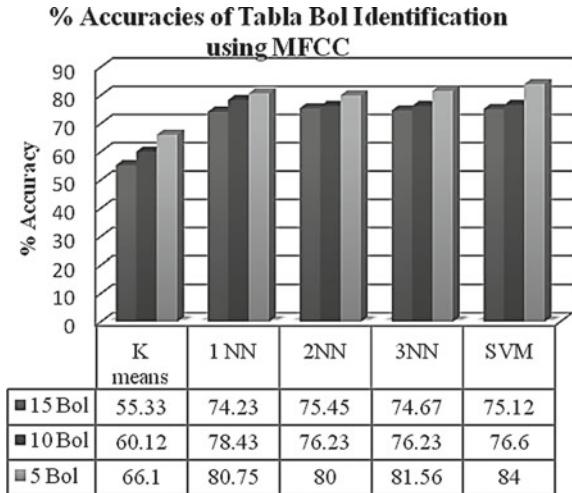
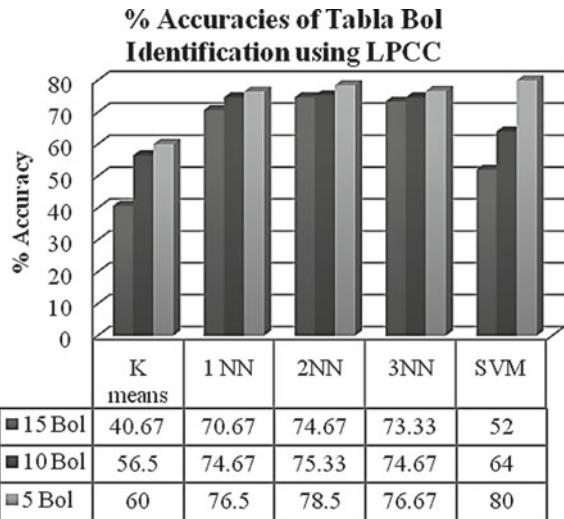


Fig. 3 Accuracies obtained for all classifiers using LPCC feature

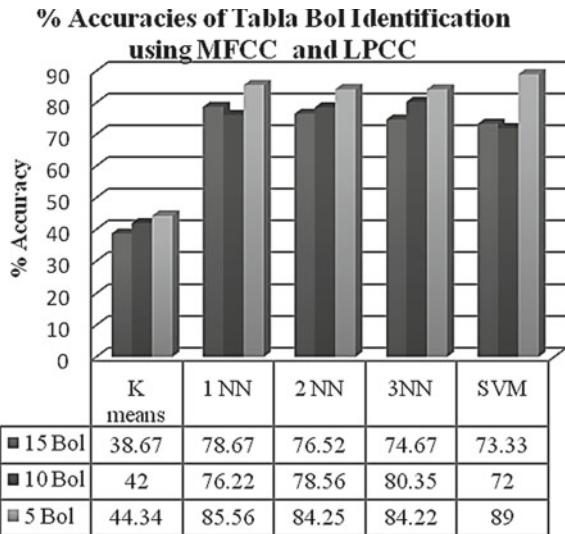


system exhibits four classification results. Even for SVM classifier when number of Tabla Bol is increased from 5 to 15. There are twenty coefficients of LPCC used.

4.3 Using MFCC and LPCC

Traditionally, MFCC is useful for human voice recognition, and LPCC is useful for musical instrument identification. MFCC represents human auditory system like

Fig. 4 Accuracies obtained for all classifiers using MFCC and LPCC feature



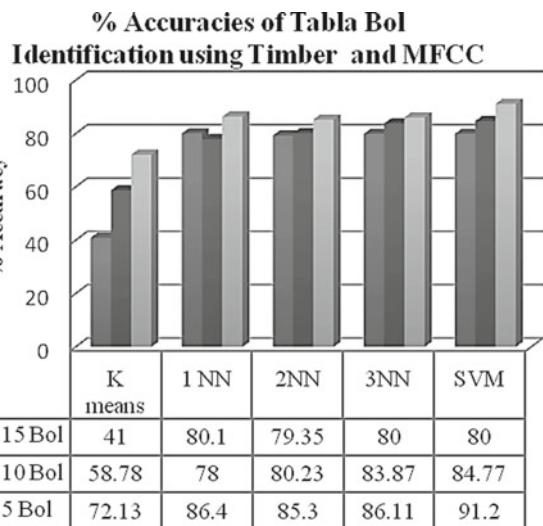
LPCC. The strength of MFCC and LPCC lies in their first thirteen coefficients. When the features of MFCC and LPCC are combined, it shows drastic impact on identification of Tabla Bol. The highest accuracy obtained is 89% for SVM classifier as shown in Fig. 4. The probable reason behind this increasing accuracy is that Tabla is a percussion instrument that exhibits dissimilar properties of tonal sound from string instruments or woodwind instruments. The typical sound produced by Tabla instrument is homophonic in texture like the vocal voice of a human which is also produced through skin. For 1NN classifiers, the results are better than 3NN classifier.

4.4 Using Timber and MFCC

Typically, MFCC is considered as Timbral audio descriptor. However, as per the Taxonomy explained by Olivier [9], there are other audio descriptors which precisely defined Timber of a musical instrument.

When combined together MFCC and other Timbral audio descriptors, the accuracy of Tabla Bol identification is found to be the highest. For SVM classifier, when MFCC and other Timbral audio descriptors are combined as audio feature, the Tabla Bol identification accuracy is found to be 91.2% as shown in Fig. 5. The Timbral audio descriptors include zero crossing rate (ZCR), roll-off, brightness, roughness, irregularity and MFCC.

Fig. 5 Accuracies obtained for all classifiers using Timber and MFCC feature



4.5 Using Timber Without MFCC

If is neglected from Timbral audio descriptors list, then Tabla Bol identification accuracy decreases drastically. As shown in Fig. 6, the Tabla Bol identification accuracy is found to be the lowest among all. This proves that the accuracy of Tabla Bol identification system could be the highest only if MFCC and other Timbral audio descriptors are combined as audio features. The individual utilization of MFCC or only Timber does not exhibit good performance as compared to their combination.

Fig. 6 Accuracies obtained for all classifiers using Timber without MFCC feature

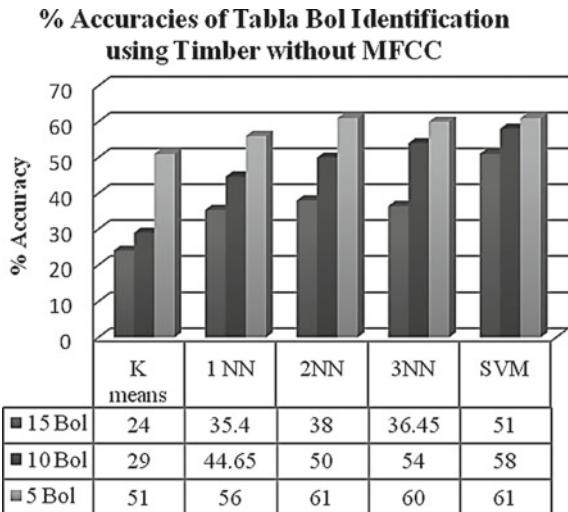
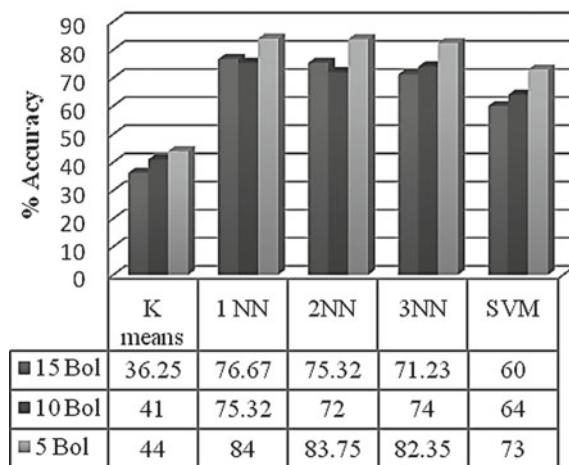


Fig. 7 Accuracies obtained for all classifiers using spectral feature

% Accuracies of Tabla Bol Identification using Spectral Features



4.6 Using Spectral Features

Under the category of spectral audio descriptors, the following features are used, namely spectral centroid, spectral spread, spectral skewness, spectral kurtosis, spectral flatness and entropy. As shown in Fig. 7, the Tabla Bol identification accuracy for only spectral features is found to be 84% for 1NN classifier. However, for higher number of classes of Tabla Bol, the accuracy decreases.

4.7 Comparison of All Classifiers

In order to cross validate the results, the results obtained from variants of nearest neighbor are compared with the results obtained from K-means and SVM. As shown in Table 1, the highest accuracy obtained is 91.2% of SVM. Followed by SVM, KNN

Table 1 Accuracies of all classifiers

Classifiers	% accuracy
K-means	72.13
1NN	86.4
2NN	85.3
3NN	86.11
SVM	91.2

Table 2 Accuracies of classifier for all features

Features	% accuracy
MFCC	84
LPCC	80
MFCC + LPCC	89
Timber + MFCC	91.2
Only timber	61
Spectral	73

classifier provides better Tabla Bol identification accuracy over K-means classifier. There is no significant change in the percentage accuracy of variants of nearest neighbor.

4.8 Comparison of All Audio Descriptors

When Tabla Bol identification accuracy obtained from all the audio descriptors sets mentioned above is compared, Timber with MFCC audio features gives the highest Tabla Bol identification accuracy as shown in Table 2. Followed by Timber with MFCC audio feature, MFCC and LPCC combination gives better Tabla Bol identification accuracy.

5 Conclusion

An automatic Tabla Bol identification system has been implemented by using various categories of different set of audio features and various classifiers. Traditionally, for content-based audio feature analysis, MFCC is best suitable audio descriptor along with SVM as best suitable classifier. When applied as input audio database of Tabla Bol containing 300 sound samples (15 Tabla Bol and 20 samples per Tabla Bol), with MFCC and Timbral audio descriptor together yields highest Tabla Bol identification accuracy as 91.2% using SVM classifier.

References

1. Misra S, Das TK, Saha P, Baruah U, Laskar R (2015) Comparison of MFCC and LPCC for a fixed phrase speaker verification system, time complexity and failure analysis. In: 2015 international conference on circuit, power and computing technologies (ICCPCT), pp 1–4

2. Srinivasamurthy A (2016) A data-driven bayesian approach to automatic rhythm analysis of Indian Art music," Music technology group, Dept. of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, thesis report TESI DOCTORAL UPF/2016
3. Beronja S (2008) The art of Indian Tabla. Rupa and Company, New Delhi
4. Gupta S (2015) Discovery of percussion patterns from Tabla solo recordings. Universitat Pompeu Fabra, master thesis report 20 Sept 2015
5. Gillet O, Richard G (2003) Automatic labelling of Tabla signals. In: 4th ISMIR conference
6. Clayton LMR (1992) The rhythmic organization of North Indian classical music: Tal, Lay and Laykari. University of London, London. Ph.D. thesis 10731377
7. Peeters G (2004) A large set of audio features for sound description (similarity and classification) in the CUIDADO project, 23 Apr 2004
8. Park TH (2004) Towards automatic musical instrument timbre recognition. Princeton University, Candidacy, PhD. thesis
9. Lartillot O, MIR ToolBox manual
10. Han W, Chan CF, Choy C-S, Pun K-P (2006) An efficient MFCC extraction method in speech. In: ISCAS 2006. IEEE
11. Zhao X, Wang DL (2013) Analysing noise robustness of MFCC AND GFCC features in speaker identification. In: ICASSP 2013. IEEE
12. Chordia P (2005) Segmentation and recognition of Tabla strokes. In: 6th international conference on music information retrieval, London. UK, 11–15 Sept 2005, pp 107–114
13. Gillet O, Richard G (2004) Automatic transcription of drum loops. In: 2004 IEEE international conference on acoustics, speech, and signal processing, vol 5, 17–21 May 2004 (Online). <https://ieeexplore.ieee.org/document/1326815/>
14. Kumar JC, Sarala P, Murthy HA, Sivaraman UK, Kuriakose J (2015) Akshara transcription of Mrudangam strokes in Carnatic Music. In: 2015 twenty first national conference on communications (NCC), Mumbai, India, p 6
15. Ozbek ME, Delpha C, Duhamel P (2007) Musical note and instrument classification with likelihood-frequency-time analysis and support vector machines. In: 15th European signal processing conference (EUSIPCO 2007), Poznan, Poland, 3–7 Sept 2007
16. Manoj Kumar PA, Sebastian J, Murthy H (2015) Musical onset detection on Carnatic percussion. IEEE
17. Manoj Kumar PA, Kuriakose J, Sebastian J, San S (2014) Onset detection and stroke recognition for percussion instruments
18. Daudet L, Abdallah S, Duxbury C, Davies M, Sandler MB, Bello JP (2005) A tutorial on onset detection in music signals. IEEE Trans Speech Audio Process, p 13
19. Banerjee K, Midya V, Chakraborty S, Sanyal S, Banerjee A, Sengupta R, Ghosh D, Patranabis A, Harmonic and timber analysis of Tabla strokes
20. Deolekar S, Abraham S (2016) Classification of Tabla strokes using neural network, vol 1. Springer
21. Deshmukh SH (2012) A hybrid selection method of audio descriptors for singer identification in North Indian classical music. In: IEEE Explorer, Himeji, Japan, pp 224–227
22. Zhang X, Zbigniew WR Analysis of sound features for music timbre recognition
23. Banchhor SK, Khan A (2012) Musical instrument recognition using spectrogram and autocorrelation. Int J Soft Comput Eng (IJSCCE) 2, ISSN: 2231-2307, 1
24. Shinde P, Javeri V, Kulkarni O (2014) Musical instrument classification using fractional fourier transform and KNN classifier. Int J Sci Eng Technol Res (IJSETR) 3, 5
25. Ramires A (2017) Automatic transcription of drums and vocalized percussion. U. Porto, master's thesis
26. Anantapadmanabhan A, Bellur A, Murthy HA (2013) Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 181–185

Chapter 30

Sentiment Analysis on Aadhaar for Twitter Data—A Hybrid Classification Approach



Priya Kumari and Md. Tanvir Uddin Haider

1 Introduction

In today's scenario, individuals express their opinion on government policies, and other contemporary issues trending on social networking sites such as Twitter, Facebook. An opinion can be positive or negative sentiment, emotion or attitude for an entity by an opinion holder [1]. A lot of research has been done on Twitter data in order to classify the tweets and analyse the classification result [2–5]. Data analysis is a set of sequential steps comprise of data collection from sources, transforming the data and organizing it so that it can be modelled with the aim to extract useful information, which is further useful in decision-making. Nowadays, the Government of India is linking the Aadhaar card with many government schemes, but there are many issues with respect to security and privacy of the Aadhaar database that needs to be addressed [6, 7]. The scope of this paper is to find the sentiments of people on Aadhaar limited to the aspect “*Enrolment Process*”, “*mAadhaar*”, “*CustomerCare*” and “*Security*” using hybrid method. Hybrid classification technique has been used in many areas to improve the accuracy of the classification problem, but in case of tweets sentiment analysis, hybrid approach has not been explored much [8–10]. In sentiment analysis, raw data is collected from a variety of sources such as from databases, crawling online sources, documents, sensors, cameras, videos. On the acquired data, pre-processing is performed to organize and prepare it for data analysis. After data pre-processing, there are several methods and algorithms available which can be used for data analysis and visualization. Figure 1 depicts the different stages of sentiment analysis.

P. Kumari (✉) · Md. T. U. Haider
National Institute of Technology Patna, Patna, India
e-mail: priya.wit123@gmail.com

Md. T. U. Haider
e-mail: tanvir99@yahoo.com



Fig. 1 Process involved in sentiment analysis

In our research work, data has been collected using Tweepy API consist of 7456 tweets. Further, the aspect of each tweet has been identified using manually created bag of words (BOW) for “*Enrolment Process*”, “*mAadhaar*”, “*CustomerCare*” and “*Security*” and barring above-mentioned aspects other tweets have been discarded for analysis. This filter data set is further labelled for polarity to define positive, negative and neutral values for training data set using SentiWordNet. Training data comprises of 515 positive, 1271 negative and 990 neutral tweets. Hybrid model has been proposed that combines machine learning classifiers with long short-term memory (LSTM) network to maximize classification accuracy. In the proposed hybrid model, wrapper method (recursive feature elimination cross-validation) is used to select the best-performing feature subset that depends on the performance of machine learning classification algorithm. Further, these features have been used as an input to deep learning LSTM network [11]. In addition to hybrid model, we have also used base classifiers such as naïve Bayes (NB), logistic regression (LR) and support vector machine (SVM) to do comparative analysis of result.

This paper has been structured in five sections. Section 2 briefs about the related work done in this domain. Section 3 discusses the framework used for sentiment analysis in detail. Model evaluation and result analysis have discussed in Sect. 4, and conclusion has been deducted and future work has been proposed in Sect. 5.

2 Related Work

The benefits and challenges of Digital India project have been highlighted through descriptive analysis on data sets from news, websites and magazines [12]. Authors have analysed the people’s opinion on different government schemes using unsupervised lexicon-based approach [3, 13] on Twitter data and classified it into three groups but no focus has been given on different aspects and its polarity. A dictionary-based approach is used to classify only 500 tweets related to Digital India Campaign and classified it into three groups but to improve accuracy through handling negation, emoticons, sarcasm in data have not been paid attention [2]. The proposed deep learning algorithm CNN for opinion mining was used with word embedding Glov2Vec method on Twitter data sets, and it performs better than baseline models SVM and LR using BoW [4]. Author has evaluated the accuracy of sentiment analysis using an ensemble classifier [8] that combines the base learning classifier (NB, SVM, LR, RF) to form a single classifier and result showed that ensemble classifier performs better than stand-alone classifier. Hybrid deep learning system using word embedding that combines CNN and RNN has used to get optimal accuracy on movie

review data sets [9]. The proposed ensemble voting classifier using base classifiers (DT, NB and SVM) has achieved higher accuracy in comparison with stand-alone base classifiers [10]. Authors have used the concept from DBpedia, SentiWordNet for additional feature selection and experiment this with domain-specific features and observed that the accuracy results of NB and SVM have slightly increased [14].

From the above-related works, it can be said that there is a lack of deep analysis of sentiment analysis on Aadhaar scheme Twitter data set. This paper focuses on evaluating the sentiment of Twitter data on Aadhaar application using hybrid approach combining machine learning base classifiers with LSTM network with the aim to improve the accuracy.

3 Proposed Approach

The main objective of this paper is to develop an approach for classification model to find the opinion of Twitter data related to Aadhaar project. The proposed work flow for this approach is depicted in Fig. 2. Our proposed framework is divided into two primary components. First component consists of data extraction, pre-processing and feature creation using term frequency (TF) and term frequency-inverse document frequency (Tf-Idf). Second component consists of hybrid model which is further divided into two submodules recursive feature selection cross-validation and deep neural network. The following sections brief about each module of proposed model.

3.1 Data Acquisition

Tweets specific to Aadhaar have been extracted from Twitter using Tweepy API into SQLite database table. Tweets consist of hashtags #Aadhaar, #mAadhaar, #AadhaarOnlineServices, #aadhaar, #Aadhar, and Government official Twitter account @UIDAI and @ceo_uidai have been taken into consideration for data extraction. Acquired tweets are categorized into aspects based on manually created BoW. These tweets' categorization is shown in Table 1.

3.2 Data Pre-processing

Data sets from Twitter have been stored, parsed and pre-processed, and then useful features have been extracted from the data. Figure 3 shows the snapshot of pre-processed data. Pre-processing has been done before feature engineering, and below-mentioned methods have been applied on data sets.

- Expansion of negation word, e.g. *won't* replace with *will not*.

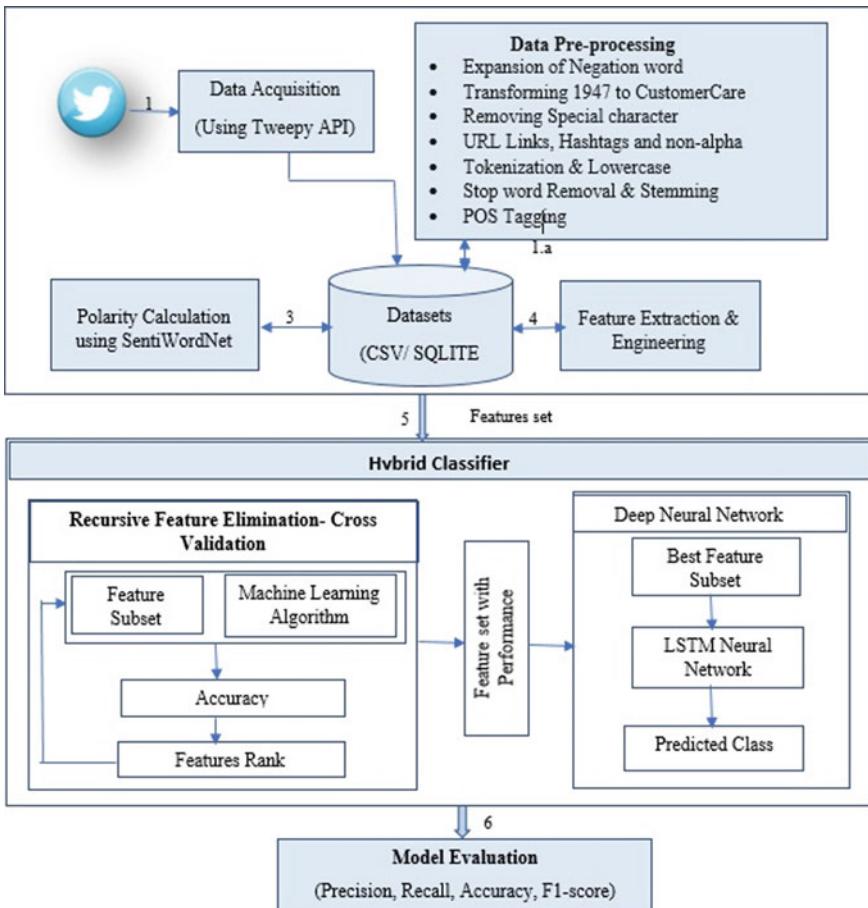


Fig. 2 Proposed hybrid system framework

Table 1 Aspect-based tweets count

Aspect	Total	Positive	Negative	Neutral
Enrolment process	759	176	349	234
Security	705	94	266	345
mAadhaar	481	88	268	125
CustomerCare	831	157	388	286
Total	2776	15	1271	990

id	Tweet	Preprocessed Tweet
0	@suchetadalal @UIDAI Send them a mail. They respond quickly	send mail respond quickly
1	@UIDAI Had a worst experience since morning by contacting UIDAI help desk, I am neither able to verify aadhaar number nor able to order reprint, while contacting help desk executive, the call disconnected by them in between and not able to resolve my problems	bad experience since morning contact uidai help desk neither able verify aadhaar number able order reprint contact help desk executive call disconnect not able resolve problem

Fig. 3 Snapshot of pre-processed data

- Removal of user mentions, URL and special characters, e.g. @UIDAI.
- Transforming digit “1947” (toll-free number related to Aadhaar complaints) to “CustomerCare” as Customer Care is one of the aspects of Aadhaar in our work.
- Tokenization, stopwords removal, lowercase and stemming.
- Part-of-speech (POS) tagging for polarity calculation using SentiWordNet.

3.3 Feature Extraction and Engineering

Feature extraction plays a vital role in text classification. It directly affects the performance of text classification model. In textual data sets, each feature is a word and to make each word understandable to machine learning, we need to convert this into feature vector. In this paper, term frequency and Tf-Idf approach have been used with unigram to analyse the impact of this on machine learning classification algorithms and the same is used to create features set for the proposed model. The purpose of feature selection is to reduce the space of feature which will minimize the computation code. Any classification model performance is highly affected by the data set quality. Irrelevant features of data set reduce the model performance as well as increase the cost of classification process in some cases [15].

3.4 Hybrid Classifier

This model combines machine learning algorithms with deep neural network in which the output of machine learning algorithms (NB, SVM and LR) is fed into LSTM neural network as input features. The second component of the proposed framework is hybrid classifier which is further divided into two submodules. In first submodule, the wrapper method (recursive feature elimination cross-validation) is used to find the best-performing feature subsets based on the machine learning classification algorithm accuracy using score function and cross-validation. In the second submodule, LSTM network has been built with the best feature subsets selected from

Table 2 LSTM network summary

Parameters	Multiple class
Features	Dynamic: best-performing feature subset
Input vector	Feature subset
LSTM layer	1
Hidden layer	1
Dropout layer and value	1, 0.2
Output layer	1
Activation function	Softmax
Optimizer	Rmsprop
Training and testing sample	70, 30%
Epochs and batches	8, 12
Performance metrics	Precision, recall, F1 score, accuracy

the first submodule and predicts the class of tweets. Detail of proposed approach is discussed below.

Recursive Feature Elimination Cross-Validation (RFECV) RFE feature selection is recursive process which finds the importance of features and ranks accordingly. It is a type of wrapper method that works on greedy optimization to find the best-performing feature subsets. This process recursively builds a model and selects the best or worst feature set based on the performance of classification model. RFE requires a specified number of features in advance, but in textual data set, it is not possible to know the best n number of features in advance. RFECV approach finds the best-performing feature subset using cross-validation. The feature selection is done iteratively by training a model, rank the features and remove the lowest ranking feature based on the performance of model using cross-validation. In this submodule, cross-validation with RFE is used to find the best-performing feature subset with three different machine learning classifiers (NB, LR and SVM) and uses this feature subset as an input to LSTM.

Deep Neural Network-LSTM It is an extension of recurrent neural network (RNN) with extended memory. It enables RNN to remember input sequence over a long period of time. LSTM architecture consists of three gates: input, forget and output gates. Due to the unique characteristic of remembering the previous input data in sequential order, it plays a key role to predict the text class accurately. The output of the first submodule is the best-performing feature subset which is fed as input to LSTM network, and finally, this network predicts the class of each tweet. The summary of LSTM network creation is depicted in Table 2.

4 Model Evaluation and Result Analysis

The performance of this system is evaluated using four metrics that are precision, recall, F1 score and accuracy. Out of 2776 tweets, 1910 has been used for training the classifier and the rest of the tweets are used for testing. The naïve Bayes, logistic regression, support vector machine classifiers are implemented individually using feature engineering algorithms which uses Tf-Idf and term frequency with unigram. Further, their performance metrics (precision, recall, F1 score and model accuracy) are calculated. Figure 4 shows the performance metrics of these classifiers. In this figure, blue line shows TF while orange line shows Tf-Idf. The result shows that accuracy of logistic regression using TF is better than other classifier, whereas the F1 score of SVM using term frequency is better than rest of the models.

The comparative result of single classifier and hybrid classifier is depicted in Table 3. The result of classifiers clearly shows that the proposed model performs better than single classifier and the algorithm RFECV-LR achieved high accuracy in comparison with another classifier. Accuracy achieved by this model is 83.2% where precision = 0.78, recall = 0.81 and F1 score = 0.80.

The accuracy results of the proposed hybrid classifier approach using RFECV-LR, RFECV-SVM and RFECV-NB with LSTM on training and test data have been depicted in Fig. 5. The figure clearly indicates that all these hybrid classifier's performances are better than single machine learning classifiers. The pattern of accuracy score of (RFECV-LR) and (RFECV-SVM) is almost the same for both training and testing data set.

A comparative result analysis of our proposed model with existing similar works is depicted in Table 4. Here, Kouloumpis et al. [16] applied a combination of feature engineering using Lexicon, n -gram and microblogging features with machine learning algorithm AdaBoost to increase the accuracy of sentiment classification on tweets. The proposed approach achieved F1 score of 0.68 and 0.65 for HASH and EMOT data sets, respectively.

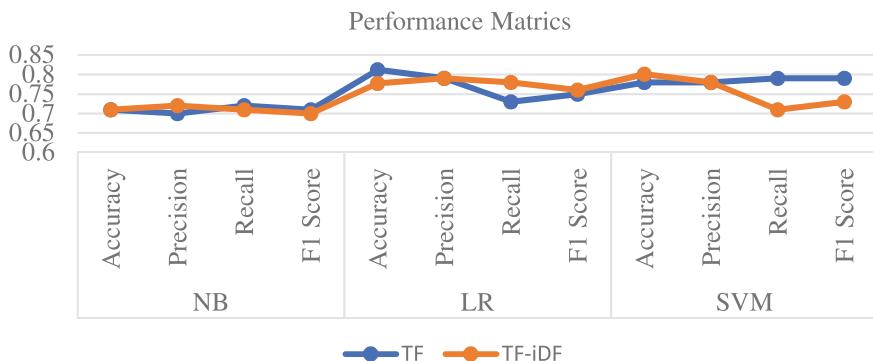
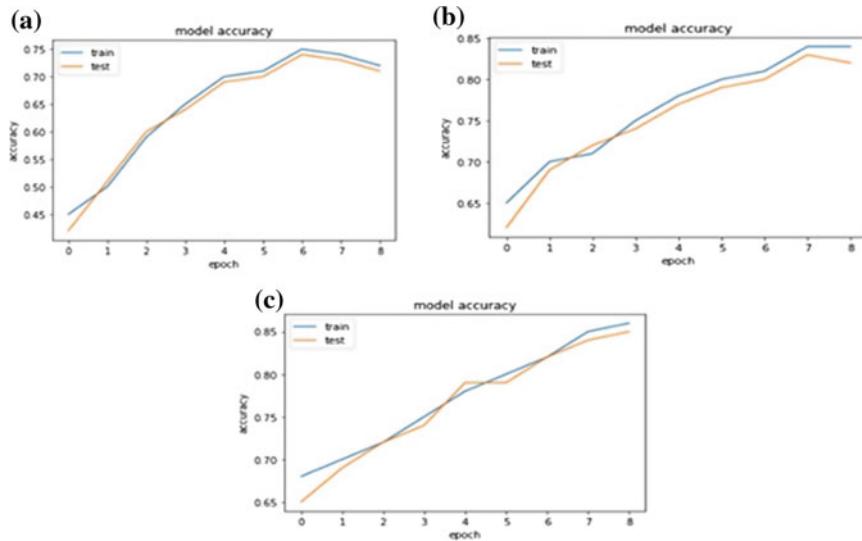


Fig. 4 Performance metrics of machine learning classifiers using TF and TF-IDF

Table 3 Summary of the performance metrics for single and hybrid classifiers

Classification approach	Feature selection	Precision	Recall	F1 score	Accuracy
<i>Single classifier</i>					
NB	TF	0.70	0.72	0.71	0.714
	TF-IDF	0.72	0.71	0.70	0.714
SVM	TF	0.78	0.79	0.79	0.786
	TF-IDF	0.78	0.71	0.71	0.8063
LR	TF	0.79	0.73	0.75	0.813
	TF-IDF	0.79	0.78	0.76	0.777
<i>Hybrid classifier</i>					
RFECV-NB + LSTM	Best-performing feature subset	0.71	0.73	0.72	0.733
RFECV-NB + LSTM		0.78	0.81	0.80	0.832
RFECV-NB + LSTM		0.79	0.84	0.82	0.853

**Fig. 5** Model accuracy for train and test data **a** RFECV-NB with LSTM, **b** RFECV-SVM with LSTM, **c** RFECV-LR with LSTM

Hamdan et al. [14] used an additional feature which is extracted from WordNet and DBpedia. Here the experimental result of F1 score is 0.53 for SVM and 0.49 for NB. Da Silva et al. [17] proposed ensemble classifier which combines the machine learning classifiers RF, SVM, NB and LR using feature representation BOW and feature hashing. In this, the experimental results of accuracy are 76.99, 76.81, 84.89 and 81.06 for HCR, OMD, Sanders and STS data sets, respectively. The above

Table 4 Comparative results with existing work

References	Data sets	Features	Algorithms	Outcomes
Kouloumpis et al. [16]	The emoticon (EMOT) and hashtag (HASH)	Unigram, lexicon	AdaBoost classifier	Achieved F1 score 0.68 for HASH and 0.65 for HASH + EMOT data sets
Hamdan et al. [14]	SemEval- 2013 data sets	Unigrams, DBpedia, WordNet, SentiWordNet	NB, SVM	The average F1 score for pos + neg data sets is 0.53 and 0.49 for SVM and NB classifier respectively
Da Silva et al. [17]	OMD, Sanders, Stanford (STS) and HCR data sets	Lexicon, BoW and feature hashing	SVM, NB, RF and LR	Ensemble classifier using base classifiers achieved accuracy scores of 76.99, 76.81, 84.89 and 81.06 for HCR, OMD, Sanders and STS data sets, respectively
Proposed work	Twitter-Aadhaar data sets	RFECV	RFECV-NB + LSTM	Accuracy – 0.733 and F1 score 0.72
			RFECV-SVM + LSTM	Accuracy – 8.32 and F1 score 0.80
			RFECV-LR + LSTM	Accuracy – 0.853 and F1 score 0.82

comparative chart clearly shows that our proposed hybrid model has done better in terms of accuracy and F1 score as compared with the existing model.

5 Conclusion and Future Work

In this paper, Twitter API using Spyder IDE has been used for data extraction. Tweets from Twitter have been collected in SQLite DB table, and pre-processing task is done using Natural Language Toolkit (NLTK) for sentiment analysis. Hybrid classifier using machine learning classifiers with RFECV feature selection method and LSTM has been used for sentiment analysis. The experimental result of classifiers confirms

that our proposed model performs better than base classifier for Aadhaar domain data sets. Considerable amount of negative and neutral opinions has been given by individuals in compare with positive tweets on Aadhaar.

Future scope includes ontology-based sentiment analysis is to find out automatic aspect's selection as our work is limited to only four aspects of Aadhaar.

References

1. Liu B (2010) Sentiment analysis and subjectivity. An NLP handbook
2. Mishra P, Rajnish R, Kumar P (2016) Sentiment analysis of Twitter data: case study on digital India. In: InCITE—The Next Generation IT Summit, IEEE
3. Vinodkuma ChR, Bhaskari L (2017) Sentiment analysis of #MakeInIndia & #demonetization using R. IOSR J Comput Eng 19(6), Ver. II:42–47. e-ISSN: 2278–0661, p-ISSN: 2278-8727
4. Jianqian Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for Twitter sentiment analysis. Published in IEEE access, vol 6
5. Amoli A, Jivane N, Bhandari M, Venkatesan M (2016) Twitter sentiment analysis of movie reviews using machine learning techniques. Int J Eng Technol 7(6)
6. Bloomberg <https://www.bloombergquint.com/aadhaar/164-aadhaar-related-frauds-reported-since-2011-most-in-2018-new-database>. Last Accessed 18/08/08
7. The Tribune, <https://www.tribuneindia.com/news/haryana/aadhaar-seeding-scam-in-food-dept/585766.html>. Last Accessed 2018/05/30)
8. Ankit, Saleena N (2018) An ensemble classification system for Twitter sentiment analysis, published by Elsevier Ltd. ICCMS conference
9. Chakravarthy A, Deshmukh S, Desai P, Gawande S, Saha I (2018) Hybrid architecture for sentiment analysis using deep learning. Int J Adv Res Comput Sci 9(1)
10. Janane SK, Keerthana MS, Subbulakshmi B (2018) Hybrid classification for sentiment analysis of movie reviews. Int J Eng Sci Res Technol. ISSN: 2277-9655
11. Kumar K, Haider MTU (2019) Blended computation of machine learning with the recurrent neural network for intra-day stock market movement prediction using a multilevel classifier. Int J Comput Appl
12. Mohanta G, Debasis SS, Nanda SK (2017) A study on growth and prospect of digital India campaign. Saudi J Bus Manag Stud 2(7):727–731
13. Naiknaware BR, Kawathekar S, Deshmukh SN (2017) Sentiment analysis of Indian government schemes using Twitter datasets. IOSR J Comput Eng (IOSR-JCE). e-ISSN: 2278-0661, p-ISSN: 2278-8727, pp 70–78
14. Hamdan H, Bechet F, Bellot P (2013) Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In: Seventh international workshop on semantic evaluation (SemEval 2013), vol 2, pp 455–59
15. Khan A, Baharudin B, Khan K, Lee LH (2010) A review of machine learning algorithms for text-documents classification. J Adv Inf Technol 1(1)
16. Koulopis E, Wilson T, Moore J (2011) Twitter sentiment analysis: the good the bad and the OMG! In: Proceeding of AAAI conference on weblogs and social media, pp 538–541
17. Da Silva NFF, Hruschka ER, Hruschka Jr ER (2014) Tweet sentiment analysis with classifier ensembles. Decis Support Syst, 170–179

Chapter 31

Song Recommendation System Using Hybrid Approach



Niket Doke and Deepali Joshi

1 Introduction

Everyday our brain is bombarded with lots of questions, what to wear? What to buy? What song to listen to? Hence, we highly rely on recommendation system to make choices. With large amount of data available on the Internet, one has millions of options to choose from. It is a big challenge to provide recommendations to people from the large data available on the Internet. E-commerce giants like Amazon and eBay provide personalized recommendations to users based on their taste and history, while companies like Spotify [1] and Pandora use ML and deep learning techniques to provide appropriate recommendations. In this paper, we have focused on implementing a good personalized recommendation system using user's history. We first implemented popularity-based model which is very simple but is not personalized followed by content and collaborative-based filtering which provide personalized recommendations based on history. We have also implemented a hybrid approach in which we combine both content and collaborative techniques to extract maximum accuracy and to overcome drawbacks of both types.

2 Dataset, Preprocessing and Notations

We used the million song dataset provided by Kaggle [2] which was provided for million song challenge for predicting half history of 11,000 users by training the other

N. Doke (✉) · D. Joshi
Department of Information Technology, Vishwakarma Institute of Technology, Pune, India
e-mail: niket.doke16@vit.edu

D. Joshi
e-mail: deepali.joshi@vit.edu

Table 1 User to rating mapping table

x	Rating	x	Rating	x	Rating
$0 < x \leq 0.2hx$	1	$0.4hx < x \leq 0.6hx$	3	$x = hx$	5
$0.2hx < x \leq 0.4hx$	2	$0.6 < x \leq 0.8hx$	4		

Table 2 Rating matrix

U set of all User	\rightarrow		\rightarrow		\leftarrow	I set of all songs
r_u rating vector of user 1 (row)	\rightarrow	User 1	Song 1	Song 2	Song 3	Song n
		User 1	2	4 ($r_{u,i}$ i.e. $r_{1,1}$)	0	0
		User 2	1	5	3	2
		User m	0	0	0	4

\uparrow \uparrow
 r_i rating vector for song 1 (column) r_i mean rating given to song n $\sum r_{i,n} / \# \text{ users}$

half and full listening history of other million users. The dataset consists of two files, viz. triplet file and metadata file. The triplet file is a collection of user id, song id and listen count, while the metadata file consists of data regarding the songs like the song id, artist, album and year which acts as features of song/feature vector. We have used only 1 million records from triplet file for our recommendation system which has over 40,000 users with 9000 songs as actual dataset is huge, hence computationally expensive. Later, we did little preprocessing on the data. After combining the triplet and metadata file, we converted listen counts to ratings between range 1 and 5 as it is easier to work with ratings. This was done by considering the highest listen count of each user as highest rating 5, for that user and ratings for other songs were calculated accordingly. Assuming x to be listen count and hx be maximum rating provided by that user then the mappings are given in Tables 1 and 2.

Let us look at all the notations that are used in this paper. Rating matrix is formed by using these triplets where rows will represent the users, while columns will represent the items. So, any row consists of ratings given by the user to all the items in his/her history and ratings of other items are unknown and for simplicity we have to be filled them with 0. So, we have a universe consisting of a set U of users and set I of songs. The rating matrix is denoted by R , with $r_{u,i}$ being the rating user u provided for item i . \bar{r}_u and \bar{r}_i are the average of a user u or an item i 's ratings, respectively. Other notations are shown in the diagram below.

3 Algorithms

3.1 Popularity-Based Model

This is the most simple and intuitive model. In this model, we suggest the user the N topmost songs—the most popular songs. This is like the trending part available on YouTube. Popularity of songs is calculated based on listen count or ratings, then the songs are arranged in descending order based on popularity and top N songs are recommended to the user. This model has several problems as it is very naive one. First, this does not give personalization, that is, everyone is recommended with same most popular songs. Also, some unpopular songs are never suggested in this model.

3.2 Content-Based Filtering

Content-based filtering focuses on the user's account in order to give suggestions to the user. User's account has all the information about user's taste, and content-based filtering uses this aspect for recommendation. User's history plays an important role in this model. We try to find songs similar to the ones which the user has rated positively in his history. Each song can be represented with a feature vector. Similarity between any two songs can be found by using cosine similarity. In cosine similarity, we try to find angle between two features vectors representing the two songs which are found by dot product of two vectors divided by the norm of the two vectors. Smaller the angle is, the closer the feature vectors are and hence more similar the two songs are [3].

The similarity between any two songs denoted by feature vectors \hat{w}_c and \hat{w}_s is given by:

$$u(c, s) = \cos(\overline{w}_c, \overline{w}_s) = \frac{\overline{w}_c \cdot \overline{w}_s}{\|\overline{w}_c\| \|\overline{w}_s\|} \quad (1)$$

3.3 Collaborative Filtering

Collaborative filtering is the most researched and frequently used algorithm in recommendation systems which was mentioned and described by Paul Resnick and Hal Varian in 1997 [4]. This algorithm is entirely based on past behavior and not on the context. This makes it one of the most commonly used algorithms as it is not dependent on any additional information, that is, it does not need any metadata/features about the songs for recommendation, and hence, it allows us to predict songs without actually knowing, what the song is about? Who is the singer? and so on. It only uses

only the ratings given by the user for finding recommendation for user. There are two types in this model: item-based collaborative filtering and user-based collaborative filtering.

User-based Collaborative filtering

This approach was proposed in the end of 1990s by the professor of University of Minnesota Jonathan L. Herlocker [5]. The main intuition behind this model is similar users listen to similar songs. In this algorithm, we assume users having a similar history or rating pattern have same taste, and hence, we can recommend songs from history of users having similar taste to our current user whom we have to recommend songs.

Similarity between the users can be calculated by using various techniques like cosine similarity or Pearson's correlation. Pearson's correlation is like cosine similarity with a little variation—mean normalized weights. The need of this normalization is due to difference in perspective of different users. So, similarity between the users using Pearson's correlation is given by [5]

$$S(u, v) = \frac{\sum_{i \in I_u \cap I_v} ((r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v))}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \quad (2)$$

Ratings can be calculated by using the following formula

$$P_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} S(u, u') (r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} S(u, u')} \quad (3)$$

Here, N is set of K most similar users. Instead of using all users, we'll use K most similar user akin KNN. Same formula can be used for item-based collaborative filtering.

Item-based Collaborative filtering

The intuition behind this method is items which are rated similarly by the users are similar. This approach was proposed by the researchers of University of Minnesota in 2001 [6]. In item-based model, it is assumed that songs that are often listened together by some users tend to be similar. It is similar to user based except here we find similar songs by using the column of rating matrix for cosine similarity. User-based filtering fails or does not perform that well when a number of users increase like if a number of users are more than a number of songs. In short, it is not scalable. The ratings in item-based model is calculated by [7]

$$P_{u,i} = \frac{\sum_{j \in S} S(i, j) \cdot r_{u,j}}{\sum_{j \in S} |S(i, j)|} \quad (4)$$

The only problem about above equation is that, if the similarity is negative, it is possible that predicted rating can be negative, and we do not want that. This can be

corrected by thresholding the similarity values so that only non-negative values are considered for calculating ratings.

Although collaborative filtering algorithms are very efficient, they have problem. Whenever we come across a new user, we do not have any or very little information about his history, and hence, we won't get good predictions or won't be able to get predictions. This problem is called as cold start problem. As the user listens to some songs, history will be generated and now we can easily use collaborative filtering algorithm for that user.

3.4 Singular Value Decomposition (SVD)

The feature vectors of songs can be of very large dimensions and working on them can be difficult. Hence, we need to find a method for reducing the dimension of these vectors. It was used by the team who won Netflix's million-dollar challenge. SVD is a linear algebra technique which also does one more interesting thing—finds latent (hidden) features of the songs. Some of the features are difficult to find which affect the listening history of users. These factors are not so obvious to find, and hence, they are called as latent features and can be computed by SVD [7–10].

Let us look at mathematical part of the algorithm.

$$R = U \sum T^T \quad (5)$$

where R is $m \times n$ matrix, U is $m \times m$ matrix, T is $n \times n$ matrix, \sum is $m \times n$ diagonal matrix consisting of singular values. The sparse rating matrix R can be decomposed into three matrices U , \sum and T . It is possible to maintain only $k \ll r$ squares of singular values by discarding other entries of S_k diagonal matrix. The reconstructed matrix R' is closest approximation of R . Here, k is number of latent features. Prediction of any song can be calculated assuming mean normalization

Prediction Generation using SVD

$$P_{i,j} = \bar{r}_i + \left(U_k \sqrt{S_k^T(i)} \right) \cdot \left(\sqrt{S_k} \cdot V_k^T(j) \right) \quad (6)$$

where

$P_{i,j}$ is the prediction for i th customer and j th product

r_i is the row average

Once the SVD decomposition is done, the prediction generation process involves only a dot product computation, which takes less time. SVD is time expensive and so can be done offline.

4 Model

We have used a hybrid model consisting of both content and collaborative models and even popularity-based model.

Let us understand the workflow of our system. First, we split the data into training and test set. After this splitting, the training data is fed to some learning algorithm like collaborative filtering which learns to make predictions. To evaluate our system, we use evaluation metrics in which test data is fed to the learned algorithm which in return generates prediction ratings. With help of actual user's rating from test set and evaluation metric, we check how good our model.

Instead of using normal cosine similarity, we have used generalized cosine similarity given by the formula [11, 12]

$$u(c, s) = \cos(\bar{w}_c, \bar{w}_s) = \frac{\bar{w}_c \cdot \bar{w}_s}{||w_c||^\alpha ||w_s||^{1-\alpha}} \quad (7)$$

Whenever we are generating recommendation for any user u, the norm of u terms remains same while calculating similarity with all the other users. Hence, to emphasize the influence of other user we use this form of cosine similarity. So we try to decrease the value of alpha. Range of alpha is between (0, 1).

Now, let us look at our algorithm in which we have combined two or more models by using aggregating method that is combining the end prediction of both the models. If we take $x\%$ recommendation from one model, then we have taken $1 - x\%$ recommendation from another model. We chose x as 0.5 as it worked better for us and we called it as user-item-based model.

Algorithm:

```

input : user id output : list of recommended songs
Get user id for recommendation if user in database:
if no of songs of users > 10 :
    apply user based collaborative filtering algorithm to get x%
    recommendation and item based to get remaining 1-x% recommendation.
else:
    apply item based collaborative filtering algorithm
    to get 80% recommendation
    apply content-based algorithm to recommend remaining
    20% recommendation
else:
    add user in database
    use popularity-based model to get recommendation

```

5 Evaluation Metrics

Evaluation metric is a tool which helps us to understand how well or how good is our recommendation system.

The most common or simple metric is absolute mean error which measures the deviation of the predicted rating from user's actual rating and then averaging over all predicted ratings.

$$|\overline{E}| = \frac{\sum_{i=1}^N |pi - ri|}{N} \quad (8)$$

We also have used precision and recall which are classical metrics for binary classifiers along with MAE [11, 13, 14]. But we have ratings in range 1–5, so we need to find a way to convert this numerical problem into binary problem. We have done this by forming two classes—relevant and not relevant by dividing or splitting the rating into two parts. For ratings equal to 3 and greater, we have assumed it is relevant else it is not relevant. We have used mean average precision in order for evaluating our model. In this metric, first we get top K predictions from our learning algorithm. Then at each rank k , if the song is relevant, we have calculated precision at that k . Let y be the list of predictions given by your system such that $y(j) = i$ means i th song is at rank j in our prediction list [15].

$$P_k = P_{k(u,y)} = \frac{\text{\# of relevant songs in } y_k}{\text{\# of songs in } y_k} \quad (9)$$

We have to average all values of precision for each rank k .

$$AP(u, y) = \sum_{i=1}^k Pi(u, y) \quad (10)$$

Finally, we have to average over all the users.

$$mAP = \frac{\sum_{u=1}^m AP(u, y)}{m} \quad (11)$$

6 Results

We have used mean absolute error for comparing the various algorithms that we implemented for our recommendation system. Above is the graph or results we got for our model. SVD did not perform well with MAE of 0.03612 while user-item based was best with MAE of 0.01825. User-based and item-based models performed the best with MAE of 0.02801 and 0.02215, respectively. Item-based model outperformed user based as user-based model suffered scalability problem (Fig. 1; Table 3).

We have also used mean average precision to test our model as mean absolute error does not test our model that is efficient and superficial. No surprises that popularity

Fig. 1 Mean absolute error of various models

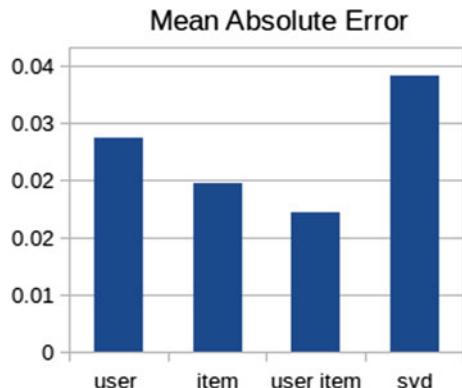


Table 3 Mean average precision for various models

Model	Mean average precision
Popularity	0.0005451
User based	0.0024897
Item based	0.0067313
User-item based	0.0070884

based performed the worst as it is not personal with least mAP. User–item model performed best again with the highest precision as it combined features of both models. User and item-based models performed extremely and were not that far from user–item-based model. Item performed better than user due to scalability issues as a number of users were more than a number of songs.

7 Conclusion

We have implemented popularity-based model, singular value decomposition, KNN algorithm (content-based filtering), user-based and item-based collaborative filtering algorithms. Popularity-based model performed the worst with no doubt as it had no personalization. KNN helped us for content-based filtering, but the only drawback was no variations in recommendations and it becomes very monotonous. User-based and item-based models performed well for us with item-based model performing the best as in user-based model we faced the scalability problem in which a number of users were more than number of songs. SVD too did well although the matrix was too sparse to converge the objective function to global minimum. Hybrid approach outperformed all models as we combined user and item-based filtering and helped us to increase precision with some level.

8 Future Work

Memory models algorithms—collaborative filtering algorithms are very time expensive, and hence, we can use Hadoop to parallelize the computation.

- To combine the user and item-based model by using linear combination and learning the weights.
- To develop algorithms for different aspects like user's mood, current time and day and so on.
- Use deep learning to process the audio files in order to procure features of songs for recommendation.

References

1. Ciocca S. How does Spotify know you so well? <https://medium.com/s/story/spotify-discovering-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe>
2. Million song dataset triplet file. <https://static.turi.com/datasets/millionsong/10000.txt>
3. Asanov D. Algorithms and methods in recommender systems. Berlin Institute of Technology Berlin, Germany
4. Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40:56–58 (Online). <http://doi.acm.org/10.1145/245108.245121>
5. Sun Z, Luo N (2010) A new user-based collaborative filtering algorithm combining data-distribution. In: Proceedings of the 2010 international conference of information science and management engineering, vol 2. ISME '10. IEEE Computer Society, Washington, DC, USA, pp 19–23 (Online). <https://doi.org/10.1109/ISME.2010.48>
6. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on world wide web. WWW '01. ACM, New York, NY, USA, pp 285–295 (Online). <http://doi.acm.org/10.1145/371920.372071>
7. Ricci F, Rokach L, Shapira B, Kantor PB. Recommender systems hand
8. Ekstrand MD, Riedl JT, Konstan JA. Collaborative filtering recommender systems
9. Mishra M, Tata R, Pandey ND. Recommender systems online update. Online pdf
10. Group3_slides.pdf
11. Recall and Precision at k for Recommender Systems. https://medium.com/@m_n_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54
12. Niu F, Yin M, Zhang CT. Million song dataset challenge
13. Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. <https://grouplens.org/site-content/uploads/evaluating-TOIS-20041.pdf>
14. Arguello J. INLS 509: information retrieval evaluation metrics
15. Garg S, Fangyan SUN (2014) 11678 EXY1329 Department of CSE, Indian Institute of Technology, Kanpur, Music recommender system CS365: Artificial Intelligence, Guide: Prof. Amitabha Mukherjee 24 Apr 2014

Chapter 32

Arrhythmia Detection Using ECG Signal: A Survey



Bhagyashri Bhirud and V. K. Pachghare

1 Introduction

According to the news by WHO in 2018, the major reason for death is ischemic heart diseases [1]. The changing lifestyle, unhealthy food leads to various heart diseases, which may lead to heart failure or attack. People are not aware if they are suffering from any disease, until it becomes worse. Suddenly they learn that there is blockage in the arteries.

Arrhythmia is the condition in which the heart rhythm is irregular. It is not life-threatening; however, it may lead to heart failure or attack if not attended properly in time. Different tests and procedures are available, which help to diagnose arrhythmia [2]. These include blood tests, cardiac catheterization, chest X-ray, echocardiography (echo), electrocardiography (ECG), ultrasound, Holter monitor, etc. Out of these tests, ECG is the most commonly used for the diagnosis of arrhythmia [2]. Also, with the help of simple hardware, ECG data can be remotely analysed. This helps in automating the diagnosis of arrhythmia.

Electrocardiography (ECG or EKG) is simple and non-invasive test which records the electrical activity in the heart [3]. Electrodes are placed on skin in order to detect the electrical impulses generated in heart due to depolarization and repolarization of ventricles and atria. It is the simple method which can be remotely performed with minimal hardware setup. Different websites provide the ECG data of patients. With the help of machine learning algorithm, the data can be analysed in order to develop system which would classify ECG signal.

B. Bhirud (✉) · V. K. Pachghare
College of Engineering, Pune, Maharashtra, India
e-mail: bhirudbp17.is@coep.ac.in

V. K. Pachghare
e-mail: vkp.comp@coep.ac.in

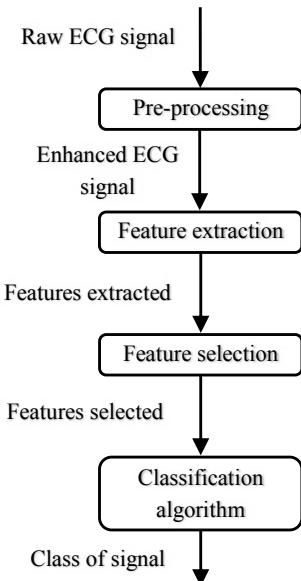
This paper presents techniques for classification ECG signal found in previous work. Different databases used, techniques for pre-processing of signal, feature extraction and selection are discussed. Classification algorithms are also discussed. The work is divided in sections. Section 2 covers literature survey. Section 3 covers the databases used. Pre-processing of ECG signal is discussed in Sect. 4. Different techniques of features extraction are discussed in Sect. 5. Section 6 discusses the feature selection techniques. Classification algorithms used for ECG classification are discussed in Sect. 7. While Sect. 8 concludes the paper.

2 Literature Survey

The literature survey is done with the goal to find out the recent methodologies used in the area of ECG data processing for Arrhythmia detection. It is found that lot of work has been done in this area. Figure 1 shows the basic architecture of ECG signal processing.

Different ECG databases are available, those are discussed further. ECG signals from datasets contain various types of noises, which mandates for pre-processing. Different technologies used for pre-processing of ECG signal are median filters [4], Discrete wavelet transform (DWT) [10], adaptive filters [4, 16], bandpass filters [4, 17, 18], low-pass [4, 8, 9 19–21] and high-pass filters [20, 21], notch filters [20] have been used.

Fig. 1 Architecture for ECG signal classification



Once the ECG signal is filtered, features are extracted from ECG signal in order to characterise those with respect to the classes. Different time domain, frequency domain and morphological features are considered. Techniques like Discrete wavelet transform [10, 12, 16, 17] with Finite Impulse ratio [13], High Order Statistics [6, 7, 10], 1D Convolutional Neural Network [22], Discrete Cosine Transform [11], Wavelet Packet Decomposition [6], Discrete Fourier Transform [6], Principal Component Analysis Network (PCANet) [14], Gaussian Mixture Model [7], Sparse Decomposition method [15] and Stacked Denoised Autoencoder (SDAE) [9] are used for feature extraction.

Few authors have used feature selection or feature reduction techniques in order to reduce the complexity and time required for computation. Principal Component Analysis [10–13, 23, 24], Independent Component Analysis [10, 13], Fast Independent Component Analysis (Fast ICA) [12], Kernel PCA [12], Hierarchical non-linear PCA (hNLPCA) [12], Principal Polynomial Analysis (PPA) [12], Linear Discriminant Analysis [13], Genetic Algorithm [20] and filter-type feature selection method [21] such techniques have been used for feature selection.

After feature extraction and selection, those are fed to classification algorithm in order to classify the ECG signals. Different classification algorithms used include Support Vector Machines [10, 13–21, 23], Neural Network [10, 13, 14, 25] which include Feed Forward Neural Network [18], Probabilistic Neural Network [12, 13], Radial Basis Function Neural Network [15], Convolutional Neural Network [22], Deep Neural Network [9] and Deep Belief Network [4, 6]. K-Nearest Neighbour [11, 14, 15, 18, 24], Random Forest [14, 24], Linear Discriminants [8], Logistic Regression [24] and Ensemble learners [7] also have been used for classification. Optimization techniques like Artificial Bee Colony [15] and Particle Swarm Optimization [15] techniques have also been used to optimise the parameters of classification algorithm.

Evaluation metrics like accuracy, specificity, sensitivity, positive predictivity, false positive rate, F score, precision, detection error rate and area under curve.

3 Dataset

Various datasets are available that provide ECG data of patients. Data is collected from patients with different health conditions, sampled at specific rate. Majorly used databases are discussed here.

3.1 MIT-BIH (*Massachusetts Institute of Technology-Beth Israel Hospital*)

MIT-BIH database contains 48 thirty-minute long records of two-channel, obtained from 47 different patients. The records are digitized at 360 Hz with 11-bit resolution over 10 mV range.

3.2 CU (*Creighton University*)

CU database contains 35 eight-minute long records of patients suffering from ventricular tachycardia, ventricular flutter and ventricular fibrillation. Signals are digitized at 250 Hz with 12-bit resolution over 10 mV range. The signals are filtered with active second-order Bessel low-pass filter.

3.3 AHA (*American Heart Association*)

AHA database contains 80 two-channel records. Signals are digitized at 250 Hz with 12-bit resolution over 10 mV range.

3.4 European ST-T

European ST-T database contains 90 two-hour long annotated records of 79 patients. Signals are sampled at 250 Hz with 12-bit resolution over 20 mV range.

3.5 QT

QT database contains 105 fifteen-minute long two-channel records. The signals are sampled at 250 Hz. According to Association for the Advancement of Medical Instrumentation (AAMI) standard, arrhythmic beats are divided 5 classes. N, S, V, F and Q [38].

4 Pre-processing

For the removal of noise from ECG, pre-processing is the crucial stage. Different types of noise are present in the ECG data, like baseline wander, muscle artefact, power-line interference, high-frequency noise, etc. For removal of baseline, among all methods, the simple, easy and mostly used method is two-stage median filter.

Mathews et al. [4] have used median filter with window size 200 and 600 ms for removing p wave and T wave. Resulted signal, containing baseline wander, is subtracted from original. Dohare et al. [5] used median filter with window/s/2 and/s, where/s is the sampling frequency. Afkhami et al. [7] used two stages of median filter in which artificial sine and cosine wave is generated. De Chazal et al. [8] and Xia [9] used 200 ms width in first stage to remove QRS complexes and 600 ms width in second stage to remove T-wave. The resulting signal subtracted from original which produces baseline corrected signal.

One of the widely used for pre-processing method is Discrete Wavelet Transform (DWT). According to Elhaj et al. [10], DWT is the method which efficiently analyses non-stationary signals. Elhaj et al. [10], Martis et al. [13] and Yang et al. [14] have used Daubechies D6 wavelet basis function. While Desai et al. [11] and Rajagopal et al. [12] used Daubechies mother wavelet of order 8 up to 9 level subbands, which is stated to be the most appropriate for denoising of ECG signal without attenuating the peaks. Instead, according to Raj et al. [15], Daubechies wavelet db4 is good as it is identical to the morphology of ECG signal. They used 4th level decomposition.

According to Mathews et al. [4], adaptive filter with Recursive Least Square (RLS) for removing motion artefacts is more efficient than Least Mean Squares (LMS) algorithm. In contrast, Venkatesan et al. [16] have used LMS as RLS requires more computations and it becomes computationally complex. However, the disadvantage of adaptive LMS algorithm is fixed step size, which affects the performance. To overcome, they have used delayed error normalised adaptive LMS algorithm (DENLMS).

Other techniques like bandpass filters, high-pass filters, low-pass filters and notch filters have been used by authors for elementary pre-processing. Mathews et al. [4] and De Chazal et al. [8] used 12-tap low-pass filter, which is a finite impulse response (FIR) filter for removal of power-line interference and high-frequency noise.

5 Feature Extraction

Feature extraction is the crucial stage of any machine learning system for classification. It highly affects the performance of the classifier, as features extracted discriminates between classes. Most of the authors have used time domain features which are extracted by easy, simple methods by detecting the QRS complex and other wave components (P, T waves). However, many authors say that hidden information

can be extracted in transform domain. This hidden information highly discriminates between classes.

Transform domain techniques are used by most authors. DWT is suited for non-stationary signals as it provides resolution in time and frequency domain at once. Martis et al. [13] used Finite Impulse Response (FIR) approximation of Mayer's wavelet (dmey). They have decomposed a heartbeat into four levels. They have shown that two sub bands of fourth level (approximation and detail) contain discriminatory information from each class. Elhaj et al. [10] have referred the method by Martis. Dimitra et al. [17] used Daubechies order 2 (db2) up to 4 levels of decomposition as done by Übeyli [26]. As claimed by Übeyli [26], Daubechies order 2, that is suitable for detecting change in ECG signal. While Rajagopal et al. [12] used Daubechies order 4 after comparing it with order 2 and 6. Venkatesan et al. [16] used DWT for R peak detection to reduce computational complexity. They used Coiflet wavelet as it is more symmetric and near linear compared to Daubechies.

High order statistics (HOS) is one of the non-linear methods for feature extraction of ECG signal. Gaussian deviation makes it towards the non-linearity of HOS. Third and fourth order statistics have been used by Elhaj et al. [10] as they can represent the non-linear model. Altan et al. [6] also have used 2nd, 3rd and 4th order moments and cumulants as features (such as mean, median, minimum and variance). While Afkhami et al. [7] considered skewness, kurtosis and 5th moment as features.

To reduce the computational complexity, many authors have used just morphological and time domain features. After detecting the QRS complex, based on the characteristics of ECG signal, wave durations and intervals are considered as features. Dohare et al. [5, 23], used 6th power of ECG signal to enhance the complexes which help detect R-peaks. Then used standard deviation for detecting QRS offset, QRS onset, P offset, P onset and T end. Mathews et al. [4] and De Chazal et al. [8] used heartbeat segmentation proposed by [27] which provides the positions of QRS onset, T offset and P onset and offset time. Azariadi et al. [17] used WFDB function wqrs() [28] for detecting exact positions of QRS complexes. Hammad et al. [18] have used Pan-Tompkins algorithm for R peak detection [29].

Kiranyaz et al. [22] used 1D convolutional neural network (CNN) for both feature extraction and classification. They have suggested the modifications to be done in the traditional CNN for 2D image classification. Desai et al. [11] have used Discrete cosine transform (DCT) for feature extraction. They have used first, one third coefficients of DCT as they contain most of variability of signal [30]. Altan et al. [6] have used Wavelet Packet Decomposition (WPD). Along with WPD, they have also used Discrete Fourier Transform (DFT). It transforms time domain samples into frequency domain coefficients.

Yang et al. [14] have used PCANet (Principal Component Analysis Network) [31] for extracting high-dimensional features. It is efficient as it has a smaller number of hyperparameters to adjust. Along with HOS, Afkhami et al. [7] have used Gaussian Mixture Model (GMM) for feature extraction. They calculated 2-component EM (Expectation Maximization) for each signal. According to them, mixture modelling of a heartbeat provides critical information that can be used for classification. Raj et al. [15] used sparse decomposition method for extracting discriminative properties from

signals using over complete gabor dictionary. Xia et al. [9] used stacked denoised autoencoder with sparsity constraint which helps to detect interesting structure in data. Shimpi, Prajwal et al. [24] used UCI Machine Learning Repository [32], dataset of cardiac arrhythmia which has 279 attributes. Those are considered as features.

6 Feature Selection

Feature selection becomes important when feature space is large. Time complexity also increases with size of feature space. Reduction of feature space is performed using different techniques.

Rajagopal et al. [12] compared PCA, Fast ICA (Fast Independent Component Analysis), Kernel PCA (Kernel Principal Component Analysis), hNLPCA (Hierarchical nonlinear PCA) and PPA (Principal Polynomial Analysis). According to them, Kernel PCA was able to provide highest accuracy of all. They used 1–10 principal components (PCs) for analysis of PCA. Fast ICA uses less memory space and is simpler than ICA. Kernel PCA uses kernel matrix instead of covariance matrix for computation. In hNLPCA, PC values and mapping function are provided by neural network. While in PPA uses polynomial curves instead of straight lines in PCA. However, according to them, nonlinear techniques are tuning of parameters, which also causes more time for computation. Martis et al. [13] also used PCA, LDA (Linear Discriminant Analysis) and ICA for dimensionality reduction. They applied PCA, LDA and ICA on approximation and detail sub-band of level four, selecting six features from each. Elhaj et al. [10] used PCA, a linear technique and ICA, nonlinear technique for feature reduction. They used combination of linear and nonlinear features. 12 PCA and 16 ICA features were used in combination for classification.

Principal Component Analysis (PCA) is the most widely used technique for feature selection. It uses covariance matrix, its eigenvectors for calculation of principal components (PCs). Dohare et al. [23] applied PCA on 220 features vector and selected 14 PCs for classification. On the other hand, Desai et al. [11] used PCA to reduce the dimensionality of Discrete Cosine Transform (DCT) coefficients. They considered first 12 PCs as they have more variability of data according to them. Shimpi et al. [24] used PCA to find out features with low dimensions. They chose 150 predictors with least variance out of 279. Li et al. [20] used Genetic Algorithm (GA) for feature selection. They defined a chromosome with 14 element binary vector, an element for each feature. Out of 14, 9 features were selected after 150 repetitions of GA.

Alonso-Atienza et al. [21] used filter-type Feature Selection (FS) method for selecting optimal features. In this method features are ranked according to predefined criterion. Authors used correlation criterion, Fisher criterion and mRMR criterion for ranking of features.

Correlation Criterion—the degree of dependence of individual features with the outcome.

Fisher Criterion—measures the ability of specific feature to discriminate between two sets of labelled data.

mRMR Criterion—aims to maximize the mutual information between outcomes and minimize the redundancy between features.

7 Classification

Once the feature set is ready, a classification algorithm plays the role to classify the input signal. Support Vector Machine is the most widely used algorithm. Different kernels have been used by authors for SVM. Widely used algorithms are discussed further.

7.1 *Support Vector Machines (SVM)*

SVMs basically draw a line separating the objects belonging to different classes. In nonlinear cases, the curve instead of a line is used to separate the objects. SVM-RBF is used by [10, 16, 17, 19, 20, 23, 24] for classification. Elhaj et al. [10] used C parameter equal to 70 and gamma parameter equal to 0.7. In order to rescale attributes in the range $[-1, +1]$, Dohare et al. [23] used Min-Max method. Martis et al. [13] used Least Square SVM proposed by [33]. Yang et al. [14] claim to have faster training and classification speed and highest accuracy among several classifiers with linear SVM with parameter C set to 1. Raj et al. [15] used least square twin SVM (TSVM) proposed by [34]. They found that TSVM as the best classifier model. Also, Shimpi et al. [24] claim that maximum accuracy was obtained by SVM among several classifiers. Hammad et al. [18] compared the performance of their proposed algorithm with linear SVM and polynomial SVM. Alonso-Atienza et al. [21] used filter-type feature selection method with SVM and implemented FS-SVM procedure.

7.2 *Neural Network (NN)*

A neural network consists of neurons arranged in layers (input, hidden and output) which convert the input vector to output vector.

Feed Forward Neural Network—FFNN is the simplest NN. It does not have any loops while processing from input layer to output layer via hidden layers. According to Elhaj et al. [10] FFNN can classify ECG signals more accurately. They used 28 nodes in input layer, 40 neurons in hidden layer and five neurons at output layer with back propagation. Ochoa et al. [25] also used a three-layered four FFNNs with back-propagation. They used transfer functions as sigmoidal hyperbolic, logarithmic tangential and linear. Martis et al. [13] used fully connected FFNN for classification and it provided the highest performance for LDA (Linear Discriminant Analysis) features among the compared algorithms. Yang et al. [14] also used BP-NN

(Back Propagation Neural Network) which is a feed-forward network with error back propagation. Authors considered two hidden layers.

Probabilistic Neural Network—PNN is feed-forward neural network which has four layers, viz input, hidden, summation and output layer. Rajagopal et al. [12] used PNN with radial basis function as transfer function. Along with FFNN, Martis et al. [13] also used PNN for classification which performed well for ICA (Independent Component Analysis) features.

Radial Basis Function Neural Network—RBFNN is the artificial neural network which uses radial basis function as activation function. Raj et al. [15] used RBFNN for classification of ECG signals.

Convolution Neural Network—CNN is a variation of multilayer perception, which require minimal processing. Kiranyaz et al. [22] used adaptive 1D CNN for classification. They used 3 CNN layers and 2 MLP layers. They used shallow training with two terminating criteria:

- i. max number of BP iterations is set to 50.
- ii. min train classification error level is set to 3% to avoid over-fitting.

Deep Neural Network—DNN is the neural network with more than two layers. Basically, it tries to mimic the connection in human brain. Xia et al. [9] used DNN with softmax regression at the supervised learning stage in DNN. They also used active learning for the fine tuning of DNN. Mathews et al. [4] used deep learning framework including Restricted Boltzmann Machine (RBM) and Deep belief Networks (DBN) for classification. For these algorithms, authors have used the toolbox developed by Wulsin [35]. Deep Belief Network—DBN is a type of DNN consisting of multiple layers of hidden units. The hidden layers are connected to each other. Altan et al. [6] used DBN which consists of stacked RBMs (Restricted Boltzmann Machines). A multistage architecture is used with four DBNs. In first stage class N is classified from S, V, F and Q classes. Second stage classifies S class from V, F and Q and so on. For hidden layers hyperbolic tangent function was used as activation function.

7.3 K-Nearest Neighbour

K-NN is the algorithm which is used for classification as well regression. It is non-parametric algorithm which indeed makes it simple and a choice amongst all available algorithms for machine learning. Desai et al. [11] used K-NN algorithm k equal to 2 and for neighbour's search Euclidian distance metric is used. Yang et al. [14] also used KNN for classification. They found that k equal to 5 gives the best performance. Also, Raj et al. [15] used this algorithm for classification. They used Minkowski distance metric for neighbour's search. Shimpi et al. [24] also used this algorithm for classification.

7.4 Others

Random Forest—RF is an ensemble learning method which uses multiple decision trees to predict the output class. Yang et al. [14] used RF with 500 number of trees. Shimpi et al. [24] used RF for classifying UCI dataset with already extracted features.

Logistic Regression—LR is used for classification as against linear regression.

It uses logistic function (sigmoid function). Shimpi et al. [24] used LR on UCI dataset. LR is helpful when there are two classes. In case of multiple classes, linear discriminant analysis is used. It uses statistical method to find linear combination of characteristics which separates classes from each other. De Chazal et al. [8] used classifiers based on linear discriminants. They used two LDs each for a set of features from a lead of ECG out of two. They combined the output of two LDs to give the final decision.

On the other hand, Hammad et al. [18] proposed a classifier based on the characteristics of normal waves and intervals of ECG signal. The performance of the proposed classifier model was compared with FFNN, SVM (Linear and non-linear) and KNN.

Ensemble Learners—Ensemble learners are another effective classifier which combines multiple base learning algorithms to yield a stronger learner (algorithm). They provide higher accuracy than the constituent learning algorithms. Afkhami et al. [7] used bootstrap aggregating ensemble method. The method combined 100 decision trees. Authors claim that statistical advantages of ensemble learners help to classify unevenly distributed arrhythmia classes.

7.5 Optimization Techniques

Optimization techniques are used to optimize the parameters of classifier. These techniques help in reducing the time for a result with good accuracy. Raj et al. [15] used two types optimization techniques discussed below:

Artificial Bee Colony—Artificial bee colony (ABC) is a population-based algorithm that was introduced by Karaboga in 2005 [36]. In this algorithm, there are three types of bees, i.e., employee bee, onlookers and scouts, food source. Here, the food source represents the possible optimization solution. The amount of food from sources indicates the quality of that solution. Number of employee bees is the number of solutions available.

Particle Swarm Optimization—This technique (PSO) was introduced by Kennedy and Eberhart in 2001 [37]. This algorithm is based on the flocking birds. When a flock of birds is searching an area for food, the bird closest to the food chirps loudly and all others swing towards the bird. This way the tightening pattern is followed until one of the birds catches the food. This logic is used behind particle swarm optimization.

8 Conclusion and Future Challenges

Several researchers have suggested improvements to be included in future work. Those are discussed here. Mathews et al. [4] have considered the time domain features with Deep Belief Network. According to them, other type embedding can be used as feature vector of ECG. Li et al. [19] suggests that additive noise models do not provide results close to reality, as the real time ECG data has nonlinear noise and distortions. Altan et al. [6] used DBN with the objective to govern the efficiency of DBN, instead of acquiring best accuracy. Yang et al. [14] suggest using dimensionality reduction techniques to improve recognition accuracy. Also, very few researchers have used dimensionality reduction techniques.

Methods and techniques discussed in the literature review are implemented using MATLAB. We suggest machine learning using python, as very less work is done with python in arrhythmia detection. Not all researchers followed AAMI (Association for the Advancement of Medical Instrumentation) standard while detecting arrhythmia. We suggest using AAMI standards while developing methodology for arrhythmia detection.

References

1. The Top 10 causes of death, Accessed on 24 May 2018, <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
2. Arrhythmia, <https://www.ncbi.nlm.nih.gov/health-topics/arrhythmia>
3. Electrocardiography, <https://en.wikipedia.org/wiki/Electrocardiography>
4. Mathews SM, Kambhamettu C, Barner KE (2018) A novel application of deep learning for single-lead ECG classification. Comput Biol Med 99:53–62
5. Dohare AK, Kumar V, Kumar R (2014) An efficient new method for the detection of QRS in electrocardiogram. Comput Electr Eng 40(5):1717–1730
6. Altan G, Allahverdi N, Kutlu Y (2018) A multistage deep learning algorithm for detecting arrhythmia. In: 2018 1st international conference on computer applications & information security (ICCAIS), pp 1–5. IEEE
7. Afkhami RG, Azarnia G, Tinati MA (2016) Cardiac arrhythmia classification using statistical and mixture modeling features of ECG signals. Pattern Recog Lett 70(2016):45–51
8. De Chazal P, O'Dwyer M, Reilly RB (2004) Automatic classification of heartbeats using ECG morphology and heartbeat interval features. IEEE Trans Biomed Eng 51(7):1196–1206
9. Xia Y, Zhang H, Lin Xu, Gao Z, Zhang H, Liu H, Li S (2018) An automatic cardiac arrhythmia classification system with wearable electrocardiogram. IEEE Access 6:16529–16538
10. Elhaj FA, Salim N, Harris AR, Swee TT, Ahmed T (2016) Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals. Comput Methods Progr Biomed 127(2016):52–63
11. Desai U, Martis RJ, Nayak CG, Sarika K, Nayak SG, Shirva A, Nayak V, Mudassir S (2015) Discrete cosine transform features in automated classification of cardiac arrhythmia beats. In: Emerging research in computing, information, communication and applications, Springer, New Delhi, pp 153–162
12. Rajagopal R, Ranganathan V (2017) Evaluation of effect of unsupervised dimensionality reduction techniques on automated arrhythmia classification. Biomed Signal Process Control 34:1–8

13. Martis RJ, Acharya UR, Min LC (2013) ECG beat classification using PCA, LDA, ICA and discrete wavelet transform. *Biomed Signal Process Control* 8(5):437–448
14. Yang W, Si Y, Wang D, Guo B (2018) Automatic recognition of arrhythmia based on principal component analysis network and linear support vector machine. *Comput Biol Med* 101:22–32
15. Raj S, Ray KC (2018) Sparse representation of ECG signals for automated recognition of cardiac arrhythmias. *Expert Syst Appl* 105(2018):49–64
16. Venkatesan C, Karthigaikumar P, Paul A, Satheeskumaran S, Kumar R (2018) ECG signal pre-processing and SVM classifier-based abnormality detection in remote healthcare applications. *IEEE Access* 6(2018):9767–9773
17. Azariadi D, Tsoutsouras V, Xydis S, Soudris D (2016) ECG signal analysis and arrhythmia detection on IoT wearable medical devices. In 2016 5th international conference on modern circuits and systems technologies (MOCAST), pp 1–4. IEEE
18. Hammad M, Maher A, Wang K, Jiang F, Amrani M (2018) Detection of abnormal heart conditions based on characteristics of ECG signals. *Measurement* 125:634–644
19. Li Q, Rajagopalan C, Clifford GD (2014) A machine learning approach to multi-level ECG signal quality classification. *Comput Methods Programs Biomed* 117(3):435–447
20. Li Q, Rajagopalan C, Clifford GD (2014) Ventricular fibrillation and tachycardia classification using a machine learning approach. *IEEE Trans Biomed Eng* 61(6):1607–1613
21. Alonso-Atienz F, Morgado E, Fernandez-Martinez L, García-Alberola A, Rojo-Alvarez JL (2014) Detection of life-threatening arrhythmias using feature selection and support vector machines. *IEEE Trans Biomed Eng* 61(3):832–840
22. Kiranyaz S, Ince T, Gabbouj M (2016) Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Trans Biomed Eng* 63(3):664–675
23. Dohare AK, Kumar V, Kumar R (2018) Detection of myocardial infarction in 12 lead ECG using support vector machine. *Appl Soft Comput* 64(2018):138–147
24. Shimpli P, Shah S, Shroff M, Godbole A (2017) A machine learning approach for the classification of cardiac arrhythmia. In 2017 international conference on computing methodologies and communication (ICCMC), pp 603–607. IEEE
25. Ochoa A, Mena LJ, Felix VG (2017) Noise-tolerant neural network approach for electrocardiogram signal classification. In proceedings of the international conference on compute and data analysis, pp 277–282. ACM
26. Übeyli ED (2007) ECG beats classification using multiclass support vector machines with error correcting output codes. *Digit Signal Proc* 17(3):675–684
27. ecgpuwave, Accessed on 28 Nov 2018, <https://www.physionet.org/physiotools/wag/ecgpuw-1.htm>
28. wqrss, Accessed on 28 Nov 2018, <https://www.physionet.org/physiotools/wag/wqrss-1.htm>
29. Pan J, Tompkins WJ (1985) A real-time QRS detection algorithm. *IEEE Trans Biomed Eng* 32(3):230–236
30. Ahmed N, Natarajan T, Rao KR (1974) Discrete cosine transform. *IEEE Transa Comput* 100(1):90–93
31. Shi J, Wu J, Li Y, Zhang Q, Ying S (2017) Histopathological image classification with color pattern random binary hashing-based PCANet and matrix-form classifier. *IEEE J Biomed Health Inform* 21(5):1327–1337
32. UCI Machine Learning Repository: Arrhythmia Data Set, Accessed on 2017, <https://archive.ics.uci.edu/ml/datasets/Arrhythmia>
33. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. *Neural Process Lett* 9(3):293–300
34. Tomar D, Agarwal S (2015) A comparison on multi-class classification methods based on least squares twin support vector machine. *Knowl-Based Syst* 81:131–147
35. Wulsin D (2010) DBN Toolbox v1. 0. Department of Bioengineering, University of Pennsylvania
36. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Vol. 200. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department

37. Kennedy J, Eberhart R (2001) Swarm intelligence
38. ANSI/AAMI, Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms, American National Standards Institute, Inc. (ANSI), Association for the Advancement of Medical Instrumentation (AAMI), ANSI/AAMI/ISO EC57, 1998-(R)2008, 2008

Chapter 33

Towards Designing the Best Model for Classification of Fish Species Using Deep Neural Networks



Pranav Thorat, Raajas Tongaonkar and Vandana Jagtap

1 Introduction

Underwater image processing introduces additional challenges absent otherwise. Due to exponential attenuation of light, as it travels through water, poor visibility is characteristic of underwater images. The resulting scenes are hazy and poorly contrasted. A subset in marine ecosystems includes the study of underwater living organisms like fishes, crabs, octopuses, whales, seals, etc. Monitoring these organisms in terms of their number and type over time can be useful for the purpose of studying the effect of environmental changes or human actions on these organisms. Therefore, the need of a vision-based system for the purpose of underwater surveillance is much required. This has led us onto the path of building one such robust system. Here, specifically, we are targeting a vision system capable of underwater fish species detection. These vision systems can be an integral part of underwater vehicles like autonomous underwater vehicle (AUVs) and remotely operated vehicles (ROVs) which can survey large areas of the water bodies non-stop and can provide with useful insights with minimum human intervention. Convolutional Neural Networks have been a useful tool for image detection and classification since 2012. Starting with models like the AlexNet to the recently perceived models like YOLOv3, there have been different CNN algorithms and architectures which have successfully solved the problem of localization and classification with ever-increasing classification speed or accuracy or both. In this paper, we will describe our approach in the classification

P. Thorat (✉) · R. Tongaonkar · V. Jagtap

Department of Computer Engineering, Maharashtra Institute of Technology, Pune, India

e-mail: thoratpranav@gmail.com

R. Tongaonkar

e-mail: tonraaj@gmail.com

V. Jagtap

e-mail: vandana.jagtap@mitpune.edu.in

of fish species using two different convolutional neural network architectures. The first one is our own “scaled-down VGG-16” architecture and the other one is the “traditional VGG-16” model. Our intention behind using two different models is to compare the accuracy of the models and the time required for training on our dataset.

2 Related Previous Work

Some previous research work has been described below: Some of the non-machine learning-based approaches like imaging sonar have also been used by researchers for detection and classification of fishes. The accuracy of this model [1] is measured using a set of manually annotated sonar images.

Ramani et al. [2] used parallel networks of three perception layers to classify four species of fish in sonar images. Storbeck et al. [3] used a three-layered convolutional network to devise a way of classifying moving fish. Marburg et al. [4] devised a method for identification of 10 classes of benthic macrofauna in optical images using stacked CNNs.

Some researchers have performed detection of underwater objects using neural networks. Byeongjin et al. [5] used Haar-like features for object detection in water. Juhwan et al. [6] detected and tracked remotely operated vehicle in sonar images using a convolutional neural network design. We have also referred to the Google Colabatory paper [7] which explains the hardware specification of Colabatory along with its implementation. Also provides a performance review of deploying computer vision, deep learning, classification and other applications on Colab.

For object detection, many researchers have used very deep convolutional network models like the VGG-16 [8], which forms the base of our project, have used networks of increasing depth with small convolutional filters which performed extremely well in the ImageNet challenge of 2014. The official VGG-based models like VGG-16 and VGG19 performed the classification task better than its predecessor AlexNet.

Some researchers [9] have performed real-time fish detection on images from actual fish videos using convolutional neural network-based techniques based on you only look once (YOLO) algorithm. The network recorded 93% classification accuracy and outperformed older non-CNN-based models like classifier trained with histogram of oriented gradient features and support vector machine and sliding window algorithm. The drawback of YOLO classifier is that the accuracy is compromised at the expense of better detection speed. As mentioned earlier, here, we are using two CNN-based architectures for classification. Following is the detailed explanation of our network for the fish species classification.

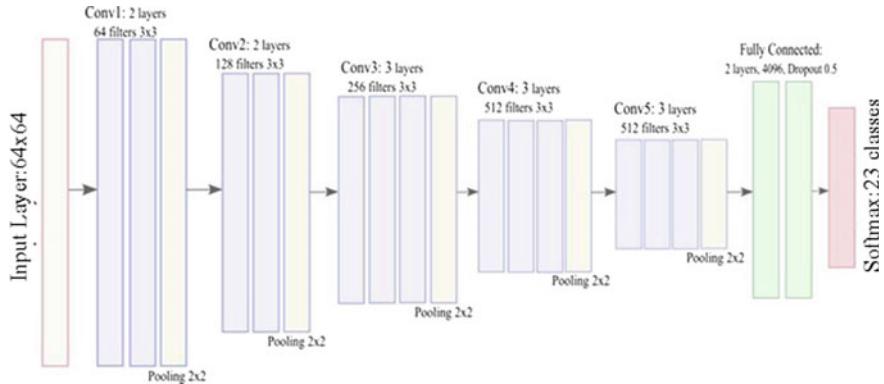


Fig. 1 VGG-16 architecture

2.1 Previous Architecture

With reference to [8], we have implemented the traditional VGG-16 architecture by Visual Geometry Group at the University of Oxford. While training, the input images to our network are resized $64 * 64$ RGB images. These images are passed through stacked convolutional layers which are thirteen in number with the number of filters varying in an increasing order in each layer as defined in the original paper and shown in the adjoining figure. The number of filters for each convolutional layer is mentioned in the architecture diagram. Parameter to each convolutional layer includes a small $3 * 3$ filter or kernel. Rectified Linear Unit or “ReLU” is the activation function used. The convolutional stride set to 1 pixel throughout. Five max-pooling layers perform spatial pooling. It is performed over a $2 * 2$ pixel window (Fig. 1).

The convolutional layers are then followed by a set of three fully connected layers. The first two have 4096 channels each and the third one has 23 channels (one for each class). This final layer is the softmax layer. Softmax is implemented using a layer just before the output layer. The softmax layer must have the same number of nodes as the output layer.

3 Proposed Method

In CNN like architecture, selection of parameters like number of layers and arrangement of layer plays an important role in the efficiency, accuracy and performance of our model. Selecting the optimum value of these parameters gives us the best model and helps us avoid tackling problems like model overfitting and excessive time and memory usage during training the neural network.

Here, we are using a scaled-down version of the VGG-16 architecture. The similarity between our architecture and the VGG-16 architectures is that both of them are

uniform and use a small $3 * 3$ kernel (or filter) in the convolutional layer. Here, the size of the filter (or kernel) is the same (i.e. $3 * 3$) throughout the architecture for each convolutional layer which results in reduced number of parameters as compared to a layer using filters of larger size like $7 * 7$ (parameters = 49) or $11 * 11$ (parameters = 121). Hence, we call it the VGG like architecture.

Also, the layers in the architecture are such arranged that the initial layers are the combination of convolutional layers (along with their activation function “ReLU”) and the pooling layer, in the end, followed by a fully connected layer for class prediction along with its probability value.

With respect to the convolutional layers, our model is partly made up of repetitive stacking of a convolution layer followed by the activation layer and batch normalization. Batch normalization is used for the purpose of normalizing the activations of the previous layer at each batch. For representational purposes, this set is referred to as one unit. Each unit is followed by a pooling layer of window size $2 * 2$. After each pooling layer, a dropout of 0.25 states the fraction of dropped input unit from the previous layer. After the stack of convolutional layers, next are the two fully connected layers. The first fully connected layer has 512 channels. A dropout of 0.5 in this layer resembles the fraction of input units which are to be dropped. The last layer is the softmax implemented through a fully connected layer just before output layer. The softmax layer has the same number of nodes as the output layer, in our case 23 as we dealing with 23 classes (Fig. 2).

This algorithm/model we used for the classification of fish species was a scaled-down version of the known VGG-16 algorithm. The VGG-16 architecture has a total of 16 layers but our scaled-down VGG-16 used a mere 8 layers, 6 convolutional layers and 2 fully connected layers. This particular model can be termed as VGG-8. The reason behind using a smaller network is to significantly reduce the time and memory required during training, taking into consideration that the accuracy is not compromised massively as compared to larger convolutional neural network architectures. Also, smaller networks are easier and faster to train, hence can be built where hardware resources are a major constraint.

4 Methodology

Our dataset has been taken from the Fish4Knowledge Project at the University of Edinburgh, School of Informatics, UK. This fish data is acquired from a live video dataset resulting in 27370 verified fish images in total belonging to 23 different fish species.

Following are the steps in which we train our model on the dataset. Initially, we load our dataset in the primary memory and resize each image to a $64 * 64$ pixel resolution. Next, we split our dataset into training and testing sets, with 75% of the total in training and remaining 25% of the dataset for testing. Next, we initialize our CNN model with appropriate parameters and define the hyperparameters and train the network.

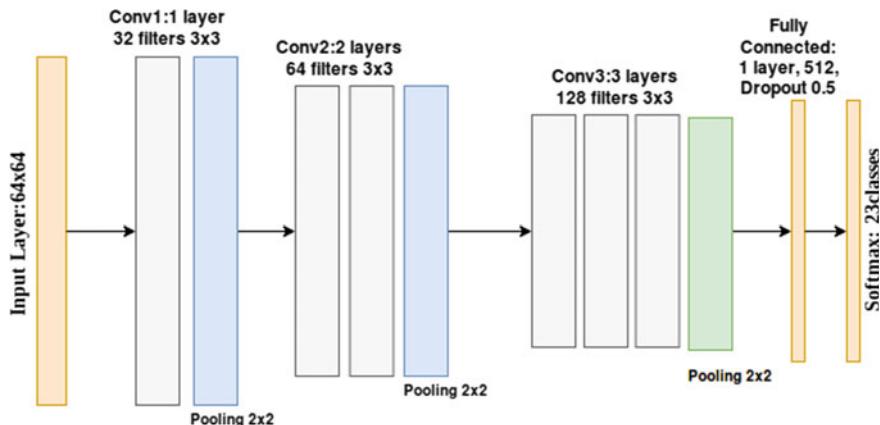


Fig. 2 VGG-8 architecture

Table 1 Hyperparameters

Epochs	Batch size	Learning rate
75	32	0.01

We have used Google Colabatory for training purpose. Google Colabatory is a cloud service based on Jupyter Notebooks which provides access to a robust GPU. The hardware available is comparable to a mainstream workstation and a robust Linux server equipped with 20 physical cores.

For our application, in Google Colabatory, we trained our model on the “Python 2 Google Compute Engine Backend (GPU)” and a total of 12.72 GB of RAM and 358.27 GB of disk space were allocated.

Two paths were taken:

1. VGG-8: This was basically a smaller version of the VGGnet algorithm. It had 6 convolutional layers and 2 fully connected layer
2. VGG-16: This was the implementation of the actual VGG-16 algorithm. It has 13 convolutional layers and 3 fully connected layers.

In both these implementations, the number of epochs was the same 75. We kept the batch size 32 with an initial learning rate of 0.01 (Table 1).

The activation function for both implementations was rectified linear unit (ReLU), with the final fully connected layer having softmax activation function.

5 Results

It can be observed from both the graphs that as the iterations during training increases there is a gradual decrease in the loss parameters, i.e. the training loss and validation

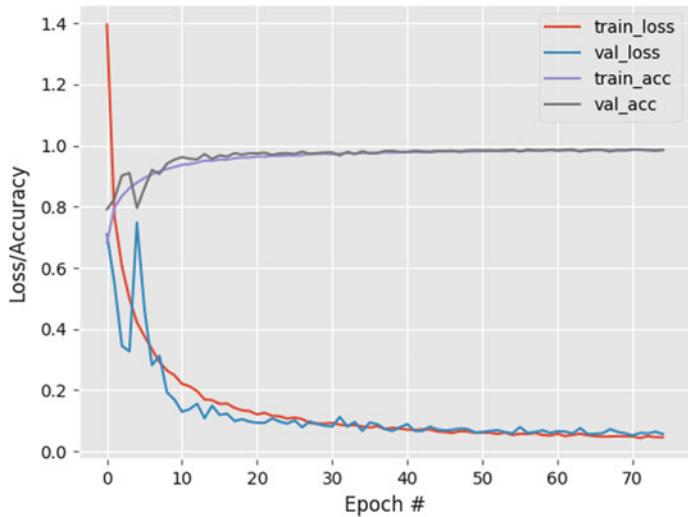


Fig. 3 VGG-16

loss values. On the other hand, accuracy parameters like the training accuracy and the validation accuracy increase gradually as the number of epochs increase. We have also tried to choose the optimum value of the hyperparameters such that the model does not overfit.

VGG-16 (Fig. 3) had a higher initial training loss than VGG-8 (Fig. 4). Loss value implies how poorly or how well a certain model behaves after each iteration

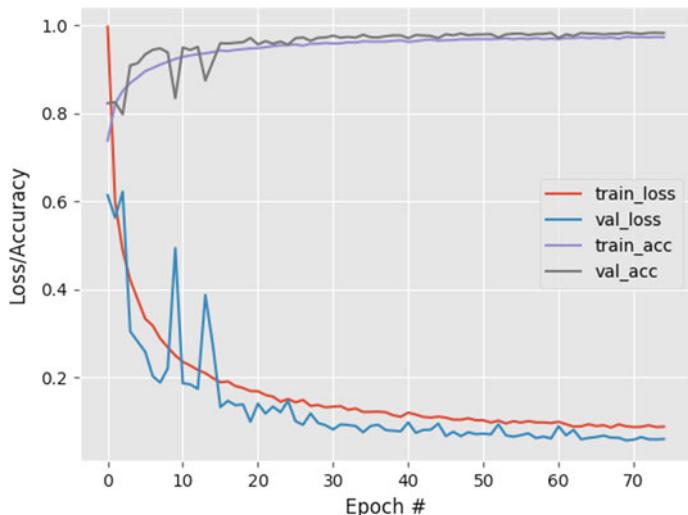


Fig. 4 VGG-8

of optimization. Training loss is the average of the losses over each batch of training data. Testing or validation loss is computed using the model as it is at the end of an epoch. It was almost equal to training loss for VGG-16 and lower than training loss for VGG-8. Both achieved a training accuracy and validation accuracy greater than 0.95 by the 50th epoch.

Overall, VGG-16 performed better than VGG-8. However, the increase in performance was marginal. VGG-8 is thus a better option in situations where there is a restriction on the resources available for training the model.

The table shows the average value for the given metric for the 23 classes.

Macro average computes the metric independently for each class and then takes the average. Hence, all classes are treated equally.

Micro average on the other hand will aggregate the contributions of all classes to compute the average metric.

In weighted average, instead of each data point contributing equally to the average, some data points contribute more than others.

VGG-8 has a marginally higher macro average value of the performance metric of precision. VGG-16 scores higher in all other metrics. VGG-16 is thus only marginally better than VGG-8.

For testing purpose, random images of the fish species “Amphiprion Clarkii” downloaded from Google images and tested on our model, which it correctly classifies and gives a significant accuracy of 98.25% on the VGG-8 model. Whereas, the same image when tested using the VGG-16 classifier, it correctly classified the species but with a lesser accuracy of 96.07%. One of the possibilities that might explain this result is that a larger network might have led to overfitting. Hence, we can conclude that a larger network might not always lead to better results as observed in this case. Thus, our proposed architecture is better suited for this dataset than the standard VGG-16 model. Following is a comparative study of the two models (Table 2).

Table 2 Performance metrics

	VGG-16	VGG-8
Micro average	Recall: 0.99 Precision: 0.99 F1-score: 0.99	Recall: 0.98 Precision: 0.98 F1-score: 0.99
Macro average	Recall: 0.89 Precision: 0.94 F1-score: 0.90	Recall: 0.86 Precision: 0.96 F1-score: 0.90
Weighted average	Recall: 0.99 Precision: 0.99 F1-score: 0.98	Recall: 0.98 Precision: 0.98 F1-score: 0.98

6 Conclusion and Future Scope

In this paper, a CNN-based approaches for fish classification were implemented and analyzed. Specifically, two models VGG-16 and VGG-8 were trained on our dataset. Both models showed similar performance in terms of various performance measures. The differentiating factor between both the models is the time required for training and the amount of space used up. The VGG-8 model is better than VGG-16 considering time and memory utilization. Hence, models like the VGG-8 can be used when there is a restriction in the hardware resources available with the user. Also, such lighter models can be used in simple classification problems in which not much detailing in the images is required for classification, whereas, denser models like VGG-16 can be used in much problems requiring much precise detailing/analysis of images for classification.

We have also tested our model on images which are outside of the dataset and the results are pretty satisfactory and accurate which is the first step in building a robust system.

Hence, the above model can be deployed on an AUV or ROV, wherein as the underwater vehicle moves around it will pick up images and if fishes are present, will be correctly classified and these results can be used for analyzing the water body by the biological researchers. We have recently built our own AUV in collaboration with another team and will soon deploy our classification module on the AUV.

In future, the above model can be tested more rigorously. The system can be trained to work efficiently in more challenging background environments and also in low lighting conditions. Also, this model can be improved to be built on more uniformly distributed images in the dataset with respect to each class.

Reference

1. Wolff LM, Badri-Hoehler S (2014) Imaging sonar-based fish detection in shallow waters, Oceans - St. John's
2. Ramani N, Patrick PH (1992) Fish detection and identification using neural networks-some laboratory results. IEEE journal of oceanic engineering 17(4):364–368
3. Storbeck F, Daan B (2001) Fish species recognition using computer vision and a neural network. Fish Res 51(1):11–15
4. Kim B, Yu SC (2017) Imaging sonar based real-time underwater object detection utilizing AdaBoost method. Underwater Technology (UT), 2017 IEEE. IEEE
5. Kim J, Yu SC (2016) Convolutional neural network-based real-time ROV detection using forward-looking sonar image. Autonomous Underwater Vehicles (AUV), 2016 IEEE/OES. IEEE
6. Marburg A, Bigham K (2016) Deep learning for benthic fauna identification. OCEANS 2016 MTS/IEEE Monterey. IEEE
7. Carneiro T, Da Nóbrega RVM, Nepomuceno T, Bian GB, De Albuquerque VHC, Rebouas Filho PP. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access (Accepted but unpublished)
8. Simonyan K, Zisserman A (2015) Very Deep Convolutional Neural Networks for Large Scale Image Recognition. ICLR

9. Sung M, Yu SC, Girdhar Y (2017) Vision based real-time fish detection using convolutional neural network. OCEANS 2017– Aberdeen
10. Convolutional Neural Network Wikipedia - The Free Encyclopedia

Chapter 34

A Study on Attribute-Based Predictive Modelling for Personal Systems and Components—A Machine Learning and Deep Learning-Based Predictive Framework



Aswin Ramachandran Nair, M. Raj Mohan and Sudhansu Patra

1 Introduction

The PS business unit is a key growth area for the company and maintaining high service levels without compromising on inventory being held is of utmost importance to meet growth and revenue expectations. However, PS business has been facing recurring shortages, overstocking of slow-moving SKUs/components and increased obsolescence instances owing to poor forecast accuracy and absence of long-term demand visibility.

1.1 Current Forecasting Scenario

The regional demand planning team uses conventional forecasting methods and limited-time series models to arrive at final predictions. The approach works well for certain segments and regions with high predictability but fails with numerous instances leading to inaccurate predictions and poor forecast accuracy. Rapid transition in technology along with SKU proliferation adds more complexity into the PS forecasting business.

A. R. Nair · M. Raj Mohan · S. Patra (✉)
HP Inc Supply Chain Analytics, Bangalore, India
e-mail: sudhansu_patra@yahoo.com

A. R. Nair
e-mail: ashwinrn@yahoo.com

M. Raj Mohan
e-mail: rajmohanm2006@gmail.com

1.2 Personal System Portfolio

Some of the challenges faced while forecasting for the PS portfolio are region/segment complexity, product proliferation, short product life cycle and high volatility (Fig. 1).

2 Solution

The traditional time series forecast methods work good for commercial HP notebook business but fail when we focus on consumer business where there are irregular shipment and order patterns. Hence, we rely on advanced methods such as machine learning and deep learning algorithms for better forecast accuracy. The overall forecasting model for PS business has been consolidated which is known as E3P solution which includes four modules: explorer, predictor, profiler and planner.

As the first step, we group individual families into clusters based on similar technical specifications and this is done mostly by analyzing at HP notebook brand, panel size and processor levels. The categorization is done even at other levels like gaming/non-gaming and clamshell/non-clamshell segments in case of multiple families with the same brand, panel size and processor. The clustering of families is done at explorer module and clusters thus formed will have shipment history and customer orders for at least five years for high volume families. Enough shipment history/customer orders help time series models in predicting trend, level and seasonality and thereby forecast high volume with high accuracy.

Once we finalize the family cluster group, we provide the inputs into predictor module of PS forecasting model. Predictor consists of group of statistical forecasting methods which use time series, machine learning and deep learning models. The initial step in the forecasting process includes time series model which includes methods like Holt-Winters, ARIMA, Theta and random walk forecast (RWF). The best possible time series method is selected among the possible ones by validating forecast accuracy for historical one-year period. Majority of family clusters demand can be predicted with high accuracy using the time series methods. We use Eq. 1 to



- Region/Segment Combination
- Product Proliferation
- Short Product Life Cycle
- High Volatility

Fig. 1 Schematic of PS portfolio and major challenges

calculate the forecast error and Eq. 2 to in general.

$$\text{Absolute Forecast Error} = \frac{|\text{Forecasted Demand-Shippments}|}{\text{Shipments}} \quad (1)$$

$$\text{WMAPE} = \frac{(\sum_{M=1}^{12} (\text{Absolute Forecast Error} \times \text{Shipments}))}{\sum_{M=1}^{12} \text{Shipments}} \quad (2)$$

where WMAPE is the weighted mean absolute percentage error.

The remaining family clusters will be forecasted using advanced techniques which use attribute-based forecasting by considering machine learning and deep learning algorithms.

As of now, attribute-based forecasting uses internal factors such as customer order pattern, cannibalization, etc., and not external factors like market trends. The paper focuses only on family level forecasting and selects notebook brand, panel size and processors which we call as independent variables and these are used to predict dependent variables like factor shipments/customer orders. We do not have future information on attributes and these are forecasted for the next one year using conventional time series models. The forecast accuracy for independent variables is calculated to be very high which is approximately 85–90%. The forecasted attribute demand will be given as input to machine learning and deep learning models which will be forecasted based on weights given to independent variables. The machine learning model uses methods such as support vector machine (SVM), decision tree algorithms and random forecast. The deep learning model uses artificial neural networks (ANN) which will estimate weights based on backward propagation method.

The above methods are used for the clustered family groups with enough shipment history for the model to train and then test the results. But for those family groups with NPI or insufficient shipment history use an alternate attribute-based forecasting which is part of profiler module. The profiler module uses combination of regression and correlation methods to cluster the family group with insufficient shipment history with other family groups based on the ranks. The highest-ranked family group shipment history will be given as input to the forecasting model based on profiler and generates the demand for the family group with insufficient shipment history.

Like time series models, the WMAPE calculated for profiler, machine learning and deep learning models is validated during the historical 12 months period. The best fit method which is known as the baseline forecast is selected not only based on highest accuracy among time series, machine learning and deep learning models but also on parameters depending on planner module which will be discussed below.

The baseline forecast is fed into planner where the forecasting team visualizes the forecast along with shipments/orders before making any manual adjustments and discussing with regional demand planning teams.

Figure 2 depicts the process flow of the forecasting model and Fig. 2 depicts the overall four-step forecasting framework (E3P forecasting framework).

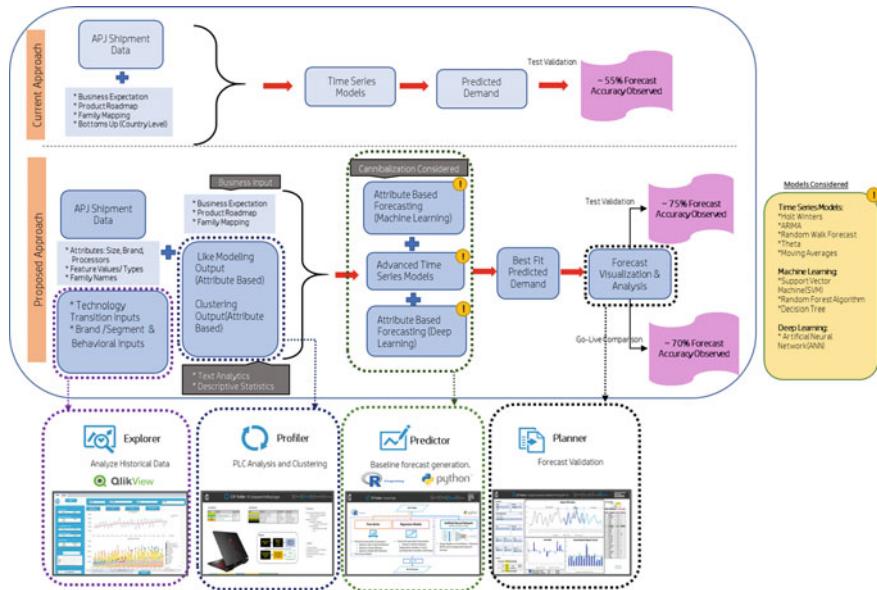


Fig. 2 Process flow—forecasting model and E3P forecasting framework

2.1 Framework and Modelling Details

Under the current approach, the regional planning teams use shipment/orders (as transactional data) and a few other factors are inputs to the demand planning process. These are analyzed using limited-time series models and the outputs from these models are considered the final forecast after some basic validation. In this section, we detail the overall forecasting framework for PS business as the “**E3P forecasting framework**” which includes four modules: **explorer**, **predictor**, **profiler** and **planner (E3P)**.

- **Explorer** is a QlikView/Power BI-based visualization dashboard which enables unstructured data analysis to identify underlying trends from historical transactional data, viz. technology transition trends, and behaviour analysis across attributes, segments, geography and brand, etc. It is important to group laptop families based on brand, panel size, processors and gaming type before feeding into profiler and predictor which are the forecasting engines.
- **Profiler** module enables predecessor/successor identification and attribute-based clustering powered by text analytics and descriptive statistics. Profiler uses the Levenshtein distance (LD)¹ to quantify and classify features or attributes enabling predecessor/successor mapping and like modelling. Levenshtein distance (LD) is the measure of similarity between two strings: the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions

required to transform s into t . The greater the Levenshtein distance, the more different the strings are.

The Profiler also has clustering capabilities which identify valid clusters which mirror the behaviour of the SKU/component being forecasted.

- **Predictor** module is the forecasting engine which uses various time series (R), machine learning (R) and deep learning (Python) algorithms to generate the best fit forecast using the minimum WMAPE method. The initial step in the forecasting process uses time series model which includes methods like Holt-Winters, ARIMA, Theta and random walk forecast (RWF). We use attributes as independent variables and these are forecasted for the next one year using conventional time series models. The forecast accuracy for independent variables is calculated to be very high which is approximately 85–90%. The forecasted attribute demand is given as input to machine learning and deep learning. The machine learning model uses methods such as support vector machine (SVM), decision tree algorithms and random forecast. The deep learning model uses artificial neural networks (ANN) which will estimate weights based on backward propagation method. The overall forecasts generated using the multiple methods go to another module which is known as best fit analyzer which selects the best fit method based on minimum WMAPE during validation period (Fig. 3).
- **Planner** is the final dashboard module which assists the forecast validation process with forecast visualization against historical actuals and regional forecast. The dashboard has also been enabled with beacons and metrics to identify outliers, high BIAS SKUs, volume-based analysis, etc. The planner module helps in adjusting the baseline forecast to include specific deals for different regions and accounts by collaborative planning with regional planners. This module is also used for comparison among baseline forecasts generated in different months and comparison of baseline forecast with regional forecast. This helps in analyzing any deviations of baseline forecast generated in a month with other forecasts and take corrective action accordingly.

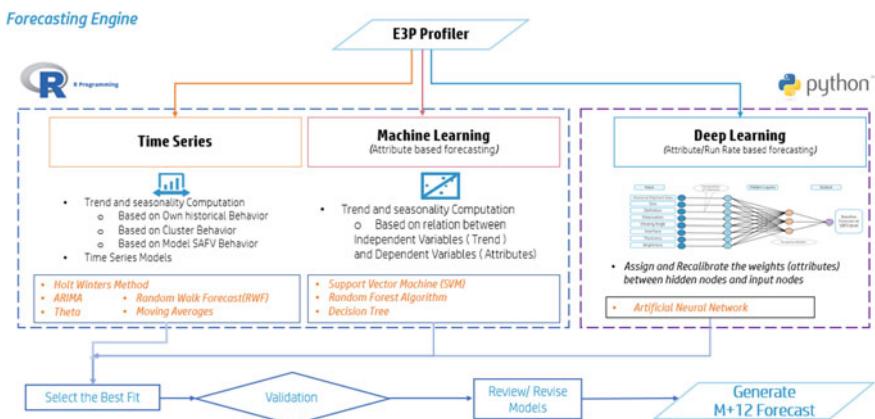


Fig. 3 E3P predictor process flow

Fig. 4 Testing period results

Region	Segment 1			Segment 2		
	Region Forecast	Baseline Forecast	Improv %	Region Forecast	Baseline Forecast	Improv %
Region 1	56%	70%	14%	50%	67%	17%
Region 2	75%	80%	5%	64%	73%	9%
Region 3	59%	70%	11%	35%	64%	29%
Region 4				57%	63%	6%

Fig. 5 Go live results

Region	Segment 1			Segment 2		
	Region Forecast	Baseline Forecast	Improv %	Region Forecast	Baseline Forecast	Improv %
Region 1	55%	64%	9%	46%	52%	6%
Region 2	50%	54%	4%	62%	77%	15%
Region 3	63%	66%	3%	30%	56%	26%
Region 4						

2.2 Evidence the Solution Works

The proposed solution greatly enhanced the forecasting/demand planning framework. These are some results from the testing (*January 2018 to December 2018*) Fig. 4 and go live (*January 2019*) Fig. 5; periods at family level.

2.3 Competitive Approaches

Forecasting seasonal footwear demand using machine learning [1] by Majd Kharfan demonstrates the value of forecast customization based on product characteristics. The model assumes the data availability of future product characteristics and this is necessary for demand forecasting of textiles. In case of rapid change in PC technology, this information might not be available, and the attribute-based statistical forecasting method will be useful.

3 Current Scenario

The solution has been designed and developed which includes time series and machine learning models in *R*, deep learning models in Python, the input databases in SQL Server and all the modules been embedded using net technology. QlikView/Power BI solution has been deployed to cater to all the visualization requirements. The overall solution has been deployed in HP Cloud which helps different users to access the solution simultaneously. The current scope of the project is limited to notebooks (base unit level) and commodities (panels and processors).

We are in the process of extending the forecasting solution to other commodities like graphics processor unit, battery, etc., and other business-like desktops.

4 Next Steps

Along with validating and enhancing the forecasting solution, the team is expected to work on increasing the existing portfolio to graphics cards, hard disk (commodities) and workstations (base unit) in the coming months. The team will also explore the possibility of forecasting at lower levels where we see attach rate complexities and potential to use machine learning.

References

1. Forecasting Seasonal Footwear Demand Using Machine Learning by Majd Kharfan, Bachelor of Economics, Accounting, Damascus University, 2011 & Vicky Wing Kei Chan Bachelor of Business Administration, Global Supply Chain Management, The Hong Kong Polytechnic University, 2011
2. Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach Rishin Haldar and Debajyoti Mukhopadhyay Web Intelligence & Distributed Computing Research Lab Green Tower, C-9/1, Golf Green, Calcutta 700095, India

Chapter 35

Text Categorization Using Sentiment Analysis



Chaitanya Bhagat and Deepak Mane

1 Introduction

Twitter Sentiment Analysis may, along these lines, be portrayed as a content digging procedure for breaking down the fundamental conclusion of an instant message, i.e., a tweet. Twitter conclusion or supposition communicated through it might be certain, negative or impartial. In its basic form, Twitter is a micro-blogging service that allows users to post brief text updates, with the unique property of not allowing more than 140 characters in one text message. This limitation turned out to be a very attractive property since it allows posting quick, even current time, updates regarding one's activities and facilitates sharing and forwarding status messages, as well as replying to them quickly. The objective is to develop a system that can effectively identify what emotion the writer wants to express and classify the given tweet into the respective class. It has a positive impact or a negative one, whether the news spread is correct or false, etc. This ecosystem presents a very rich, source of data to mine. Nonetheless, because of the constraint as far as characters (i.e., 140 characters per tweet), mining such data present lower performances than that when mining longer texts. In addition, classification into multiple classes remains a challenging task: Binary classification of text usually relies on the sentiment polarity of its components (i.e., whether they are positive or negative); whereas, when positive and negative classes are divided into subclasses, the accuracy tends to decrease remarkably. In this system, Get authenticate tweet one by one from Twitter. Check first that tweet contains the number of positive words according to positive word label classification and the number of negative word label classification till the total number of a tweet

C. Bhagat (✉) · D. Mane
Department of Computer Engineering, PICT, Pune, India
e-mail: chaitanyab29@gmail.com

D. Mane
e-mail: dtmane@gmail.com

from an individual tweet. Also, count the total number of positive and negative words from all tweets. Then using Naïve Bayes and K-Nearest Neighbors (KNN), find out the probability of each positive word with the total number of positive words, negative word. If the count of positive words is greater than negative word, classify as positive and this tweet as a positive tweet; negative words are greater than positive word, classify this tweet as a negative tweet. In this new system, we propose an approach that relies on writing patterns and special unigrams to classify tweets into eight different subclasses and demonstrate how the proposed approach presents good performances. Our scope of the project is tweet classified using Naïve Bayes and K-Nearest Neighbors into three main categories and eight subcategories with more accuracy than the existing system. We observe in [5, 6] that social media is more effective than journalism and printing media. We notice that people write comments to express their emotion on any issue or achievement. Calculate the most active category on Twitter like sport, celebrity and politician. Calculate today's generation busy with Twitter and also categorizations of tweet according to particular user tweet multiclass sentiment analysis approach.

The rest of the paper is arranged as follows: Sect. 2 presents related work, Sect. 3 presents the proposed system approach and Sect. 4 presents experimental results and Sect. 5 concludes the paper.

2 Literature Survey

Bouazizi and Ohtsuki [1] proposed a novel approach to classify sentiment into binary and ternary classes, that is, main classes. And also classify text collected from Twitter into seven subclasses. For this, they introduced SENTA tool to the user to select features and then run classification into seven different classes (i.e., fun, happiness, love, neutral, sadness, anger, hate). For classification, they used Random Forest classifier on textual Twitter data and achieve accuracy 70.1% on ternary classes and 60.2% on seven classes. Mohammad and Kiritchenko [2] proposed hashtag feeling vocabulary strategy are utilized to produce a substantial dictionary of the word—feeling relationship from this feeling named tweet corpus and further order on manual marks of feelings in tweets. Regardless, on account of the limitation to the extent characters in this article, exhibit that feeling word hashtags are incredible manual characteristics of emotions in tweets. In these papers, informational collection is utilized and hashtag feeling vocabulary, the SVM strategy is utilized. A method to deliver a far-reaching vocabulary of the word—feeling relationship from this inclination checked tweet corpus. This is the important vocabulary with the certifiable regarded word—feeling alliance scores. Begin with examinations for six fundamental emotions and show that the hashtag clarifications are dependable and facilitate with the remarks of arranged judges. The framework shows how the removed tweet corpus and word—feeling affiliations can be used to improve feeling gathering exactness in another no tweet area. Identity may be connected with any of the few emotions and in light of

the fact that our hashtag approach scales easily to incalculable, widen our corpus by social occasion tweets with hashtags identifying with 585 fine sentiments.

Plank and Hovy [3] introduced the utilization of Internet-based life as an asset for substantial scale, open vocabulary identity discovery like data, for example, sexual orientation, number of supporters, statuses or list participation, include important data. Cerebrum science looks at recommendations that specific personality characteristics identify with semantic direct. This relationship can be sufficiently exhibited with quantifiable trademark vernacular taking care of strategies. Desire accuracy, generally, upgrades with greater data tests, which furthermore think about increasingly lexical features. Most existing work on personality desire, regardless, fixates on little models and shuts vocabulary examinations. The two factors compel the comprehensive explanation additionally, the quantifiable force of the results. In this framework, sex controlled dataset and natural language processing (NLP) method are accustomed to exploring the use of online life as a benefit for extensive scale, open vocabulary personality area. Analyze which features are insightful of which character characteristics and present a novel corpus of 1.2 M English tweets remarked on with Myers–Briggs personality make and sexual introduction. Our preliminaries exhibit that Internet organizing data can give satisfactory semantic confirmation to reliably predict two of four-character estimations.

Goel et al. [4] state that tweets can be characterized utilizing SentiWordNet alongside Naive Bayes for characterization into various classes dependent on their significance with the point searched. Twitter1 is a scaled down scale blogging Web site which gives a phase for people to share and express their viewpoints about subjects, happenings, things and distinctive organizations. Tweets can be orchestrated into different classes relying upon their significance with the subject looked for. Distinctive machine learning computations are starting at now used arranged by tweets into positive and negative classes reliant on their estimations, for instance, Baseline, Naive Bayes Classifier, Support Vector Machine, etc. Twitter contains utilization of Naive Bayes using sentiment 140 getting ready data using Twitter database and proposes a system to make progress portrayal. This framework comprises a large movie review dataset (LMRD). In this continuous feeling investigation order approach is utilized. Usage of SentiWordNet close by Naive Bayes can upgrade precision of course of action of tweets, by giving vitality, negativity and objectivity score of words present in tweets. For real execution of this structure, Python with NLTK and Python-Twitter APIs are used. Garimella and Mihalcea [5] presented sexual orientation distinction via web-based networking media. The surface-level content characterization ways to deal with sex separation and endeavored to pick up bits of knowledge into the contrasts among people by utilizing semantic strategies that can point to remarkable word classes or contrasts in idea use. Information is gathered from we gathered from the BlogSpot look again at the issue of sexual orientation segregation and endeavor to move past the run of the mill surface-level content arrangement approach, by (i) recognizing semantic and psycholinguistic word classes that reflect efficient contrasts among people and (ii) discovering contrasts between sexes in the manners in which they utilize similar words. The framework depicts a few tests and report results on a

vast gathering of Web sites composed by people. Trust these qualifications at a more profound semantic dimension.

Bamman and Smith [6] to the investigation of sarcasm and other discourse follow up via web-based networking media locales with complex groups of onlookers, and this contextualized sarcasm recognition on Twitter is utilized. Most computational ways to deal with mockery recognition, in any case, treat it as an absolutely semantic issue, utilizing data, for example, lexical signs and their relating conclusion. Sarcasm requires some shared learning among speakers and get-together of individuals; it is an essentially significant marvel. Most computational approaches to manage mockery area, regardless, see it as a basic semantic issue, using information, for instance, lexical prompts and their relating supposition as judicious highlights. We show that by including extra etymological information from the particular circumstance of a verbalization on Twitter, for instance, properties of the maker, the gathering of spectators and the brief open condition—we can achieve gains in exactness appeared differently in relation to completely phonetic features in the disclosure of this unpredictable wonder, while also uncovering knowledge on features of social joint effort that enable sarcasm in discourse. Huq et al. [7] introduced two methods: One of the techniques is known as sentiment classification algorithm (SCA) based on K-Nearest Neighbor (KNN), and another one is based on Support Vector Machine (SVM).

3 Proposed System Approach

In this proposed system, tweets are collected from Twitter by using Twitter API. Create positive words dataset and negative word dataset for checking the positive label class and negative label class. After collecting the tweet first, classify the label classification according to positive word dataset and negative word dataset. Get authenticate tweets one by one from Twitter. Check first that tweet contains the number of positive words according to positive word label classification and the number of negative word label classification till the total number of the tweet from an individual tweet. Also, count the total number of positive and negative words from all tweets. Then using Naïve Bayes and K-Nearest Neighbors (KNN), find out the probability of each positive word with the total number of positive words, negative word. If the count of positive words is greater than negative word, classify as positive and this tweet as a positive tweet; negative words are greater than positive word, classify this tweet as a negative tweet. If any tweet contains the number of positive word and number of the negative word is same, then that tweet is categorized into a neutral category and also finds out the probability of each negative word with the total number of negative words. According to that probability and count number of words, tweet's main category as a positive, negative and neutral, respectively. In that case also, find out the probability of each word to result of main category result class (i.e., positive, negative and neutral class.) Then using Check first that tweet contains the number of love words, number of happiness word, number of fun words, number of neutral words, number of hate word, number of sadness word, number of anger word and

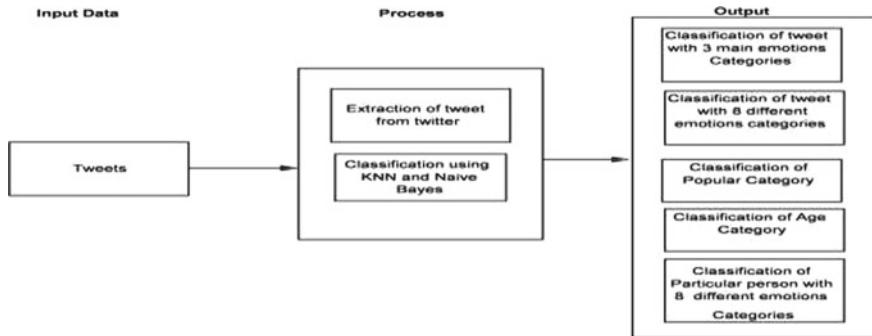


Fig. 1 Block diagram of proposed system

number of excitement word till total number of the tweet from an individual tweet. After that find out the total number of each word in the love word class category, the total number of each word in the happiness word class category and all remaining different categories. In this case, find out the total number of words according to that subclass category in particular tweet. We also classify tweets into popular category like celebrity, sports and politics. Using this approach system can get a better result than the existing system. User and admin can perform various kinds of functionality like view classification of the tweet into three main categories. The proposed methodology text categorization using Naive Bayes and K-Nearest Neighbor algorithm can be analyzed for its implementation details with the below-mentioned steps (Fig. 1).

Step 1: Dataset collection in which one is the dictionary that contains a set of words corresponds to each class. Other dataset consists of tweets for training and testing.

Step 2: Pre-processing—once the tweets are extracted, it is pre-processed. To remove unwanted text and symbols.

Step 3: In Naive Bayes here, all pre-processed tweets are taken as input, and each word of a tweet is considered as a token. Compare the token to the words in the dictionary. If any word is a match then find the probability of word with respective class.

$$P(X/C_i) = \prod_{k=1}^n P(X_k/C_i) \quad (1)$$

$$P(X/C_i)P(C_i) > P(X/C_j)P(C_j) \quad (2)$$

where $i \leq j \leq m, j \neq i$.

Step 4: K-Nearest Neighbor probability values are used to calculate Euclidean distance between training data and testing data.

$$X_{ij} = \sqrt{\sum_{k=1}^n (X_{ik} - X_{jk})^2} \quad (3)$$

Step 5: After checking the majority of voting from k nearest sample and classify tweet into class.

4 Experimental Results

To run our project, we required a hardware system which is feasible for our project like Intel i3 processor, 4 GB RAM, 100 GB hard disk. We also need standard keyboard, mouse and LED monitor. The system can use Microsoft as the operating system platform. To run this application, we need JDK 1.7 and above as Java platform and Apache Tomcat as server. To store data, we need MySQL database. Our project is run on Firefox, Chrome or any browser.

To evaluate the performance of this approach, we are exploring the accuracy of this approach. **Accuracy** is the fraction of predictions our model got right. It is the ratio of correctly classified data samples over the total number of data samples.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

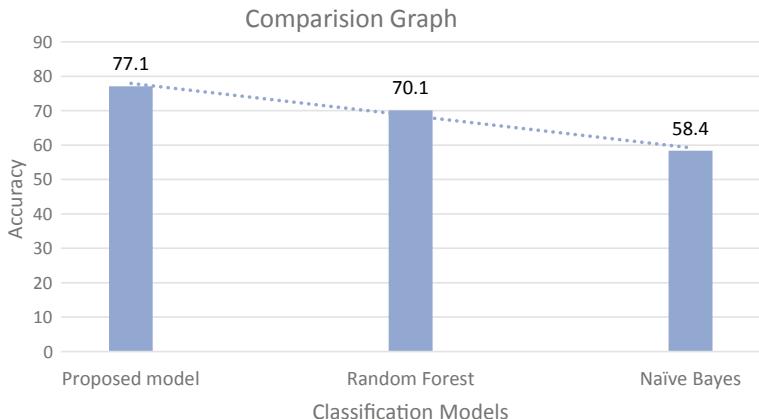
In our experimental setup, as shown in Table 1, the total numbers of tweets are 21,000. These tweets are then divided into three main categories; among which 10,563 are positive, 5628 are negative and rest 4809 are neutral. We have divided 21,000 tweets in the ratio of 80:20 where 80% is the training data and 20% is testing data.

In the year 2016, Goel et al. [4] got 58.40% accuracy by using Naïve Bayes classifier on 1.6 million tweets, and tweets are classified into positive and negative classes. After that in the year 2017, Bouazizi and Ohtsuki [1] got 70.10% accuracy by using Random Forest classifier on 21,000 tweets, and tweets are classified into positive, negative and neutral class (Graph 1).

In the proposed approach, we are getting 77.10% accuracy on main class tweet classification that includes positive, negative and neutral class. In the existing approach, the Naïve Bayes algorithm is used to calculate probability and based on probability class is assigned. And in another approach, Random Forest is used in

Table 1 Comparison of results

Category	Number of tweets
Positive	10,563
Negative	5628
Neutral	4809

**Graph 1** Comparison graph**Table 2** Comparison of results

Model	Accuracy (%)
Proposed model	77.10
Random forest	70.10
Naïve Bayes	58.40

which the majority of voting is taken and those class having the majority of votes that class is assigned to a given tweet. In the proposed approach, we are using a hybrid approach of Naïve Bayes and K-Nearest Neighbors in which we first calculate the probability of tweet belong to which class and store probability value. When a new tweet comes, we calculate probability and then calculate Euclidean distance between previously stored probability value and new tweet probability value. Finally, take the majority of the vote and based on the majority of the voting class is assigned.

In the existing system, accuracy of classification of tweets into main class is comparatively less than the proposed approach (i.e., the classification accuracy of main class in the existing system is 70.1% and the proposed approach having 77.10%) (Table 2).

5 Conclusion

There are various approaches available for text classification with some advantages and disadvantages. In the proposed system, we analyze the sentiment of different tweets from Twitter. Here, tweets are categorized into three main categories that are positive, negative and neutral by using Naïve Bayes and K-Nearest Neighbor classifier. By using a hybrid approach of Naïve Bayes and K-Nearest Neighbor

classifier, we can achieve good accuracy than other existing approaches. In the future, by using this system we analyze and detect the emotion from text data on different social media like Facebook and Instagram.

References

1. Bouazizi M, Ohtsuki T (2017) A pattern-based approach for multi-class sentiment analysis in twitter. *IEEE Access* 5:20617–20639
2. Mohammad SM, Kiritchenko S (2015) Using Hashtags to capture fine emotion categories from tweets. *Comput Intell* 31:301–326
3. Plank B, Hovy D (2015) Personality traits on Twitter—or—how to get 1500 personality tests in a week. In: Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp 92–98
4. Goel A, Gautam J, Kumar S (2016) Real time sentiment analysis of tweets using Naive Bayes. In: 2nd International conference on next generation computing technologies (NGCT-2016), Dehradun, India, pp 257–261
5. Garimella, A., Mihalcea, R.: Zooming in on Gender Differences in Social Media. In: Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media Osaka, Japan, pp. 1–10 (2016)
6. Bamman D, Smith NA (2015) Contextualized sarcasm detection on twitter. In: Proceeding of the 9th international AAAI conference on web and social media. Citeseer, pp 574–577
7. Huq MR, Ali A, Rahman A (2017) Sentiment analysis on twitter data using KNN and SVM. In: Proceeding IJACSA, pp 19–25

Chapter 36

Automated Real-Time Email Classification System Based on Machine Learning



Sudhir Deshmukh and Sunita Dhavale

1 Introduction

Many corporates like IT firms, commercial institutes (like investment banks), manufacturing and process industries receive a large amount of unstructured data from various sources like existing/potential customers, vendors and internal communication within the organization, queries related to product/service, other private and government organizations, etc. In most of the organizations, these unstructured emails are manually classified according to the expertise needed to tackle and act on the information content of the email with the help of dedicated customer service representatives. However, the massive nature of incoming emails makes this approach tedious, error-prone and time-consuming.

Thus, there is a need for an automated real-time system that can analyze and classify the email based on text content, its contextual meaning and actions required. Classified email can be automatically routed to specific teams for further appropriate action.

A lot of work has been undertaken in this direction [1–9]. Research started with a rule-based system (searching for a specific keyword) and gradually shifted to the machine learning-based classification systems [2, 3], which are effective for email classification. In [1], the authors compared keyword-based classifier with Naive Bayes classifier and found that Naive Bayes classifier outperforms even in the case of small training dataset. In [4], the authors used a vocabulary-based approach along with effective stemming algorithms to detect both text and image-based spam emails.

S. Deshmukh · S. Dhavale (✉)

Department of Computer Science and Engineering, Defence Institute
of Advanced Technology (DIAT), Girinagar, Pune 411025, India
e-mail: sunitadhavale@diat.ac.in

S. Deshmukh
e-mail: bokeydeshmukh@gmail.com

In [5], the authors have applied several techniques for enhancing the effectiveness of Naïve Bayes classifier for text classification problem.

This paper establishes some common trends and directions in this research space. This paper is organized as follows: Sect. 2 elaborates the proposed methodology, Sect. 3 presents experimental results and analysis, followed by the conclusion in Sect. 4.

2 Proposed Methodology

Figure 1 shows the proposed machine learning-based email classification system. Different modules present in the system are explained below.

The interface between email client and classification system, which is created with Python code and Gmail API, connects with an email client and feeds the email to the classification system (in HTML form). The same interface will also be used to route the email after classification. Content extractor extracts and combines relevant textual content from an email, viz. subject and the body of an email. In the preprocessing module, the goal is to clean and make the data ready for constructing features that can be passed to the machine learning algorithm. In feature construction and representation module, feature vectors are constructed out of the textual content and are represented in such a way that a machine learning algorithm can understand (textual content needs to be represented in numbers). The classifier is nothing but a machine learning algorithm that has to be trained on the data available to achieve good performance on an unseen dataset (unseen email). The trained classifier is then used in real time to predict, to which email ID a particular email should be routed.

Developing this model poses various practical questions like

- What preprocessing techniques should be applied?
- From textual data how the feature should be constructed and represented?
- What classification techniques should be applied?
- Do interaction term and dimension reduction techniques improve performance?

To answer these questions, exhaustive experiments are performed.

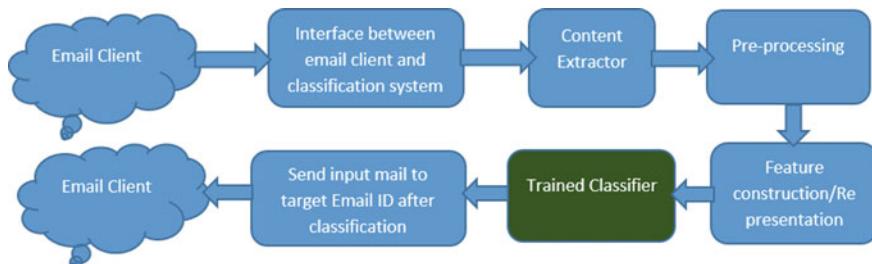


Fig. 1 Structure of machine learning-based email classification system

3 Experiments and Results

The experiments are performed in order to find a suitable classifier, feature representation and construction technique for the proposed email classifier. Effect of different preprocessing methods (viz. stop words removal, stemming, converting text to single letter case, different tokenization schemes) on the performance of the classifier is investigated. Effect of interaction terms and dimension reduction in the performance of classifier is also analyzed. Best parameters (hyper-parameter tuning) for each classifier are chosen using grid search (exhaustive search over all possible values) over a range of parameters, and multiclass classification is achieved through the One-verses-Rest strategy (training separate classifier for each class).

All the experiments are carried out on two different datasets, viz. 20 news group [10] (only four classes are considered) and demo email dataset (consisting of more than 300 emails created to demonstrate the effectiveness of this system in real time).

3.1 Analysis of the Performance of Different Classifiers

Classifiers including logistic regression [11], multinomial Naïve Bayes [5, 8], SVM [12, 13], decision tree [14], random forest [15] and KNN [6] are evaluated against two different datasets. Figures 2 and 3 show the comparison in performance for various classifiers.

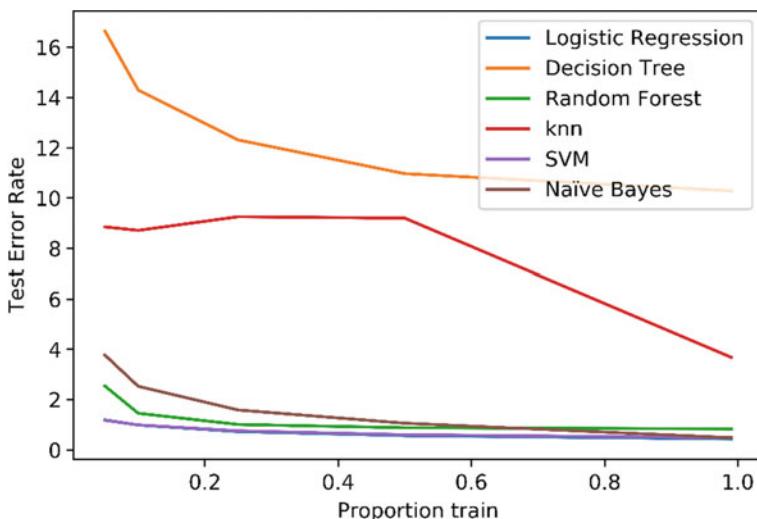


Fig. 2 Graph representing the test error rate for various proportions of training data. In this case, measure of test error rate is taken as log loss. Dataset used is 20 news group dataset

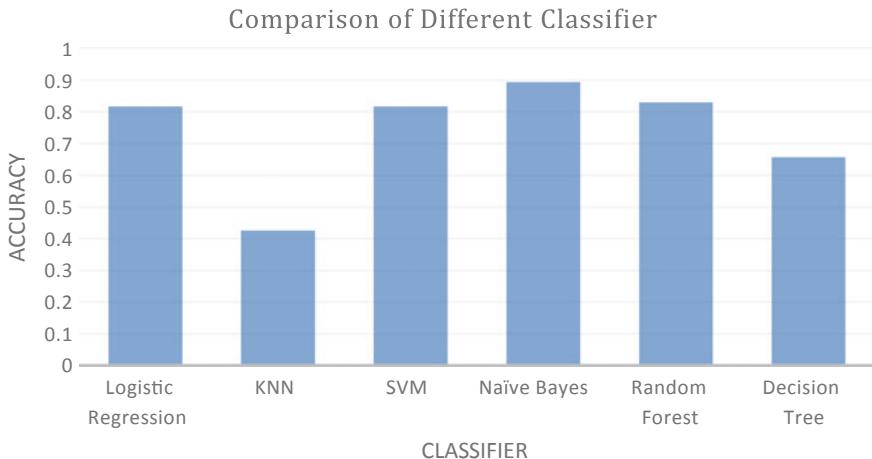


Fig. 3 Bar chart showing a comparison between different classifiers based on score/accuracy. Dataset used is 20 news group dataset

Results. Logistic regression, SVM, Naïve Bayes and random forest perform well for 20 news group dataset in terms of accuracy as well as log loss (cross-entropy loss). Further, SVM and random forest are computationally expensive. Thus, during training, they perform quite slowly. Training becomes further slow and computationally expensive with a grid search for SVM and random forest. Naïve Bayes gives the best score, but logistic regression performs best in terms of log loss. So, if one has to be very confident about their prediction, they should use logistic regression. Otherwise, Naïve Bayes is best for real-time text classification systems both in performance and computational efficiency.

Learning curves (Figs. 4 and 5) depict that proposed model using both Naïve Bayes and logistic regression as a classifier do not have high bias (or under-fitting) problem. But with logistic regression, we may have a little of high variance (or over-fitting) problem. In order to tackle high variance (or over-fitting) problem, the number of training examples can be increased along with selecting a smaller (but more informative) set of features.

3.2 Analysis of Feature Construction/Representation Techniques

The performance of the classification model is tested for feature construction using unigram, bigram as well as unigram and bigram combined (N-gram) [9]. Feature representation techniques are compared for Count Vectorizer (count of the occurrences of every token in each document is represented in feature vector), TF-IDF Vectorizer [7] and Hash Vectorizer [16]. Figures 6 and 7 show the variation in accuracy for

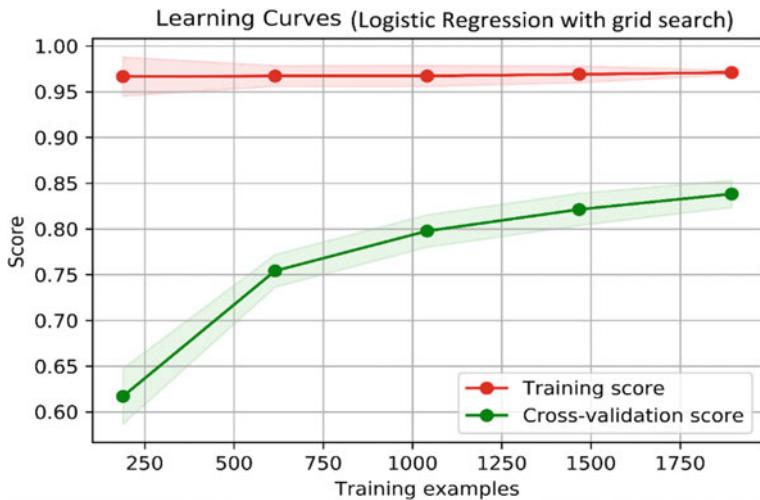


Fig. 4 Learning curve for logistic regression. Dataset used is 20 news group dataset

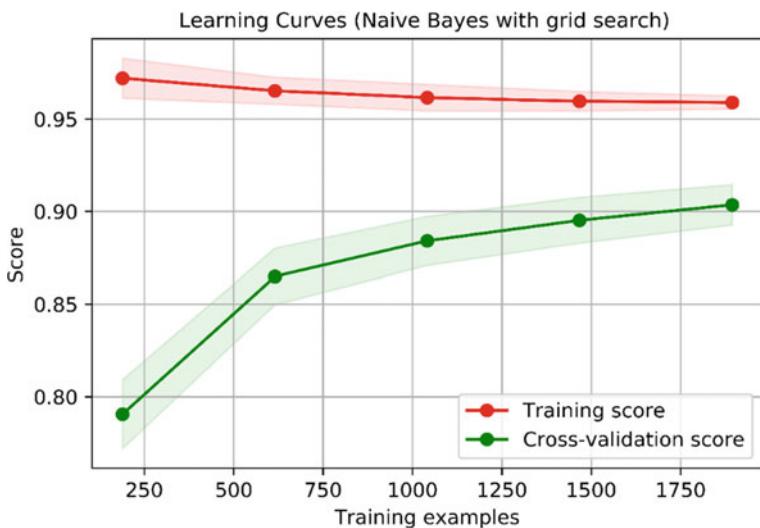


Fig. 5 Learning curve for Naïve Bayes. Dataset used is 20 news group dataset

various feature representation and construction techniques, respectively.

Result. From the observation on both datasets, we can say that TF-IDF performs best both in terms of accuracy and log loss. And Hash Vectorizer is suitable than simple Count Vectorizer in case of a text classification problem. Using Hash Vectorizer also improves computational efficiency.

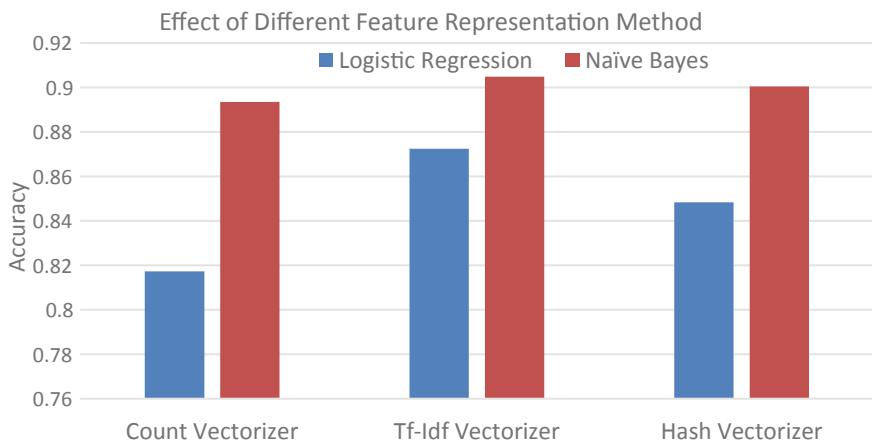


Fig. 6 Bar graph comparing different feature representation techniques on the basis of score/accuracy for both Naïve Bayes and logistic regression classifier. Dataset used is 20 news group dataset

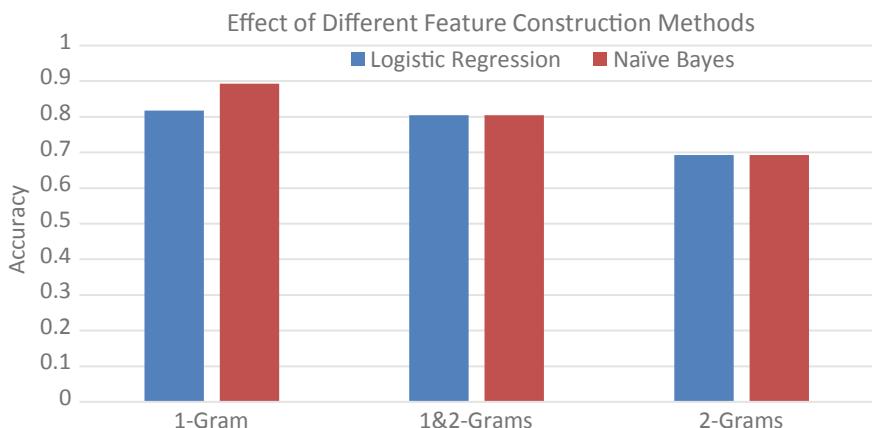


Fig. 7 Bar graph comparing different feature construction techniques on the basis of score/accuracy for both Naïve Bayes and logistic regression classifier. Dataset used is 20 news group dataset

Bag-of-word representation with unigram feature vector outperforms both feature vector with bigram and feature vector with unigram and bigram together. We can argue that constructing a feature vector with n-gram ($n > 1$) may confuse classifier in case of the text classification system as ours resulting in poor performance when the number of features is fairly large.

Thus, unigrams are used to construct feature vectors, and TF-IDF measure is used to represent tokens in the feature vector for the final classification system.

3.3 Analysis of Preprocessing Techniques

Effect of different preprocessing methods (viz. stemming [17], different tokenization schemes, stop words removal, converting text to single letter case) is expressed in the following graphs (Figs. 8, 9, 10 and 11).

We tested three different tokenization schemes, which are mentioned as follows:

- Tokenizing on spaces. This selects tokens of alphanumeric characters (spaces are completely ignored and always treated as a token separator).

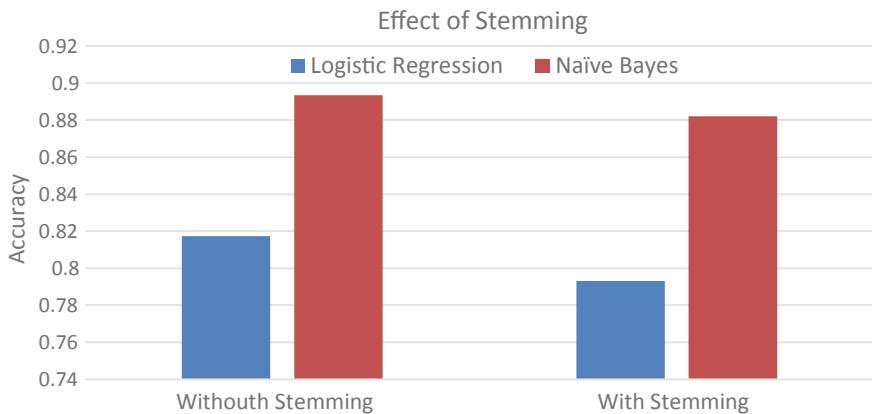


Fig. 8 Bar graph comparing the performance of the model with and without stemming on the basis of score/accuracy for both Naïve Bayes and logistic regression classifier. Dataset used is 20 news group dataset

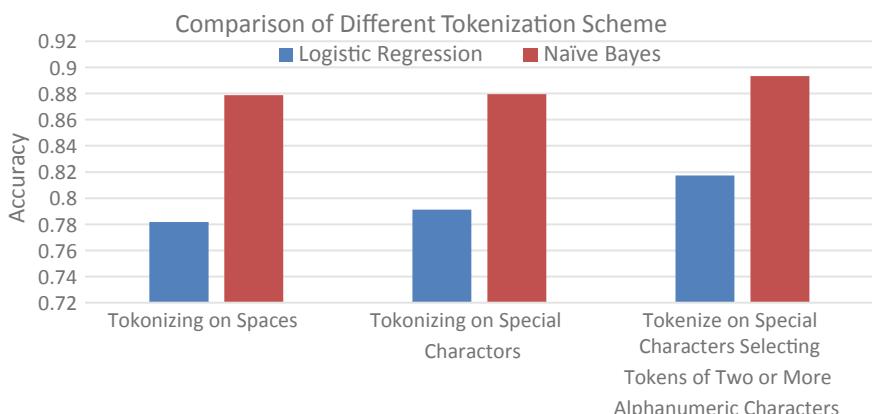


Fig. 9 Bar graph comparing different tokenization schemes on the basis of score/accuracy for both Naïve Bayes and logistic regression classifier. Dataset used is 20 news group dataset

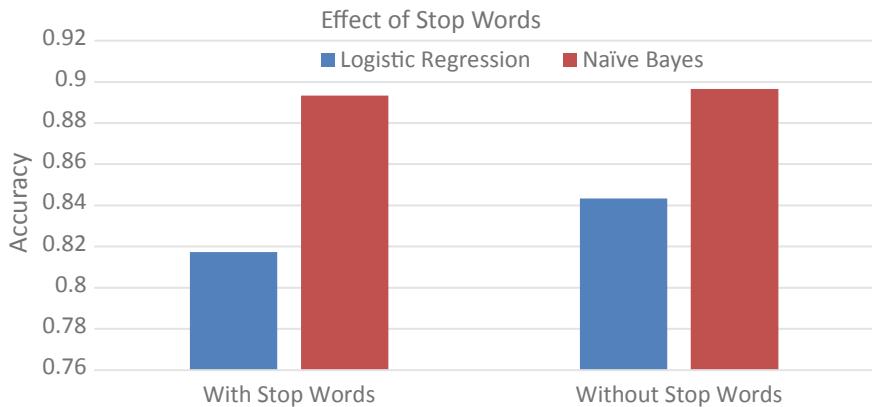


Fig. 10 Bar graph comparing the performance of the model with and without stop words on the basis of score/accuracy for both Naïve Bayes and logistic regression classifier. Dataset used is 20 news group dataset

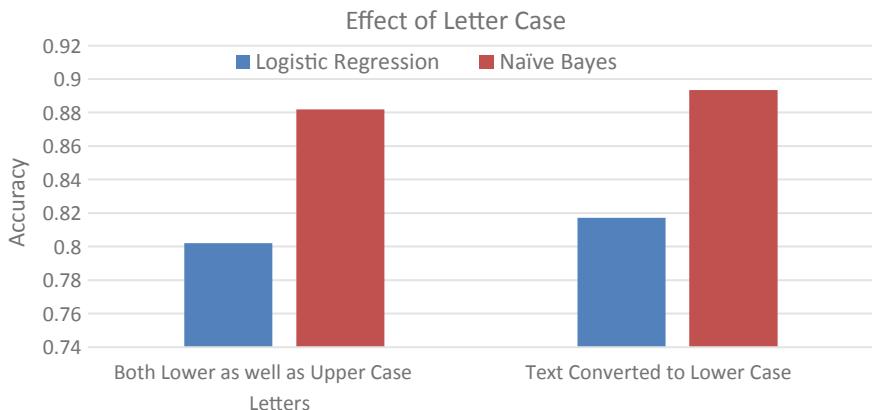


Fig. 11 Bar graph comparing the performance of the classification model with letters of both lower case as well as upper case and text data converted to lower case on the basis of score/accuracy for logistic regression and Naïve Bayes classifier. Dataset used is 20 news group dataset

- b. Tokenizing on special characters. This selects tokens of alphanumeric characters (spaces and special characters are completely ignored and always treated as a token separator).
- c. Tokenize on special characters, but select tokens of two or more alphanumeric characters. (Spaces and punctuations are completely ignored and always treated as a token separator).

Result. Stemming resulted in a slight decrease in performance, but the training time for classifier was reduced. Therefore, logic, which will check the size of the training dataset and make a decision about applying stemming accordingly, should be

incorporated. (As an alternative size of word vocabulary can also be checked (Bag-of-words dictionary) to decide whether to include stemming in our classification system.)

Tokenizing on special characters, but selecting tokens of two or more alphanumeric characters gives better results since tokens/words having less than 2 alphanumeric characters are usually mistakes or stop words or noise. Empirical results show that removing stop words and converting text into one single letter case (either upper case or lower case) improve the performance.

3.4 Effect of Term Interaction and Dimension Reduction

Interaction between every two features is considered (e.g., for every two features x_1 and x_2 , one new feature $x_1 * x_2$ is created), and principal component analysis [18] is used for dimensionality reduction.

If interaction terms are used, then the size of the feature vector will increase considerably; therefore, along with term interaction, dimension reduction techniques need to be used for better computational efficiency. Before giving the output of dimension reduction, to a classifier, data needs to be scaled. From all this, it is clear that term interaction will make our system computationally expensive.

Result. Interaction terms and dimension reduction reduced performance in terms of accuracy for both Naïve Bayes and logistic regression (Fig. 12). Log loss for Naïve Bayes decreased after applying interaction terms and dimension reduction, but for logistic regression, it increased. Thus, interaction term along with dimension

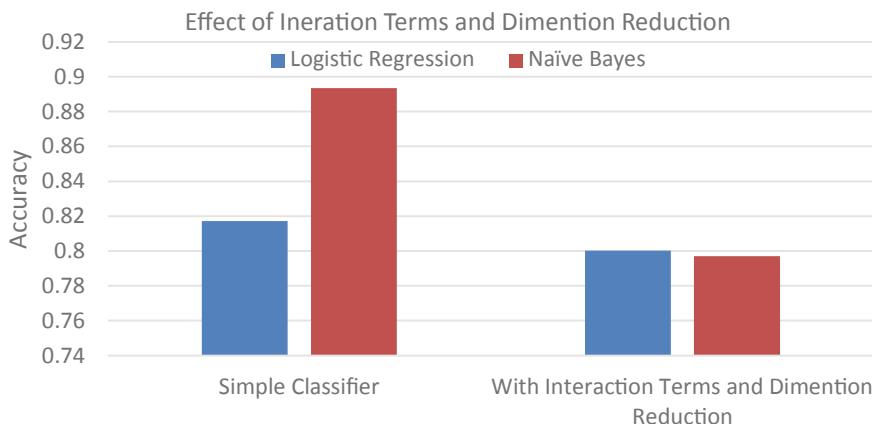


Fig. 12 Bar graph comparing the performance of the model for two different cases (viz. with and without interaction terms and dimension reduction) on the basis of score/accuracy for both Naïve Bayes and logistic regression classifier. Dataset used is 20 news group dataset

Table 1 Performance of optimized classification system. Dataset 1 is 20 news group dataset, and dataset 2 is demo email dataset

Naïve Bayes	Logistic regression
Score/accuracy Dataset-1 0.909898477	Score/accuracy Dataset-1 0.89213198
Dataset-2 1	Dataset-2 1
Log loss Dataset-1 0.276137658	Log loss Dataset-1 0.532203319
Dataset-2 0.007749	Dataset-2 0.021976
Confusion matrix (dataset-1)	Confusion matrix (dataset-1)
$\begin{bmatrix} 360 & 5 & 13 & 11 \\ 53 & 44 & 34 & 13 \\ 5 & 4 & 384 & 4 \\ 14 & 12 & 22 & 346 \end{bmatrix}$	$\begin{bmatrix} 345 & 20 & 8 & 16 \\ 4 & 373 & 8 & 11 \\ 5 & 30 & 353 & 9 \\ 18 & 34 & 7 & 335 \end{bmatrix}$

reduction is not suitable for the text classification problem, while using Naïve Bayes and logistic regression as a classifier.

Table 1 depicts the performance of final optimized classification system.

4 Conclusion

Automated real-time email classification system will be very useful for big companies, corporates, industries and organizations for the analysis of massive email data. Once the proposed email classifier is trained on a labeled dataset, it can be used for achieving real-time classification, and also it can be integrated with any email client. The proposed method is developed in the form of a stand-alone application that can scale to the need of any organization. Use of Naïve Bayes classifier, TF-IDF measure to represent features, construction of feature vector using unigrams of tokens along with text preprocessing will improve the overall performance of the system. The research can be extended further to analyze the efficacy of deep learning techniques for email classification problem due to its massive nature.

References

1. Androutsopoulos I, Koutsias J, Cbandrinos K, Spyropoulos C (2000) An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages.

- In: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR, ACM, New York, USA, pp 160–167
- 2. Yu B, Zhu D (2009) Combining neural networks and semantic feature space for email classification. *Knowl-Based Syst* 22(5): 376–381
 - 3. Rio U, Erlin R (2013) Text message categorization of collaborative learning skills in online discussion using support vector machine. In: International conference on computer, control, informatics and its applications (IC3INA), IEEE, Jakarta, Indonesia
 - 4. Rajalingam M, Raman V, Sumari P (2016) Implementation of vocabulary-based classification for spam filtering. In: 2016 international conference on computing technologies and intelligent data engineering (ICCTIDE'16), IEEE, Kovilpatti, India
 - 5. Rennie JD, Shih L, Teevan J, Karger DR (2003) Tackling the poor assumptions of naive Bayes text classifiers. In: Proceedings of the twentieth international conference on machine learning, vol. 3, pp 616–623. Washington DC
 - 6. Soucy P, Mineau G (2001) A simple k-NN program for text categorization. In first IEEE international conference on data mining (ICDM_01), vol. 28, pp 647–648. California, USA
 - 7. Zhang W, Yoshida T, Tang X (2008) TFIDF, LSI and multi-word in information retrieval and text categorization. In: IEEE international conference on systems, man and cybernetics, Singapore
 - 8. Kibriya A, Frank E, Pfahringer B, Holmes G (2004) Multinomial Naive Bayes for text categorization revisited. In AI 2004: advances in artificial intelligence, pp 488–499. Springer, Berlin, Heidelberg
 - 9. Alsmadi I, Alhami I (2015) Clustering and classification of email contents. *J King Saud Univ—Comput Inf Sci* 27(1):46–57
 - 10. UC Irvine Machine Learning Repository, Twenty Newsgroups Data Set. <http://archive.ics.uci.edu/ml/datasets/twenty+newsgroups>. Last accessed 2019/04/14
 - 11. Bishop C (2006) Pattern recognition and machine learning, Chapter 4.3.4, 1st edn. Springer, New York
 - 12. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
 - 13. Guyon I, Boser B, Vapnik V (1993) Automatic capacity tuning of very large VC-dimension classifiers. In: 7th NIPS conference, Denver, Colorado, USA
 - 14. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth International Group, Belmont, CA
 - 15. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
 - 16. Weinberger K, Dasgupta A, Langford J, Smola A, Attenberg J (2009) Feature hashing for large scale multitask learning. In: ICML proceedings, Changchun, Jilin, China
 - 17. Porter MF (1997) An algorithm for suffix stripping. In: Morgan Kaufmann, readings in information retrieval, Morgan Kaufmann Publishers Inc, pp 313–316
 - 18. Jolliffe IT (2002) Principal component analysis, 2nd edn. Springer, New York

Chapter 37

Smart Detection of Parking Rates and Determining the Occupancy



Deepali Javale, Aatish Pahade, Rushikesh Singal, Akshay Potdar and Mohit Patil

1 Introduction

In this fast-changing globalized world with large number of vehicles commuting, it has become a severe problem for day-to-day travelers to park their vehicles given the variance in parking rates at different parking slots. In order to combat this problem, we have developed a smart parking system which will enable the user not only to determine the current occupancy of any parking slot but will also be able to see the current parking rates of those slots. These parking rates of various slots will be calculated in real time based on various dependent and non-dependent factors. Through this, not only will the user get benefited but also the parking areas will be able to charge their parking slots based on the real-time scenario in the market. We have proposed a smart parking for paid parking systems in the city, wherein the user will get benefits of seeing the current occupancy along with future occupancy of the parking slots. With the implementation of this system, the economical and time costs associated with the traffic will be considerably reduced, and the user will be benefited overhand in deciding where to park his vehicle based on his travel location and current traffic scenario.

D. Javale · A. Pahade (✉) · R. Singal · A. Potdar · M. Patil
MIT College of Engineering, Pune, India
e-mail: aatish97.ap@gmail.com

D. Javale
e-mail: deepali.javale@mitcoe.edu.in

R. Singal
e-mail: rushikeshsingal1998@gmail.com

A. Potdar
e-mail: akshaypotdar179@gmail.com

M. Patil
e-mail: mohitrpatil@outlook.com

The second section of this paper describes the various dependent and independent factors. These factors are then described along with their parking factor rates.

The third section of this paper describes the algorithm for rate calculation based on the proposed algorithm taking into consideration the different factors and the factor values.

The fourth section of this paper describes the prediction of future occupancy of the parking slots. This takes into account the historic data of the transactions of all the incoming and outgoing vehicles and the past occupancies of the parking slot.

The fifth section of this paper describes the actual working of the model. The model uses Raspberry Pi along with infrared/pressure sensors to detect the motion of vehicles and store the data at the back-end system. Then, the optimal parking rate for the individual parking slot is calculated and displayed to the user using a mobile application.

2 Parking Rate Factors

Having a constant parking rate throughout might not be as beneficial as having rates that change dynamically according to various factors that influence the real-time parking. As discussed earlier, the paid parking services would benefit by having a dynamic optimal rate that takes into account various factors. This therefore takes into account various dependent and independent factors of any parking slot. These factors ensure that during low occupancy, the parking rate is reduced, and during high occupancy of the parking slot, the rates increase, thereby making it a profitable situation for the parking services.

The low rates during less occupancy help the users as they have to pay less and also ensure that the parking slots are occupied more easily due to low rates in low occupancy situations.

The parking services thus have a predefined threshold parking rate such that the parking rate does not go below a certain threshold value. The various factors that any parking service takes into account are the kind of day, current occupancy and the presence of nearby parking slots.

Each factor is then subdivided and given a certain factor value based on its relevance (Tables 1, 2).

1. Factor F_1 is ‘day’ which is subdivided into three sub-factors: normal day (factor value = $1/3$), normal weekend (factor value = $2/3$), special events day (factor value = 1).
2. Factor F_2 is ‘occupancy’ which is subdivided into five sub-factors: 0–20% occupancy (factor value = $1/5$), 20–40% occupancy (factor value = $2/5$), 40–60% occupancy (factor value = $3/5$), 60–80% occupancy (factor value = $4/5$), 80–100% occupancy (factor value = 1).

Table 1 Factor rate values

Factor number	Factor	Attributes	Factor value
<i>F1</i>	Day	(1) Normal day (<i>F11</i>)	<i>F11</i> = 1/3
		(2) Normal weekend (<i>F12</i>)	<i>F12</i> = 2/3
		(3) Special events	<i>F13</i> = 1
		Day (<i>F13</i>)	
<i>F2</i>	Occupancy	(1) 0–20% (<i>F21</i>)	<i>F21</i> = 1/5
		(2) 20–40% (<i>F22</i>)	<i>F22</i> = 2/5
		(3) 40–60% (<i>F23</i>)	<i>F23</i> = 3/5
		(4) 60–80% (<i>F24</i>)	<i>F24</i> = 4/5
		(5) 80–100% (<i>F25</i>)	<i>F25</i> = 1
<i>F3</i>	Dependency on nearby parking slot	Dependency matrix	Dependency matrix value

Table 2 Dependency matrix factor values

	<i>P1</i>	<i>P2</i>	<i>P3</i>
<i>P1</i>	0/1	0/1	0/1
<i>P2</i>	0/1	0/1	0/1
<i>P3</i>	0/1	0/1	0/1

Where

P1/P2/P3: Parking slots

Factor value:

0: If parking depends on any nearby parking slot

1: If parking is independent, i.e., does not depend on any nearby parking slot

- Factor *F3* is ‘Dependency on nearby parking slots,’ wherein if a parking slot does not depend on any parking area nearby, then its factor value becomes 1, and whereas if a parking slot depends on any nearby parking slot, then its factor value becomes 0.

3 Parking Rate Determination

The proposed algorithm takes into account the aforementioned factors and calculates the optimal parking rate for the parking services.

$$P_{ij} = (F1i, F2i, F3i, \alpha_j, \beta_j, \delta_j)$$

$$P_{ij} = F1i * \alpha_j + F2i * \beta_j + F3i * \delta_j$$

$$Z_i = P_{ij} - \text{Min}/\text{Max} - \text{Min}$$

where Z_i ranges from [0,1].

$$\text{Final Parking Rate} = T_j * Z_i$$

where

T_j = Threshold parking rate for Parking P_j

Z_i = Normalized value

This proposed formula therefore implies a mathematical function that takes into account various factors to calculate the parking rate. When all the factors are maximum ($F_1 = 1, F_2 = 1, F_3 = 1$), then the parking slot will have its maximum parking rate which will be equal to the threshold parking rate. When all the factors are minimum ($F_1 = 1/3, F_2 = 1/5, F_3 = 0$), then the parking slot will have its minimum parking rate. Each parking slot has its predefined value of $\alpha_j, \beta_j, \delta_j$ for each factor F_1, F_2, F_3 , respectively.

4 Future Occupancy Prediction

The user will benefit if he/she gets to see what the occupancies of the parking slot will be throughout the day. Therefore, it is important to predict future occupancy so that the user could make a comparison between various parkings to see where he/she can get a parking easily. Whenever the parking sensors at the parking slots encounter an incoming or outgoing vehicle, the corresponding entry is made into the database of that particular parking slot. Also, these entries are stored in an excel file (CSV file). This CSV file is then used for future occupancy prediction. Various training models can be applied on this CSV file for prediction. Linear Regression, Regression tree, support vector regression, time series prediction and neural networks are some of the training methods that can be incorporated to determine and predict the future occupancy. The CSV file stored contains the following data entries (Fig. 1):

1. Parking slot ID
2. Day
3. Date
4. Time
5. Occupancy.

5 Implementation

The main objective of this paper is to determine an optimal parking rate and provide the user the future occupancies of the parking slots to facilitate good decision-making. The proposed model takes into use the various components of Internet of Things

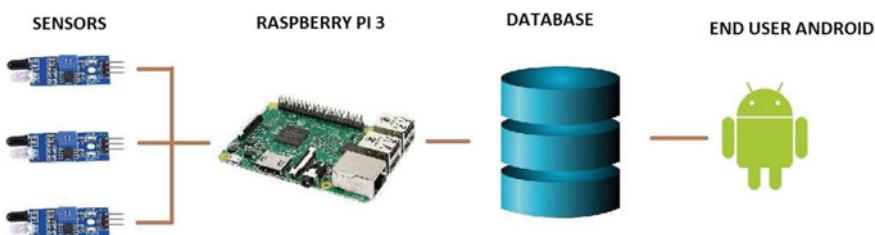
A	B	C	D	E	F
1	ID	day	date	time	occ
2	1	Tuesday	15-Jan-19	0	10
3	2	Tuesday	15-Jan-19	0	9
4	1	Tuesday	15-Jan-19	0	12
5	1	Tuesday	15-Jan-19	1	23
6	3	Tuesday	15-Jan-19	1	30
7	3	Tuesday	15-Jan-19	2	24
8	2	Tuesday	15-Jan-19	3	8
9	2	Tuesday	15-Jan-19	3	20
10	1	Tuesday	15-Jan-19	4	5
11	3	Tuesday	15-Jan-19	5	20
12	2	Tuesday	15-Jan-19	6	15
13	1	Tuesday	15-Jan-19	7	20
14	1	Tuesday	15-Jan-19	8	35
15	3	Tuesday	15-Jan-19	8	40
16	3	Tuesday	15-Jan-19	8	30
17	2	Tuesday	15-Jan-19	8	45
18	2	Tuesday	15-Jan-19	9	50
19	2	Tuesday	15-Jan-19	9	44
20	1	Tuesday	15-Jan-19	9	48
21	3	Tuesday	15-Jan-19	10	55
22	1	Tuesday	15-Jan-19	10	60
23	1	Tuesday	15-Jan-19	10	65
24	2	Tuesday	15-Jan-19	10	50

Fig. 1 Snapshot of CSV file

(IoT). At the entry point of each parking slot, there are infrared sensors deployed. Once a vehicle enters or exits the parking slot, the corresponding entry is made into the database along with the excel sheet.

These values are then extracted and used for determining the rate of the parking slots. For each parking, its maximum occupancy, current occupancy and parking rate are stored in the database.

The various components as seen from Fig. 2 are:

**Fig. 2** Architectural model

1. Infrared/Pressure Sensors and Raspberry Pi
2. Back-end system
3. End-user application (Android application).

5.1 Infrared/Pressure Sensors and Raspberry Pi

An effective and low-cost solution to detect the presence of any incoming or outgoing vehicle is the use of sensors at the gateways of the parking slots. These infrared sensors sense the surroundings by detecting the infrared radiation and the motion of the vehicles. These sensors can work efficiently in a small prototype model, but for an actual implementation, these sensors may not be a feasible option.

In this proposed prototype model, the sensors sense the vehicle and give the count of the vehicles through the use of Raspberry Pi, and this data is then stored at the back-end system. The current status of any parking slot is depicted by LED sensors. Once all the slots are filled, the LED turns red and does not allow any incoming vehicle to enter the parking slot.

A further extension to the use of infrared sensors is the pressure sensors. These pressure sensors can be used to determine each parking spot within a given parking slot. This gives the exact details of the availability of each parking spot within the parking slots and can be used to give directions to the user to guide him/her to the vacant spot in the parking area. Figure 3 shows the Raspberry Pi hardware that is used in this prototype. The Raspberry Pi version that is used is Raspberry Pi 3.0. The Raspberry Pi contains various components and enables us to send the data to the back-end system. Figure 4 shows the sensors and LEDs used in the model.

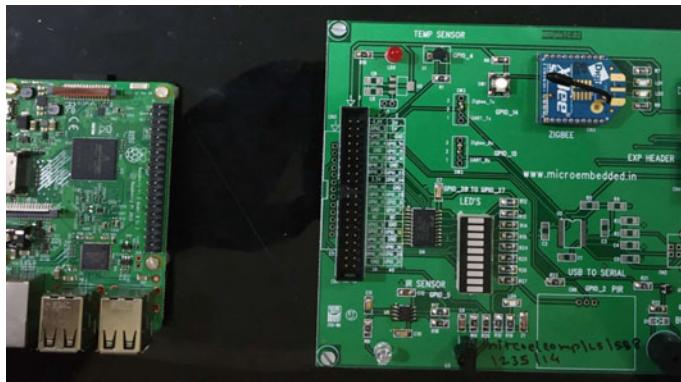


Fig. 3 Raspberry Pi

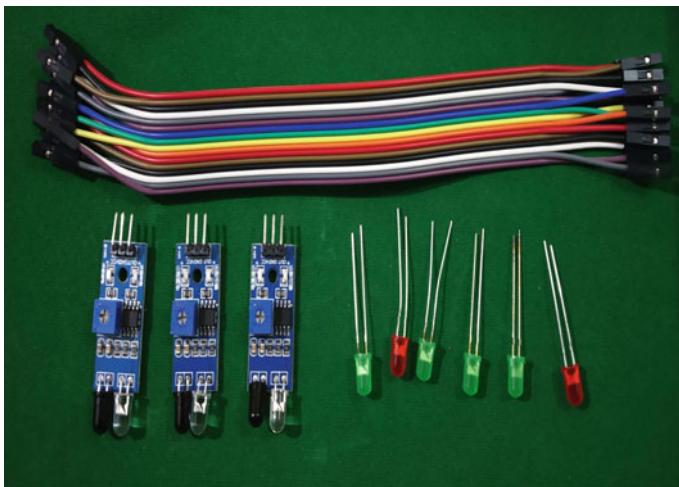


Fig. 4 IR sensor and LED

5.2 Back-End System

For keeping track of the vehicles, the sensors detect the motion of the vehicles, and with the help of Raspberry Pi, these data is then stored in an online format at the back-end system. The proposed model uses xampp as the back-end database system. The use of xampp makes it convenient to extract and store data in the back-end system. It also facilitates easy use of python language as a programming tool for rate prediction calculation using this back-end system.

In the database, the following entries are stored:

1. Parking ID of parking slot
2. Current occupancy
3. Maximum occupancy
4. Parking rate
5. Parking name.

5.3 End-User Application

In the era of smart mobiles, it is convenient to build a system where the user gets all the features on his/her smartphone. The various parking slots are displayed to the user via android mobile application. The use of android app enables the user to check the rates and occupancies of the parking slots on his/her phone while traveling and makes the system more robust. The mobile application displays various parking

spaces along with their location. In each parking space, its corresponding rate, current occupancy and future occupancy are displayed.

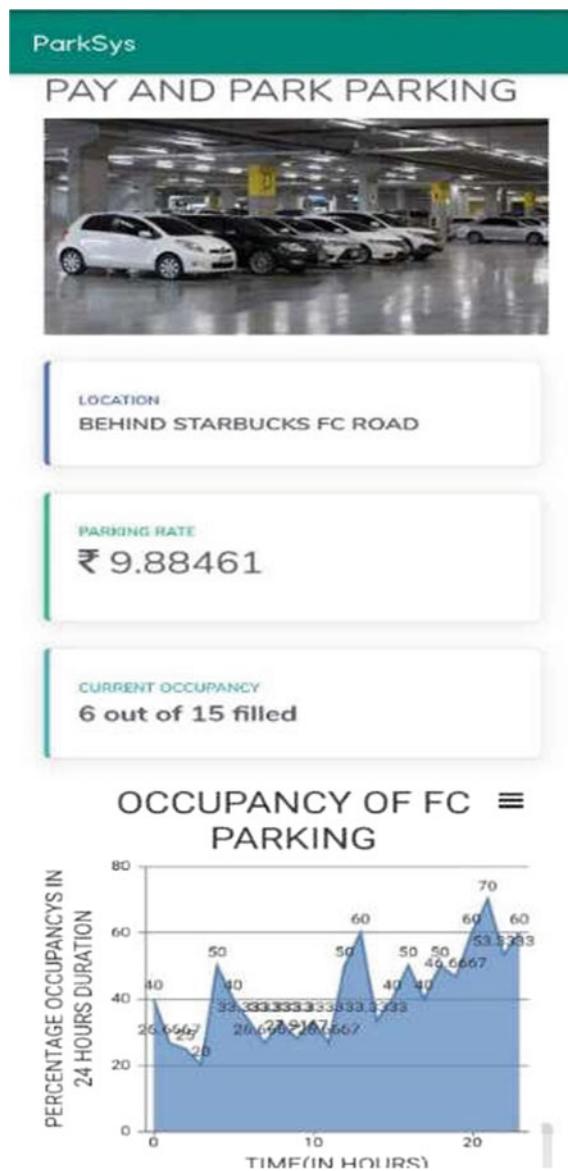
6 Results

The proposed prototype was successfully implemented and tested using the Raspberry Pi connected to infrared sensors. These sensors detected any motion of the prototype vehicle, and this motion detection was successfully recorded and stored in the back-end system database in the xampp server. A mobile application was developed to display the results of the proposed model, and any mobile having an Internet connection was able to use the application with the help of web server. The screenshots of the mobile application are shown in Fig. 5.

7 Conclusion

We have designed a prototype model to determine the optimal parking rate and predict the future occupancy of the paid parking services on a real-time basis. The model uses Raspberry Pi along with the sensors to detect any motion and record this transaction in the database. The end user is able to view various parking slots in the nearby area along with their current parking rates and future occupancy with the help of mobile application. Both the parking services and the end users are benefited as the rates are not static for any particular day but vary according to the various factors, thus making the system more dynamic and efficient. The use of the proposed model makes the system cost-efficient and easy storage and retrieval of data. This model can further be expanded to cater the needs of an entire city. The sensors can be effectively used to optimize the parking areas and detect vehicles with ease. An extended model would thus be able to give the user an entire perspective of the available paid parking systems in the city.

Fig. 5 Snapshot of mobile application-parking slot



References

1. Sadhukhan P (2017) An IoT-based e-parking system for smart cities
2. Dong S, Chen M, Li H (2018) Parking rank: a novel method of parking lots sorting and recommendation based on public information
3. Grodi R, Ravat DB, Rios-Gutoerrez F (2016) Smart parking: parking occupancy monitoring and visualization system for smart cities

4. Ioannou TRPA (2015) On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Trans Intell Trans Syst* 16(5):2913
5. Basavaraju (2015 Dec) Automatic smart parking system using Internet of Things (IOT). *Int J Sci Res Publ* 5(Issue 12). ISSN 2250-3153
6. Kanteti D, Srikanth DVS, Ramesh TK (Apr 2017) Smart parking system for commercial stretch in cities. In: International conference on communication and signal processing
7. Rajabioun T, Ioannou PA (2015) On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Trans Intell Transp Syst* 16(5):2913–2924
8. Yuan C et al (2016) A smart parking system using WiFi and wireless sensor network. In: IEEE international conference on consumer electronics-Taiwan IEEE, 1–2
9. Chen C-L, Chiu W-C (2017) A recommendation model of smart parking. In: 13th International conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD 2017)

Chapter 38

Deep Learning-Based Approach to Classify Praises or Complaints from Customer Reviews



Sujata Khedkar and Subhash Shinde

1 Introduction

Nowadays, it is essential for business organizations to provide exceptional services to customers. More and more data in both structured and unstructured formats is generated at higher rate due to social media and review sites, which can be transformed into information and can be used for business intelligence. It is important for hotels to collect data from various channels, analyze it and measure customer satisfaction, understand customer behavior, and recognize customer needs through reviews or social media channels. Many business intelligence applications across various domains use opinion mining as the key enabling technology for their business intelligence [1–3] from big review data. The system is built by aggregating the opinions from various channels like blogs, comments, reviews, or tweets. The major challenge is to find the extreme opinions in terms of praises and complaints to understand customer true opinions w.r.t. service rather than just analyzing positive or negative sentiments [4].

Sentiment analysis analyzes people's opinions and sentiments from unstructured text. Sentiment analysis works at three different levels of granularity, namely document level, sentence level, and aspect level [5, 6]. Most early research on sentiment analysis was focused on document level and is used in the domains like hotels, restaurants, e-commerce products, politics, etc. These online reviews have a significant impact on customer's purchase decisions [7].

S. Khedkar (✉)

Computer Engineering Department, Thadomal Shahani College of Engineering, Bandra (W), Mumbai, India

e-mail: sujata.khedkar@ves.ac.in

S. Shinde

Computer Engineering Department, Lokmanya Tilak College of Engineering, Koparkhairane, Navi Mumbai, India

e-mail: skshinde@rediffmail.com

The sentiment classifier classifies the text into three predefined categories as positive, negative, or neutral based on the set of features using a classification method [8, 2]. However, it is possible to make other types of classifications by considering informativeness of reviews and to detect and classify reviews as praise or complaint. Praises are a subset of positive reviews and complaints are a subset of negative reviews. These praises and complaints can be called as extreme opinions.

Praises express the best views and complaints express the worst views or judgments w.r.t. service. The customer always wants to know about the best and the worst aspects of product or service while making online purchase decisions. These very positive praise reviews and very negative complaint reviews have a very strong impact on product sales [9].

With the massive increase in the number of customer reviews, identifying praise/complaint sentences is important for analyzing customers' likes/dislikes w.r.t. product/service. The manual way of finding these extreme opinions is not scalable. In this work, we focus on the problem of classifying a review sentence as praise/complaint. The earlier work uses manual feature extraction. However, recently deep learning methods have shown good results over traditional methods.

In this paper, we compared the performance of multiple classifiers such as random forest, SVC, K-neighbors, and MLP. Main contributions of our paper are as follows: (1) We investigate the application of deep learning methods for the task of extreme opinion classification. (2) We have proposed linguistic hybrid features based on characteristics of praise and complaint sentences. (3) We have evaluated the supervised classification algorithms with base features and proposed hybrid features for extreme opinion classification task. (4) We have compared the performance of machine learning-based and deep learning-based approach for identifying praise/complaint sentences from review data. (5) Our method gives better results than state-of-the-art methods.

Customer reviews contain numerical rating. The challenges are (i) we cannot apply the same rating to all sentences. Positively rated reviews also contain negative sentences. The existing approaches do not consider this while analyzing sentiments. (ii) Some sentences are more informative than other sentences. Existing approaches treat all sentences equally.

In this paper, we used linguistic features of praise and complaint sentences and Affin dictionary to calculate sentiment and the overall score of each sentence. Only praise and complaint sentences whose rank is above the threshold are given as input to the analytical model. We compared supervised machine learning algorithms and deep neural network-based models such as dense NN, CNN + dense NN, and multichannel CNN for praise/complaint classification.

The rest of paper is organized as follows. In the following Sect. 2, we discuss the related work. Then, Sect. 3 describes the proposed methodology. Section 4 describes the experiments, evaluations, and results. Conclusion and future work are described in Sect. 5.

2 Related Work

In related work, we found three main approaches to find the sentiment at the sentence level. First, machine learning techniques based on training corpus annotated with polarity information and, second strategies based on lexicons, and third is deep learning-based models.

In supervised learning, the classification performance depends on the identification of quality features and classification methods. Various types of features have been used in classification tasks such as lexical features, stylistic features, content features, sentiment features, and semantic features [10]. The influence of these features for classification task has been evaluated and analyzed by some sentiment analysis studies [9, 11, 12]. But some reviews are more informative (praise/complaints) as compared to plain positive/negative text. These informative reviews are very important from the business intelligence point of view for decision making. With the abundant amount of opinionated text, methods for automatic extraction of praises or complaints from given text are not there in existence

Customers often read only a few reviews and cannot get useful information from it. There are many noninformative sentences in the reviews. Most of the research previously was based on quantitative ratings given by the customer on online websites. The extreme opinions like praises and complaints are very challenging to identify from review corpus. Only 5–10% reviews contain extreme opinions [13].

Ganesan [8] studied the properties of praise and complaint sentences where praise is a subset of positive only reviews and complaint is a subset of negative only reviews. The praise sentence contains more adjectives, intensifiers, and nouns; the length of praise is longer than average length of sentence. On the contrary, complaint sentence has fewer adjectives and nouns, more past tense and conjunctions. The existing sentiment analysis algorithms can be improved by only considering the analysis of praise and complaint sentences from big customer reviews data (Table 1).

Mahmoud Othman, Hersham Hassan [14] has developed an opinion summarization system based on linguistic properties of review sentences and further classified as direct, comparative, or superlative opinion. The sentences are POS tagged and the sentences having POS tags as JJR and RBR are considered as comparative opinionated sentences; the sentences which contain JJS and RBS tags are considered as superlative opinionated sentence; and the sentences having JJ are considered as the direct opinioned statement. They have considered the overall review and extracted

Table 1 Examples of different types of sentences

Sentence	Type
I don't like this hotel	Negative only
Unhappy with this hotel, price too high, rooms not clean	Complaint
I like this hotel because it's very cheap, nice food, good service, and big rooms	Praise
I like this hotel	Positive only

only opinionated sentences without further analysis of positivity or negativity within opinionated sentences.

Saumya [1] had developed review ranking system based on helpfulness score. The system classifies low-quality and high-quality reviews based on hybrid features like nouns, adjectives, verb, difficult words, wrong words, entropy, review ranking, etc., and random forest classifier. Low-quality reviews are ignored and only high-quality reviews are displayed.

Almatarneh and Gamallo [15, 16] have used linguistic features to identify extreme opinions; they have used a bag of words, word embeddings, polarity lexicon and set of textual features to identify complaint reviews and from a set of reviews. De Souza [17, 18] have used convolutional neural network approach to sentiment analysis applied to hotel reviews and compared it with machine learning models.

Kim [19] used convolution neural network for sentence classification and showed that deep neural networks show comparatively better results than state-of-the-art techniques. Martín [20] has developed and compared various deep learning-based models based on CNN and LSTM for hotel review corpus. The most accurate and robust estimators are those based on LSTM recurrent neural networks. Customer reviews contain numerical rating. The challenges are (i) we cannot apply the same rating to all sentences. Positively rated reviews also contain negative sentences. The existing approaches do not consider this while analyzing sentiments. (ii) Some sentences like extreme opinions are more informative than other sentences. Existing approaches treat all sentences equally. So, this paper proposes a novel approach to deal with big review data by applying linguistic feature-based filtering technique and hybrid feature-based praise or complaint classifier to get more precise opinions from customer reviews.

3 Proposed Approach

The proposed approach involves the following steps as shown in Fig. 1.

3.1 Data Preprocessing

Customer reviews are collected from review sites. The reviews are converted into sentences. Non-textual sentences and sentences that are irrelevant for the analysis are identified and eliminated. Data preprocessing techniques such as stop word removal, and stemming are used for preprocessing. The review might contain positive as well as negative sentences so the same numerical rating cannot be applied to all sentences. The Affin Dictionary is used to compute the sentiment score of each sentence. The sentences which are neutral sentences are eliminated from further analysis.

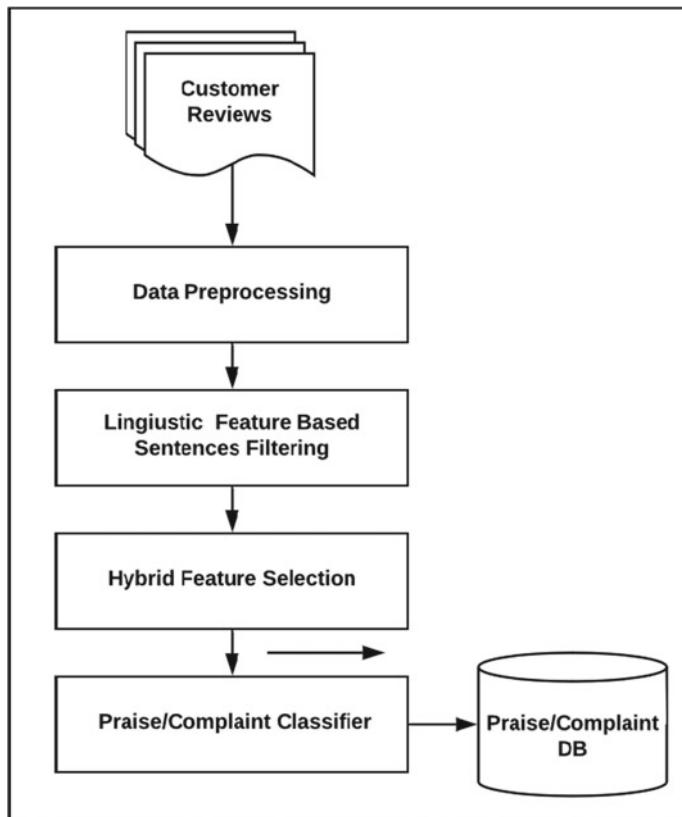


Fig. 1 A proposed approach for praise or complaint classification

3.2 *Linguistic Feature-Based Sentence Filtering*

In this step, the extracted sentences are linguistically evaluated based on linguistic features present such as number of nouns, adjectives, and intensifiers. The final praise and complaint score of the sentence is calculated using Formulas (2) and (4).

$$\begin{aligned} \text{P_score} = & 0.1 * \text{len(sentence)} + 0.2 * \text{len(nouns)} + 0.3 * \text{len(adj)} \\ & + 0.4 * \text{len(intensifier)} \end{aligned} \quad (1)$$

$$\text{Praise_score} = \text{P_score(sentence)} * \text{Sentiment_score(sentence)}/10 \quad (2)$$

$$\begin{aligned} \text{C_score} = & -0.1 * \text{len(sentence)} + 0.1 * \text{len(nouns)} + 0.2 * \text{len(conjunction)} \\ & + 0.3 * \text{len(Adverb)} + 0.3 * \text{len(Verb)} \end{aligned} \quad (3)$$

Table 2 Description of all linguistic features

Features	Descriptions
FS1	Unigram TF-IDF (1-gram)
FS2	Bigram TF-IDF (2-gram)
FS3	Trigram TF-IDF (3-gram)
FS4	FS1 (Unigram TF-IDF) + FS4 (Hybrid features-(Meta + Synthetic + content + Semantic))
FS5	FS2 (Bigram TF-IDF)+ FS4 (Hybrid features)
FS6	FS3 (Trigram TF-IDF) + FS4 (Hybrid Features)

$$\text{Complaint_score} = \text{C_score(sentence)} * \text{Sentiment_score(sentence)}/10 \quad (4)$$

All neutral sentences having zero sentence score are eliminated from the corpus.

3.3 Hybrid Feature Selection

This module selects a subset of relevant features. Hybrid feature selection method uses various features like

- Meta features—length of sentence, number of nonstop words, and average word length.
- Synthetic features—number of nouns, number of adjectives, number of verbs, number of intensifiers, number of adverbs, etc.
- Content features—words which are verbs, nouns, adjectives, adverbs, verb phrases, etc.
- Semantic features—positive or negative rating and praise/complaint score of the sentence from the sentence filtering module.

Table 2 shows features that are considered for evaluation.

Various experiments with a combination of the different feature set are performed on the selected dataset.

3.4 Praise/Complaint Classifier Module

Many researchers have focused on the use of traditional classifiers such as Naïve Bayes, SVM, maximum entropy, etc., to solve classification problems. We have studied the usefulness of various classification algorithms such as SVC, random forest, MLP, and K-neighbors algorithms for this problem. We have also evaluated the performance of dense, CNN + dense, and multichannel CNN networks on the hotel reviews dataset.

4 Experiments

4.1 Data Preprocessing

The dataset is preprocessed using the following steps: (1) Neutral reviews (3-star rating) are deleted since they do not convey any views; (2) review sentences having short lengths are eliminated; (3) Affin dictionary is used to calculate the sentiment score of each sentence; (4) neutral sentences having Affin sentiment score zero is eliminated; and (5) linguistic properties of each sentence is computed and praise score/complaint score of each sentence is computed based on Eqs. (1–4). Table 3 shows the number of reviews in each class. The hotel reviews dataset (<https://www.kaggle.com/harmanpreet93/hotelreviews/activity>) is used in this case study (Fig. 2).

Exploratory data analysis shows that there are 35.1% praises and 23.5% complaint sentences and 26.9% plain positive and 14.5% plain negative non-informative sentences. Plain positive and plain negative review sentences are eliminated from further analysis.

Table 3 Hotel dataset with a total number of review sentences in each class

Datasets	# of review sentences	Negative only	Complaints	Positive only	Praises sentences
Hotel	1,00,012	14,491	23,545	26,896	35,080

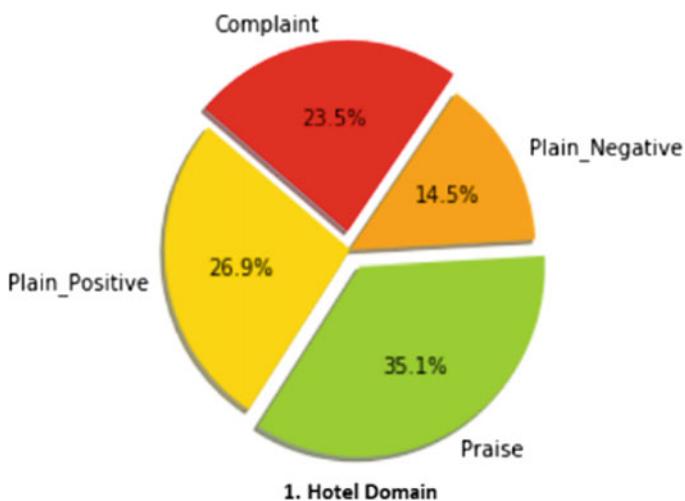


Fig. 2 Exploratory data analysis of customer reviews for hotel reviews

4.2 Training and Testing

We have used four supervised classification algorithms random forest, K-neighbors classifier, SVC, and multilayer perceptron. The dataset was randomly partitioned into training (80%) and testing (20%). We have used stratified sampling and used fivefold cross-validation. We have evaluated the performance of dense, CNN + dense, and multichannel CNN networks on the above dataset. The output is evaluated using four parameters: precision, recall, F1-score, and accuracy.

For the development of the neural network model, the Keras framework was used. Keras is an open-source neural network library written in Python; it provides a high-level API for fast experimentation. Keras is capable of running on top of TensorFlow, CNTK, or Theano as backend.

The best parameters for our models were selected and further fine-tuned through manual testing. The neural network is trained based on only important words from sentences like nouns, verbs, intensifiers, connectives, etc. The dense neural network is built with an input vector of length 100, no of neurons in the dense layer is 64, the dropout rate is 0.5, and activation function used is Relu as shown in Fig. 3.

The CNN dense neural network model, multichannel CNN with kernel size 2, 4, dropout rate = 0.2, activation = Relu, Optimizer = adam is build (Fig. 4) and evaluated.

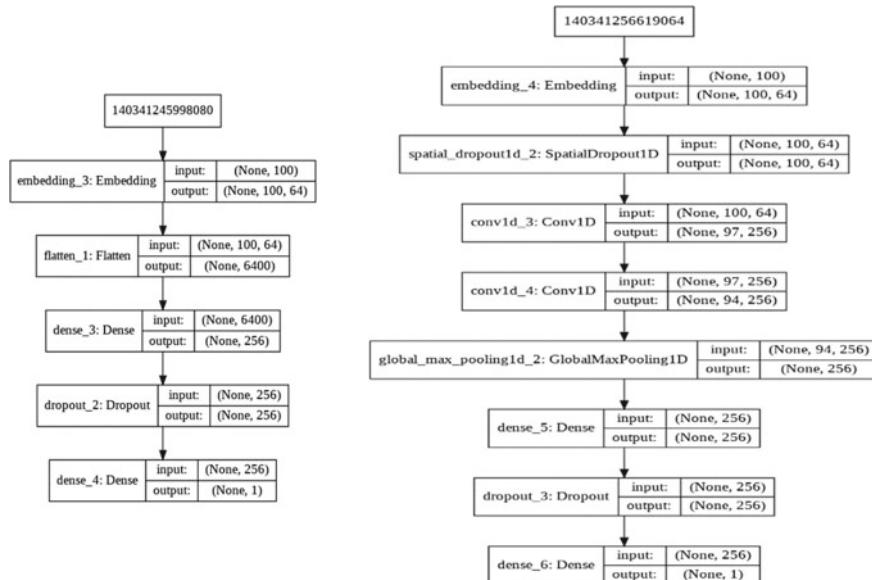


Fig. 3 a Dense neural network, b CNN dense NN architecture model

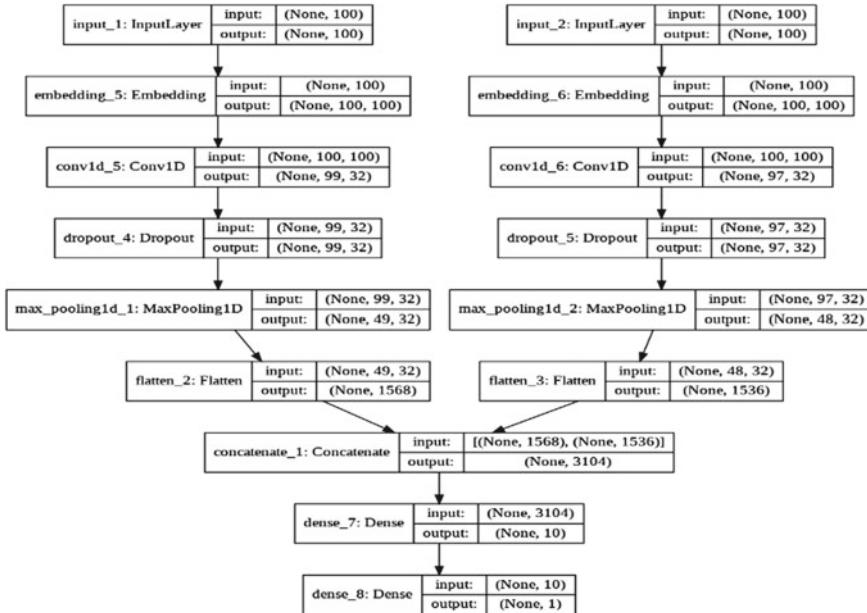


Fig. 4 Multichannel CNN with kernel size 2, 4

4.3 Results

Table 4 shows the classification results obtained by random forest, K-neighbors, SVC, and multilayer perceptron classifiers for all datasets. These classifiers are evaluated based on base features and linguistic features and their combinations. The final scores were computed using precision (P), recall (R), F1-scores, and accuracy. Tables 4 and 5 show classification results obtained using random forest, K-neighbors, SVC, and MLP with base features, linguistic features, and combinations.

For hotel domain reviews, the random forest, MLP algorithms with Unigram TF-IDF, and hybrid linguistic features give F1-Measure of 0.997 and outperform other classifiers with accuracy 99.6% (see Table 4; Fig. 5).

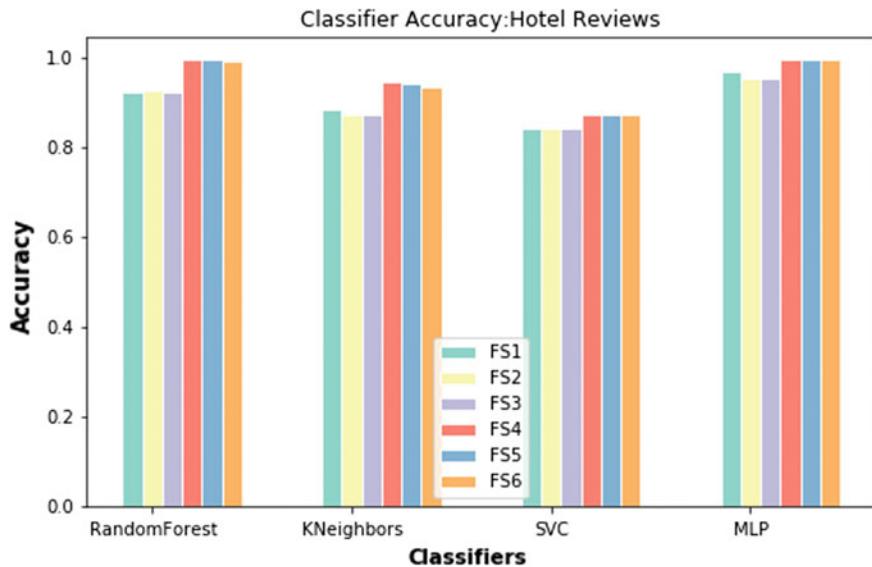
Base features combined with hybrid features have given better results as compared to simple base features. The deep learning models when given with selected features in terms of important words such as nouns, verbs, adjectives, intensifiers, and connectives have shown comparable performance as that of machine learning algorithms with proposed hybrid features. Multichannel CNN outperforms other deep learning models with F1-score of 96.23% as shown in Table 5.

Table 4 Classification results for hotel reviews

Hotel domain reviews	Random forest			SVC			KNeigbor			MLP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
FS1	0.877	0.820	0.844	0.420	0.50	0.456	0.809	0.705	0.740	0.948	0.928	0.983
FS2	0.870	0.839	0.925	0.420	0.50	0.456	0.788	0.677	0.710	0.924	0.898	0.910
FS3	0.864	0.838	0.851	0.420	0.50	0.456	0.793	0.652	0.687	0.923	0.897	0.909
FS4	0.998	0.997	0.997	0.841	1.0	0.913	0.946	0.990	0.967	0.997	0.998	0.997
FS5	0.996	0.999	0.997	0.841	1.0	0.913	0.945	0.989	0.967	0.995	0.998	0.994
FS6	0.998	0.994	0.996	0.841	1.0	0.913	0.945	0.987	0.966	0.995	0.998	0.994

Table 5 Classification results using deep learning models

Hotel reviews	P	R	F1	Train accuracy	Test accuracy
Dense NN	0.953	0.968	0.960	0.999	0.933
CNN + dense NN	0.970	0.949	0.960	0.990	0.933
Multichannel CNN	0.960	0.964	0.9623	0.999	0.936

**Fig. 5** Classifier accuracy for hotel reviews

5 Conclusions

In this paper, to deal with big review data, we have proposed the approach for filtering noninformative review sentences from the review corpus based on linguistic properties of extreme opinions (praise/complaint). We have used properties of extreme opinions such as praise and complaint sentences and proposed hybrid features for praise and complaint classification. We have examined the performance of these features with supervised learning classifiers (random forest, SVC, K-neighbors) to identify extreme opinions like praises or complaints about hotel domains review dataset.

The method proposed in this paper shows that existing sentiment analysis techniques can be improved by only considering informative sentences in the sentiment analysis process. The machine learning algorithms with hybrid features and deep learning models with important words showed comparable results. The random forest classifier outperforms all other classifiers with unigram+ hybrid features. Multichannel CNN outperforms other deep learning models and gives highest F1-score

of 96.23%. In the future, we will be evaluating other classifiers and deep learning models like LSTM, RNN, and GRU for classifying extreme opinions. We will also evaluate the use of pre-trained word embeddings such as a glove, fast, and text and applicability of RNN, LSTM, and GRU deep learning models in identifying extreme opinions from review corpus.

References

1. Moschitti A, Basili R (2004) Complex linguistic features for text classification: a comprehensive study. *advances in information retrieval*, 181–196
2. Liu Y, Jiang C, Zhao H (2018) Using contextual features and multi-view ensemble learning in product defect identification from online discussion forums. *Decis Support Syst* 105:1–12
3. Alaei AR, Becken S, Stantic B (2017) Sentiment analysis in tourism: capitalizing on big data. *J Travel Res.* 004728751774775
4. Hu N, Zhang T, Gao B, Bose I (2019) What do hotel customers complain about? Text analysis using structural topic model. *Tour Manag* 72:417–426
5. Liu B (2012) Sentiment analysis and opinion mining. *Synth Lect Hum Lang Technol* 5:1–167
6. Turney PD (2002) Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics, Philadelphia, PA, USA, 7–12 July 2002. Association for Computational Linguistics: Stroudsburg, PA, USA, pp 417–424
7. Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting on association for computational linguistics, pp 115–124. Association for Computational Linguistics
8. Ganeshan K, Zhou G (2016) Linguistic understanding of complaints and praises in user reviews. In: Proceedings of NAACL-HLT
9. Pang B, Lee L (2008) Opinion mining and sentiment analysis. *Found Trends Inf Retr* 2:1–135
10. Abrahams AS, Fan W, Wang GA, Zhang Z, Jiao J (2015) An integrated text analytic framework for product defect discovery. *Product Oper Manag* 24:975–990
11. Zhao Y, Xu X, Wang M (2018 Mar) Predicting overall customer satisfaction: Big data evidence from hotel online textual reviews. *Int J Hosp Manag*
12. Saumya S, Singh JP, Baabdullah AM, Rana NP, Dwivedi YK (2018) Ranking online consumer reviews. *Electron Commer Res Appl*
13. Kharde VA, Sonawane S (2016 Apr) Sentiment analysis of Twitter data: a survey of techniques. *Int J Comput Appl* 139(11): 5–15
14. Krishnamoorthy S (2015) Linguistic features for review helpfulness prediction. *Expert Syst Appl* 42(7):3751–3759
15. Almatarneh S, Gamallo P (2018) Linguistic features to identify extreme opinions: an empirical study. *Lect Notes Comput Sci*, 215–223
16. Almatarneh S, Gamallo P (2018) A lexicon-based method to search for extreme opinions. *PLoS ONE* 13(5):e0197816
17. De Souza JGR, de Paiva Oliveira A, de Andrade GC, Moreira A (2018) A deep learning approach for sentiment analysis applied to hotel's reviews. *Lect Notes Comput Sci*, 48–56. https://doi.org/10.1007/978-3-319-91947-8_5
18. Chatterjee S, Deng S, Liu J, Shan R, Jiao Wu (2018) Classifying facts and opinions in Twitter messages: a deep learning-based approach. *J Bus Anal* 1(1):29–39
19. Kim Y (2014) Convolutional neural networks for sentence classification. In: EMNLP, pp 1746–1751
20. Martín CA, Torres JM, Aguilar RM, Diaz S (2018) Using deep learning to predict sentiments: case study in tourism. Complexity 2018(Article ID 7408431), 9 pages. <https://doi.org/10.1155/2018/7408431>

Chapter 39

Psychological Behavioural Analysis of Defaulter Students



Rucha Karanje, Sneha Jadhav, Divya Verma and Shilpa Lambor

1 Introduction

It has been observed that a number of defaulter students are significant. Instead of having 3–4 defaulter students, it is observed that we now have around 20–30 defaulter students out of a class of 60 students. Some students are defaulters in multiple subjects. According to this calculation on an average, each class has 4050% of students whose attendance is not above 75% for a specified month. This has been observed in many universities. This is an alarming percentage of students. One to one teaching is very beneficial but if students don't attend lectures then in the long run it's going to impact their grades. There can be many reasons for which a student does not attend lectures.

Maybe they do not find the course interesting, maybe they are facing some health issues, maybe they are going through some family problems or some psychological issues due to some reason, or maybe they are too much engaged in some other extracurricular activities.

The main objective of our project is to understand why a student is a defaulter by student's psychological test and by observing the student's responses to those

All authors have contributed equally.

R. Karanje (✉) · S. Jadhav · D. Verma · S. Lambor

Department of Electronics Engineering, Vishwakarma Institute of Technology, Pune,
Maharashtra, India

e-mail: rucha.karanje15@vit.edu

S. Jadhav

e-mail: sneha.jadhav15@vit.edu

D. Verma

e-mail: divya.verma15@vit.edu

S. Lambor

e-mail: shilpa.lambor@vit.edu

questions. Then, our target is to design such an algorithm using which we can predict whether a random student will become a defaulter or not in the future. This will be done by observing the student's responses to the psychological questions. Through this, we will be able to understand the reasons behind a student's unwillingness to attend college and whether if someone is going through some serious mental, financial and psychological problems so that we can provide them with proper guidance and help which they need.

Using K-NN algorithm, random forest algorithm and decision tree algorithm, the process is carried out so as to ensure maximum efficiency while predicting defaulter students. Many authors have previously used machine learning as a tool for predicting solutions to psychological issues. K-NN classifier has been used before for the analysis of emotion recognition [1]. It has also been used for facial expression recognition [2]. Behavioural modelling has also been done [3]. Mood detection of psychologically disturbed patients is also done [4]. There are many researchers published their work on the decision tree. The publishers in [5] have focused another aspect of prediction. Its prediction is directed towards the scholarship and the financial aid that students receive to keep them motivated.

Some work has also been done on random forest classifier. Using random forest regression, the battery state-of-charge (SOC) estimation method is also proposed in some papers [6]. The bike-sharing systems allow people to rent a bike from one location and return the bike to the location of their choice [7]. Hence, it offers to use a random forest model and a GBM packet to improve the decision tree.

2 Proposed Solution

To increase efficiency, we propose to carry out the same problem using three different algorithms:

1. K-NN classifier
2. Decision tree
3. Random forest algorithm.

2.1 K-NN Classifier

K-NN is the simplest and influential method of classification of an emotional state; similar observations belong to similar classes is the key idea behind K-NN classification. K-nearest neighbour is an instance-based learning classifier. In this model, when a new data point is added, that new data point is compared with k number of nearest neighbours and then the new data point is classified into the class with the maximum number of nearest data points. Now that new point is considered as a part of that class. The advantage of using K-NN is that it has zero cost of learning process,

no assumptions need to be made, and complex concepts can be learned by approximation. K-NN is a memory-based approach due to its instance-based learning. The classifier keeps adapting and changing as we go on adding new data which makes this useful for real-time inputs. K-NN can show good accuracy if more amount of training data is provided.

2.2 *Decision Tree*

A decision tree can be used for classification and regression. It takes the data set, runs the algorithm and creates a model or a classifier. When the model is generated in a tree-type structure, it is called a decision tree classifier. Here, we are using a decision tree to predict. A decision tree is a flowchart-like tree structure where an internal node signifies feature, the branch represents a decision rule, and each leaf node represents the outcome. The tree consists of various nodes. There are two types of nodes: decision node and a leaf node. A decision node can be considered as a test which can have more than one result. The results of the test are leaf nodes. Branches connect the decision node to the leaf node. Branches depict the decision that has been taken that lead to the leaf node. The test is performed on the feature or attribute value. The data set is first fed to a root node. The very first node decides on how to segregate the data set based on the decision (yes or no for example) that it takes. Entropy, as we all know, is the extent of randomness in any system.

2.3 *Random Forest Algorithm*

In machine learning, the random forest algorithm is also known as the random forest classifier. It is a well-known and mostly used algorithm. There are number of trees in a single forest. Trees in the forest vary from each other. In random forest algorithm, each tree of forest resembles the decision tree. So, the random forest algorithm contains a random collection of a forest tree (decision tree). It is an advancement to the decision tree algorithm with more accuracy. After merging them, more stable and accurate prediction is obtained to increase the accuracy and robust prediction data set should be large. To increase the data set, we need to increase the number of trees in the forest.

3 Working and Output

3.1 K-NN Classifier

The first thing we did is we took the data and reduced the total number of variables. We cannot hard code the program; otherwise, it will not be useful for future use. So after the data conversion, we connect the code and the database. Size of the database provided will change every time. So 75% of the data is used as training and then the remaining 25% is used as a testing set. Note that more the data we have, better will be the accuracy. After loading the data, feature scaling is done. These algorithms only take in the magnitude of features ignoring the units. The results will vary greatly if only the magnitudes are considered and units are ignored. For example: there is huge difference between 5 kg and 500 gm. The features with high magnitudes will differ a lot more than those of lower magnitudes in spite of their units. To suppress this effect, we need to bring all features to the same level of magnitude. This can be achieved by scaling. After scaling, initialise the value of ‘k’. Always keep an odd value for ‘k’ so that we get one finite answer every time. Then, we fit K-NN to the training set. The output value for a defaulter is set as 1 and that for a non-defaulter student is set 0. Now suppose 25% of the testing data consists of five students, out of which three are a defaulter and two are not, then in that case we get an output as 1 or some value closer to 1 since the majority of a defaulter (Fig. 1).

3.2 Decision Tree

The data set is first to read into the code. Pandas is used for data manipulation and preprocessing the data. Data set has been split in the ratio of 7:3. Seventy per cent is the training set and the remaining is for testing. The answers to all the questions that the students had filled were converted into integer values for apparent reasons (Fig. 2).

All the attributes namely timestamp user name and all the integer values against their respective questions. Accuracy calculated was 0.983, i.e. 98%. The decision tree is created with all the mathematical terms visible for each node like the Gini index and the sample values. The sample value is the number of entries or cases that is considered for the calculations of that specific node.

The prediction for lastest entry is: 1

Fig. 1 Output of K-NN classifier

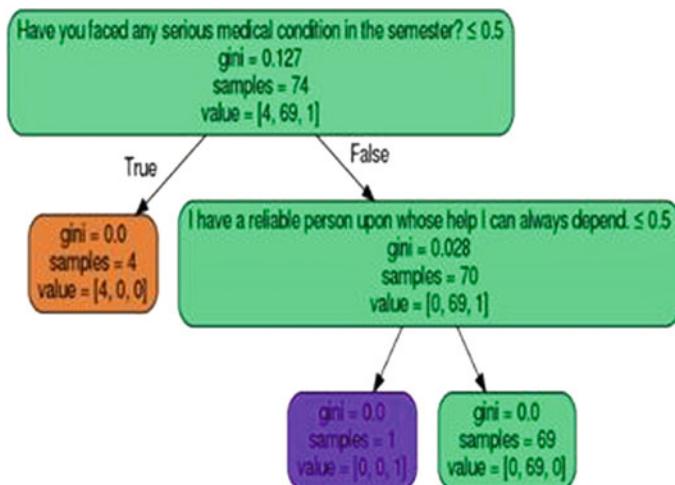


Fig. 2 Decision tree created

3.3 Random Forest Classifier

As we mentioned earlier, random forest can be used for classification which is used for the prediction of the defaulter student. In this case, classification is going to happen according to the attendance of the student and psychological behavioural survey. According to the responses of students to survey of analysing and training data, we can conclude about his defaulter status and the reason behind his less attendance, so that proper guidance can be given.

Because of the randomness in the model, while growing the trees, instead of searching for the most important feature like his attendance or psychological status while splitting a node, it gives preference to search the best feature among a random subset of features. Consider student 1, while working on his data, it will go as which features are effective in student 1. According to that, tree will be formed. It does not work like K-NN or Bayes classifier wherein we take the property first and then check whether it's available in a particular student. This gives large difference that generally results in a better model. This will give the most accurate result about student about his status of the defaulter. So, in random forest, while splitting a node, only a random subset of features is taken for the consideration. It is also possible to make trees more random. Instead of searching for the best possible thresholds (like a normal decision tree does), it can just be achieved by additionally using random thresholds for each feature.

Firstly, we conduct the psychological behavioural survey for all the students so that it will help us to compare the psychological differences in the students. According to the survey rating for each section like health section, feeling and thoughts, relation with important persons, etc., will be calculated. Now we have data from every student (defaulter and non-defaulter). That database will be used as training data. Now if data

of new student is entered, then it will give the prediction about the status of the student whether he is defaulter or not. If he is defaulter, then we can also predict the possible reason. According to that, proper guidance or consultation will be provided to the student, and so, we can help him to maintain his attendance. It will help to increase the overall attendance of students.

Now consider student 1 (Table 1).

Now for this student, the rating for perceptions and thoughts is 4. It is above average so there is the probability of student of the defaulter, so it may be the reason behind his less attendance. So faculty can guide the student as per necessity. So the output will be (Fig. 3)

Now consider student 2 (Table 2),

Now for this student, the rating for each student is below average. So the probability of this student of the defaulter is very less or equal to no. So the output will be (Fig. 4)

Table 1 Rating of defaulter student

Section	Health	Various areas of life	Perceptions and thoughts	Feelings and thoughts	Relation with important people	Attitude to life	Wishes and aspirations
Rating	3	2	4	1	1	2	2

```
In [268]: runfile('C:/Users/adity/Desktop/sneha.py', wdir='C:/Users/adity/Desktop')
```

```
The prediction for test data is:
```

```
Result Reason
1      Thought and Perception
```

```
In [269]:
```

Fig. 3 Output 1 of random forest classifier

Table 2 Rating of non-defaulter student

Section	Health	Various areas of life	Perceptions and thoughts	Feelings and thoughts	Relation with important people	Attitude to life	Wishes and aspirations
Rating	2	2	1	1	1	2	2

```
In [271]: runfile('C:/Users/adity/Desktop/sneha.py', wdir='C:/Users/adity/Desktop')

The prediction for test data is:

Result Reason
0      NVA
1    Thought and Perception
```

Fig. 4 Output 2 of random forest classifier

4 Observation

So by the analysis of the data, we observe that most of the students have health issues and problems with various areas of life which include studies, relation with friends or parents, performance in college, etc. So large-scale guidance can be given. Out of 207 students (aged 19–22) that gave the psychology test, almost 54% of defaulter students have a problem with various areas of life and 35% of students have health problems (Figs. 5 and 6).

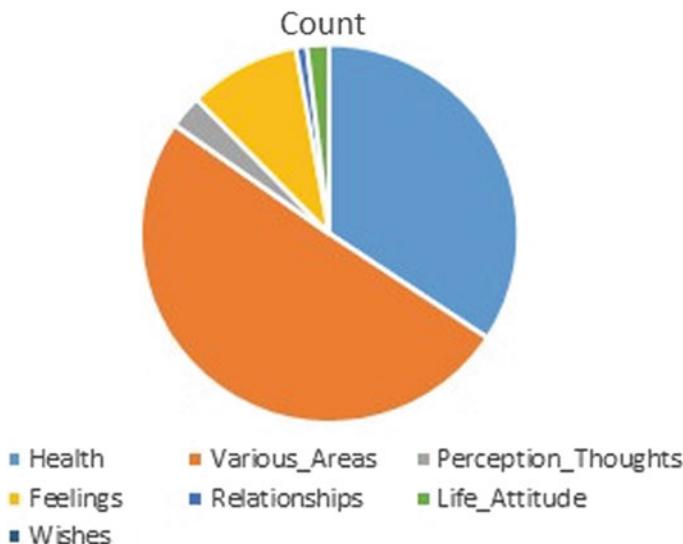


Fig. 5 Percentage of possible reasons that affect students

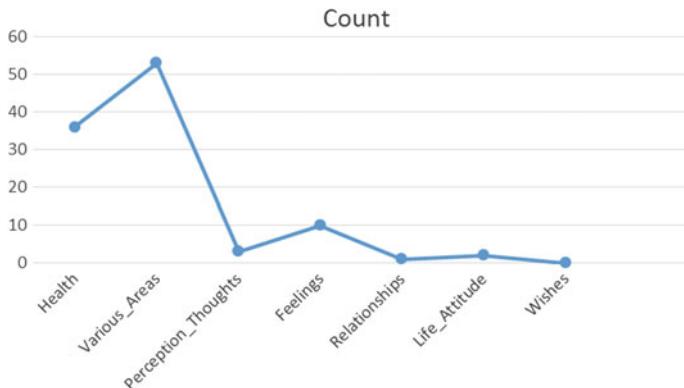


Fig. 6 Count of students affected by a particular reason

5 Conclusion

The research presented aims at predicting defaulter students and determines the possible reasons behind it. Comparing all three algorithms, we can see a similar trend in output with varying accuracy. By observing the results we can conclude that the primary cause of a student being a defaulter is: The Various areas of life and Health. Based on this, it is necessary to give required attention and guidance to the student in time before it impacts student's grades. Further analysis is in progress to create even better predicting algorithm.

References

- Prakash C, Gaikwad VB, Singh RR, Prakash O (Mar–Apr 2015) Analysis of emotion recognition system through speech signal using KNN & GMM classifier. In: IOSR J Electron Commun Eng (IOSR-JECE) 10(Issue 2, Ver.1), 55–61. e-ISSN: 2278-2834, p-ISSN: 2278-8735
- Thakare PP, Patil PS (2016 Dec) Facial expression recognition algorithm based on KNN classifier. IJCSN Int J Comput Sci Netw 5(Issue 6)
- Srividya M, Mohanavalli S, Bhalaji N (2018) Behavioral modeling for mental health using machine learning algorithms. Springer Science+Business Media, LLC, part of Springer Nature
- Gulraj M, Ahmad N (2016 Aug) Mood detection of psychological and mentally disturbed patients using machine learning techniques. IJCSNS Int J Comput Sci Netw Secur 16(8)
- Wei W, Han J, Kong J, Xia H, Prediction of the scholarship using comprehensive development. In: 2016 4th international conference on enterprise systems (ES)
- Hastie T, Tibshirani R, Friedman J (2009) Statistical learning. Springer, Berlin
- Feng YL, Wang SS, A forecast for bicycle rental demand based on random forests and multiple linear regression. In: 2017 IEEE/ACIS 16th international conference on computer and information science (ICIS)

Chapter 40

TNM Cancer Stage Detection from Unstructured Pathology Reports of Breast Cancer Patients



Pratiksha R. Deshmukh and Rashmi Phalnikar

1 Introduction

1.1 *Cancer Statistics in India*

India has one of the highest cancer cases and death rates in the world. Every year, over 11, 50,000 lakh new patients registered for cancer and out of those almost 4, 13,000 lakh in men, 3, 70,000 lakh in women and collectively 7, 80,000 lakh deaths are due to cancer. Breast cancer is the topmost cancer in Indian women [1]. These cancers can be stopped, identified prior and operated at low-level stage. This could help in reduction of the death rate because of these cancers [2].

1.2 *TNM Staging*

TNM staging is the most important prognostic factor. It represents stage of cancer patients and according to that specialist starts their treatment. From TNM, we get tumor size, its primary place, how many lymph nodes affected because of this tumor and whether this tumor start expansion in other parts of body or not. For breast cancer patients, it also shows that whether it is invasive or non-invasive. As these TNM values increases, severity for cancer increases and this TNM staging helps doctors to select treatment option based on their previous experience regarding same stage patients. Clinicians have to read these textual pathology reports which are mostly in unstructured form. For example pathology reports contain lesion place, lesion size it may be measures in cm or mm, clinicians will identify value of T using AJCC

P. R. Deshmukh (✉) · R. Phalnikar

School of Computer Engineering and Technology, MIT-WPU, Pune, Pune, India
e-mail: deshmukhpratikshar@gmail.com

cancer staging manual. Similarly according to information given about lymph node, clinicians will identify value of N. But writing skill and representation style of these reports vary from hospital to hospital; hence, there is a need of generalized system for cancer stage detection. Existing system for cancer staging works on structured input only.

1.3 *Cancer Registry*

Cancer registration is an essential part for analysis, statistics of cancer which helps to prevent, supervise and select appropriate treatment of cancer. In India, cancer registration was started in 1964 [3].

Existing cases and death rate information is collected by cancer registry. For particular cancer case, tumor registrars merged all information collected by registry. Most of the time, stage information collected by registry is improper and not complete. Tumor registrars of cancer registry perform manual cancer stage data collection required for cancer registry which may cause accidental human mistake and degrades data of cancer registry [4]. Most of Indian population-based cancer registries (PBCRs) do not gather cancer stage information which is important prognostic factor for statistical analysis of cancer [3].

This paper is organized as follows: Sect. 2 discusses related work to detect cancer stage from structured input and from unstructured input. Section 3 describes proposed generalized system for cancer stage detection. Section 4 describes methods as well as dataset used and analyzes experimental results of proposed system. Finally, Sect. 5 presents future direction of research and concludes the paper.

2 Related Work

Cancer staging helps to interpret patient's prognosis and helps to decide treatment option. Specialists have to read all reports thoroughly for detection of cancer stage. This is often tough process and takes lot of time.

For statistical analysis of cancer, cancer registries require cancer stage information of patients. Developing a software system which automatically extracts cancer stage information from medical reports is challenging task and number of researchers have contributed for this work.

Afzal et al. [5] developed forecast model to guide and support medical experts for decision making. They developed this model for head and neck cancer patients. They used classification and regression tree algorithm but they did not compare their results with existing system.

Martinez et al. [6] developed framework of colorectal cancer staging for two different hospitals. They used colorectal cancer pathology reports and found 20% mismatch between their results and gold standards. Nguyen et al. [7] developed

symbolic rule-based classification system for 718 lung cancer pathology reports. They observed that results for determination of primary stage tumor were lagging.

Johanna Johnsi Rani et al. [8] determined the cancer stage from natural language text report and SNOMED annotated text report. They proposed that cancer stage determination from SNOMED annotated text shows better results. They used 150 breast cancer pathology reports. Their study considered impression section only. They did not consider other section of reports. Warner et al. [9] proposed new approach for identification of stage summary from EHR data of lung cancer patients. They used 2327 lung cancer pathology reports but they considered single-institution study. Martinez et al. [10] designed a framework to detect significant categories from reports with minimum help of experts. They used 217 colorectal cancer pathology reports for their study.

McCowan et al. [11] used 700 lung cancer patient's textual histology reports for classification of T and N stage of cancer. They used machine learning techniques to classify these reports. Johanna Johnsi Rani et al. [12] developed automated system for breast cancer stage extraction. They used 150 de-identified pathology reports of breast cancer but they worked on impression section only. They did not consider other sections of the reports which also include important information about staging. Nguyen et al. [13] proposed symbolic rule base classification of lung cancer stages from textual pathology reports. They used pathology reports of 710 lung cancer patients. They compared their results with machine learning component for lung cancer staging system proposed by McCowan et al. [11] and showed that their results were encouraging.

The pathological TNM stage is determined based on surgery for resection of lesion or find the expanse of the cancer. Cancer registries require the use of pathological stage information to find the most accurate stage group.

Existing system works on structured input. Most previous studies have focused on structured input. To extract TNM stage information from free-text pathology reports, previous studies focused on symbolic rule-based system or machine learning system. Most previous works have concentrated on single-institution study and some of them have considered only some subsection of report. Most Indian PBCRs do not collect stage determination information [3].

3 Proposed Work

The proposed work has used reports from different hospitals and labs to make generalize clinical decision support system because clinicians and pathologist from different institutes have a different style of writing and interpreting reports. Proposed study focused on combination of pattern matching and rule-based technique to extract TNM stage information from free-text pathology reports which shows improvement in accuracy of stage detection of breast cancer patients.

This work will help all cancer hospitals to quick diagnose TNM stage of patient. Figure 1 shows architecture of proposed work. The main motivation of the proposed

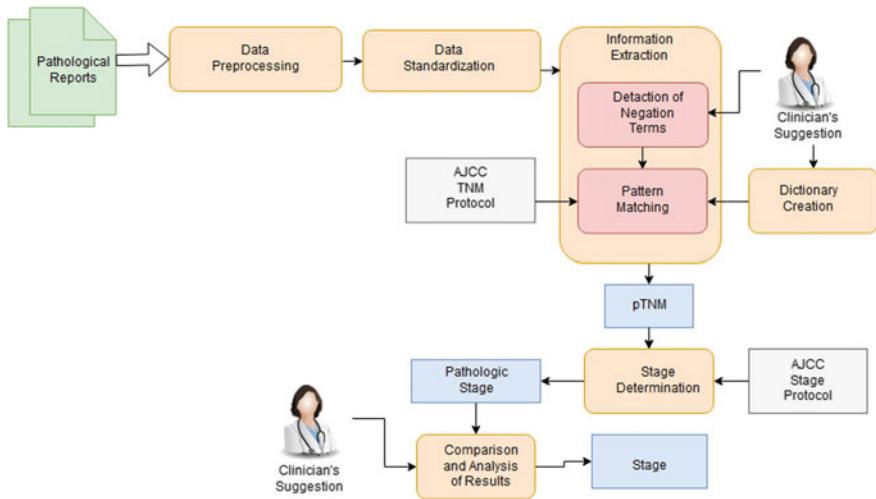


Fig. 1 Architecture of proposed work

methodology is to reduce the efforts of experts, to minimize manual data entries and to help in clinical decisions as well as research.

3.1 Data Preprocessing

This module performs section segmentation, sentence segmentation, text preprocessing and tokenization. Section segmentation will be done as pathological reports include different sections like specimen description, microscopic section, gross section, impression section, diagnosis section, etc.

3.2 Data Standardization

As clinicians and pathologist from different institutes have a different style of writing and interpreting reports, this function will include measure standardization, numerical value standardization and numeric representation.

3.3 Information Extraction

To extract size and place of tumor, number of lymph node affected and whether tumor is spread in different part of body or not this module will use pattern matching.

To extract these TNM values using pattern matching, we created table for T and N as per the instruction given in AJCC manual. When tumor cannot be assessed value of T is Tx and if tumor can be assessed then value of T ranges from T0 to T4. Value of T varies as per size, location and extension of tumor. Similarly when regional lymph node cannot be assessed, value of N will be Nx and if lymph node can be assessed its value ranges from N0 to N3. Value of N depends on number of positive lymph node when level I, II and III axillary lymph nodes get dissected.

M value cannot assess from pathological reports. It can assess through clinical report. But according to AJCC manual stage 0 to stage 3 value of M is M0 and for stage 4 value of M is M1. Hence, this work considers default M value as M0 for study.

3.4 Dictionary Creation

A number of medical terms have different synonyms. So this function creates dictionary for medical terms. For example let say some reports used ‘tumor’, other have used ‘ill defined mass’, ‘suspicious mass’, ‘lump’, ‘lesion’ to represent tumor.

3.5 Stage Determination

This will create stage protocol table from AJCC manual and determine stage for each TNM value using rule-based technique. Breast cancer stage ranges from stage 0 to stage 4. We can determine it with different combination of T, N and M values. When these values are Tis, N0 and M0, then it is stage 0 and this anatomic stage increases as value of T, N and M increases.

4 Results

4.1 Experimental Setup

This study used Python 2.7 to perform tokenization, preprocessing of the words, information extraction and performance analysis. The PyCharm with packages NLTK, pandas and NumPy is used for experimentation. Performance evaluation is done by calculating Precision, Recall and Accuracy of data. Table 1 shows confusion matrix for TNM.

Table 1 Matrix for TNM

		Predicted TNM value	
		TNM	Not TNM
Original TNM value	TNM	TP	FN
	Not TNM	FP	TN

Recall of TNM (R) = TP/(TP + FN)

Precision of TNM (P) = TP/(TP + FP)

Accuracy of TNM (A) = (TP + TN)/(TP + TN + FP + FN)

Table 2 Dataset details

Dataset	T value		N value	
Pathological reports 200 cases	Tx	20	Nx	20
	T0	59	N0	82
	T1	41	N1	68
	T2	60	N2	20
	T3	13	N3	10
	T4	7		

4.2 Dataset

Authors collected 200 pathological reports from renowned hospitals and labs in India. These reports are de-identified before using for study. Because of confidentiality of data, collection of dataset is the most difficult task in this study. Proposed work is practically significant as authors have used real dataset of breast cancer patients from some of the hospitals and labs in India. Ethics committee approval is taken for dataset and research study (Table 2).

4.3 Gold Standard

This study checked all reports from two medical experts in this field and considers their evaluation as gold standard. Authors compared system-generated stage value with stage evaluation given by clinicians.

4.4 Performance Analysis

- Figures 2 and 3 show that proposed work gives promising result with 80%, 83% and 93% accuracy for T, N and M values, respectively.

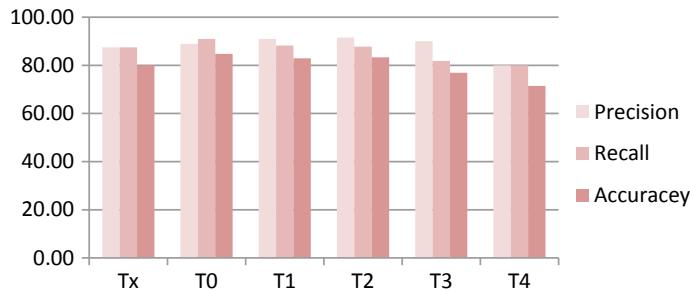


Fig. 2 Performance measure obtained by gold standard and proposed work for T

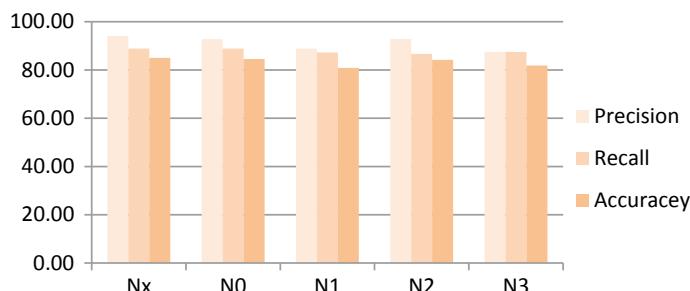


Fig. 3 Performance measure obtained by gold standard and proposed work for N

- Authors also used some reports where no malignancy found to test Tx, Nx and Mx value results.
- Detection of N value from information given on involvement of lymph node in reports was quite difficult part of implementation than extraction of information for T value.
- Following results that is Table 3 proved that the main goal of proposed work to make generalized system for breast cancer stage detection is achieved with strong performance accuracy.

Table 3 Performance measure for stage determination

Stage	TP	TN	FP	FN	P	R	A
Stage 0	60	10	4	6	93.75	90.91	87.50
Stage I	40	9	2	4	95.24	90.91	89.09
Stage II	25	5	3	2	89.29	92.59	85.71
Stage III	13	3	1	3	92.86	81.25	80.00
Stage IV	5	2	1	2	83.33	71.43	70.00

5 Conclusions and Future Work

This research and analysis are necessary because of rapidly increasing death rate due to breast cancer in Indian women. Stage detection helps the medical community to understand the most common stage of cancer patients at the time of diagnosis of the disease.

To find out most accurate stage of cancer clinicians require pathologic stage. This work will have practical applicability to diagnose and start treatment of breast cancer in Indian women.

In future, authors will extend their work for other cancer subtypes. Authors will try to more generalize work such that it will be suitable for various cancer hospitals though structures of report formats are different.

Acknowledgements Authors would like to acknowledge NDMCH as well as all those hospitals and labs for providing pathology reports. Authors would also like to acknowledge Ethics Committee to approve proposed research study. Authors would like to thank Dr. Nene, Dr. Joshi, Dr. Shilpa, Dr. Vibhute and Dr. Mane for guiding us in medical domain. Thanks to all patients to allow authors to use their reports and thanks Pratik Patil for his help.

References

1. Cancer Statistics in India, <http://cancerindia.org.in/statistics/>
2. Indian Council of Medical Research, <http://www.icmr.nic.in/>
3. Chatterjee S, Chattopadhyay A (2016) Cancer registration in India—current scenario and future perspectives. Asian Pac J Cancer Prev 17
4. National centre for Disease Informatics and Research, National Cancer Registry Program, <http://www.ncrpinIndia.org/>
5. Afzal M, Hussain M (2017) Comprehensible knowledge model creation for cancer treatment decision making. In: Computers in biology and medicine. Science Direct, Elsevier, Amsterdam
6. Martinez D, Pitson G (2014) Cross-hospital portability of information extraction of cancer staging information. Artificial Intelligence in Medicine. Elsevier, Amsterdam
7. Nguyen AN, Lawley MJ (2010) Symbolic rule-based classification of lung cancer stages from free-text pathology reports. J Am Med Inform Assoc 17:440–445. <https://doi.org/10.1136/jamia.2010.003707>
8. Johanna Johnsi Rani G, Gladis D (2017) Comparison of breast cancer staging in natural language text and snomed annotated text. Int J Pure Appl Math 116(21):243–249
9. Warner JL, Levy MA (2016 Feb) Feasibility and accuracy of extracting cancer stage information from narrative electronic health record data. American Society of Clinical Oncology 12(Issue 2)
10. Martinez D, Li Y (2011) Information extraction from pathology reports in a hospital setting. In: CIKM'11, Oct 24–28 2011. ACM
11. McCowan I, Moore D (2007) Classification of cancer stage from free-text histology reports. IEEE, New York
12. Johanna Johnsi Rani G, Gladis D (2015) Breast cancer staging using natural language processing. IEEE, New York
13. Nguyen A, Moore D (2007) Multi-class classification of cancer stages from free-text histology reports using support vector machines. IEEE, New York

Chapter 41

Restructuring of Object-Oriented Software System Using Clustering Techniques



Sarika Bobde and Rashmi Phalnikar

1 Introduction

An ideal characteristic of software in the present era is its necessity to evolve. In recent developing software paradigm, the widely used object-oriented model is regarded as a significant aspect. Reduced cost and time are considered as the two important attributes of quality in software engineering, and quality of the software system is highly expected. According to the researchers in recent years, object-oriented model is considered as an effective tool for developing high-quality large software systems [1]. Here, high-quality system means it is easy to maintain, reusable, and simply extensible according to requirement. Cohesion is a major characteristic of object-oriented software development. The design quality of a software system is decided by its cohesion [2, 3].

When a software system undergoes maintenance processes such as modification and adaptation according to new demands, the complexity of the code increases and deviates from its novel design, resulting in poor quality of the software. Due to this, software maintenance consumes most of the part of the total software development price. Introducing improved software development techniques and means also does not solve this problem, since they occupy more space resulting in additional requirements, thereby making the system much more complex. To handle this issue, new techniques are needed which will improve the software internal quality by limiting the software complexity. The terms coined for such domain that attends to this problem are restructuring, remodeling, or refactoring [2].

S. Bobde (✉) · R. Phalnikar

School of Computer Engineering and Technology, MIT-WPU, Pune, Pune, India
e-mail: sarika.bobde@mitwpu.edu.in

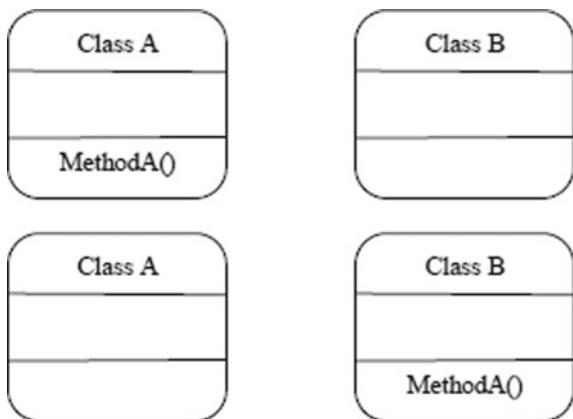
R. Phalnikar

e-mail: rashmi.phalnikar@mitwpu.edu.in

Fig. 1 Refactoring

Refactoring applies tiny changes on the source code by preserving the behavior of software (Fig. 1). An example of refactoring operation is when developer moved method from one class to another class. This refactoring is called as move method which is used when method is more used in another class than its own class. There are several other refactoring types that are used for different purposes. Fowler suggested different refactoring types such as move method, move field, extract method, extract class, pull-up field, pull-up method [4] (Fig. 2).

A continuously changing software system that undergoes nonstop change, such as having new functionality added to its original design, will ultimately become more complex and can become incompetent as it grows and hence will lose its original design structure known as entropy. Software entropy can create spots in the software source code, thereby affecting the software interior design quality. These spots are symbol of design defects, also known as bad smells. Therefore, owing to these problems the source code becomes erroneous, hard to understand and unfavorable to upcoming changes [5, 4]. Refactoring provides a method to remove bad smells and enhance the structure of software system without affecting its external performance. Though by hand, finding and implementing functional refactoring techniques are a great challenge. Fully automated and semiautomated refactoring schemes were proposed by some in the literature [6–8].

Fig. 2 Example of move method

Clustering algorithms are used to group objects which are related to one another in some way. They have been widely used to detect subsystems, derive features from object interactions, and provide software views of applications in cohesive systems as a restructuring technique [9]. Clustering assigns individual data points to exactly one cluster by producing sharp partitions for each dataset. Nevertheless, the need for data points with multiple and unbranched cluster relationships has led to the emergence of fuzzy clustering algorithms. Fuzzy c-means (FCM), a fuzzy partitioned clustering scheme is a popular fuzzy-based clustering algorithm among all other methods, and it is a generalized development of k-means algorithm. Due to the factors like large computational load and relation to k-means, it is expected that implementation of the FCM algorithm in proposed restructuring would significantly improve the system performance [10].

Any refactoring scheme should comprise of the following steps.

- i. Analyze whether there is a need for the software to be refactored.
- ii. Finding the place, in which, this factor needs refactoring.
- iii. The suggested refactoring method should guarantee and it won't change the system behavior.
- iv. Implementation of refactoring technique.
- v. Evaluate the impact of the proposed refactoring on software system quality in both quantitative and qualitative attributes.
- vi. Managing the balance between the refactored code and other software components [11, 12].

The rest of this paper is ordered as follows: Sect. 2 provides a literature survey, Sect. 3 provides proposed method for software restructuring using clustering, and finally, Sect. 4 concludes.

2 Literature Review

Search-based approach [8, 4] has been proposed for refactoring of software system. Fowler [4] proposed an automated refactoring by genetic algorithm-based approach using fineness function to remove bad smell. These techniques identify the best solution to reduce bad smell. Seng et al. [8] propose an approach that uses a unique model to simulate refactoring with all essential pre- and post-conditions with mutation, crossover operators, and a fitness function.

Clustering techniques [3, 7, 13] are another way to deal with restructuring of object-oriented system: Clustering uses to reconstruct the class formation of software system and to find the appropriate refactoring. Rathee and Chhabra [3] proposed a new refactoring algorithm to restructure the source-based using hierarchical agglomerative clustering and introduced a new similarity metric using frequent usage patterns (FUP). The proposed refactoring algorithm improves the cohesion of classes by enhancing quality of software system. Han et al. [13] studied an adaptive refactoring method to get a better structure of the software system effectively without failing

the accuracy of the final outcome. They proposed a hierarchical agglomerative adaptive clustering algorithm. Kebir et al. [7] focused on restructuring object-oriented system by employing k-medoids-based clustering algorithm.

Automated approach to improve the software maintenance [14] is another approach to deal with restructuring. [14] Used maximal independent set (MIS) to find multiple refactoring functions which could be implemented at the same time.

Graph transformation [9] is another way to deal with extract class refactoring opportunities. [9] proposed approach that uses a weighted graph to characterize a class to be refactored. Author used a MaxFlow-MinCut algorithm to divide the built graph which can be used to create two new classes having higher cohesion than the original class.

By analyzing all the literature review, we come to a conclusion that still many things need to be achieved in “code refactoring.” So as a solution the coming edition mainly focuses on “a model for enhancing the performance of the software code.” However, in this paper, we propose a novel method for software restructuring using clustering with multiple refactoring methods. Clustering algorithm will be used to get better design of software system.

3 Proposed Methodology

Cohesion is regarded as an important quality characteristic in object-oriented software development, which is selected to evaluate the design quality of a software system. Due to the adaption of new requirements in software, the code becomes more complex, thereby decreasing the cohesion. So our proposed approach aims at increasing the cohesion of the software system by lowering the maintenance cost.

The proposed approach will be shown in Fig. 3.

The basis code of the software is taken as input with classes, and then member variable sets are drawn out along with a set of member functions from the classes present in the software. Next, the model of member variables between different member functions is computed using vector-based approach. Then base on the similarity values, clustering of member functions is completed using fuzzy c-means clustering. The member functions of each cluster are refactored and optimized according to actions recommended by the proposed refactoring algorithm.

3.1 Extracting Member Variables

All the member variables (public, private, and protected) are extracted by extract class refactoring that is defined by the classes of the software system. The data of the member variable combined with the information concerning their respective class is noted in a data structure, which later helps in detecting which member variable belongs to which class in the proposed refactoring algorithm.

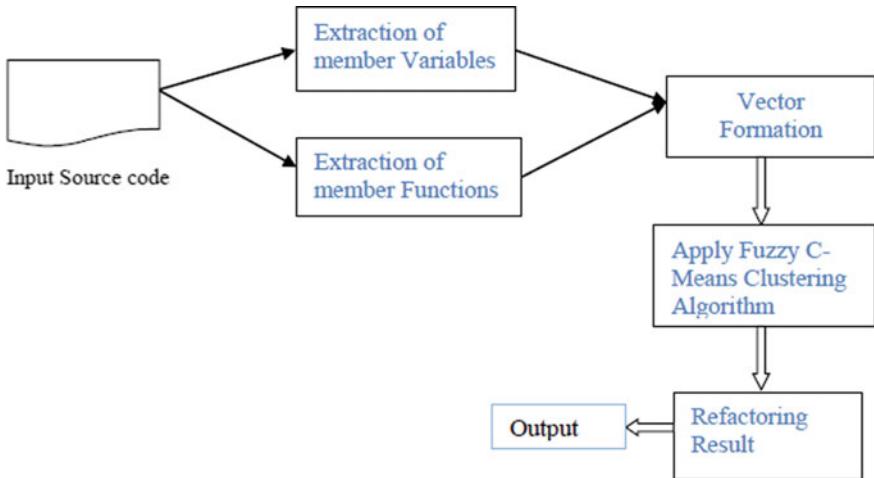


Fig. 3 Proposed methodology

3.2 *Extracting Member Functions*

Here, all the public and private member functions are extracted that are defined by the classes of software.

3.3 *Clustering Algorithm*

Our proposed approach employs fuzzy c-means clustering algorithm. Introduction of fuzzy c-means clustering algorithm is given below.

Fuzzy c-means clustering derived from hard c-mean clustering method that is introduced in 1981, and it comes under unsupervised clustering algorithm. It is used in solving problems which are associated with clustering, feature analysis, and classifier structures. FCM application includes astronomy, image analysis, target recognition, agricultural engineering, geology, medical diagnosis, shape analysis, and chemistry. The growth of the fuzzy theory led to the expansion of Ruspini fuzzy clustering theory to the FCM clustering algorithm which is actually was proposed in 1980s. This scheme is utilized for analysis based on distance between many input data points.

In recent years, fuzzy clustering has been developed extensively. In various real applications, when the limits between classes are not distinct, fuzzy clustering performance is better than other hard clustering methods. Fuzzy clustering is based on minimizing objective function to determine an ideal classification. In fuzzy clustering, every cluster that is based on objective function is exhibited by a cluster model. This model comprises of a cluster center and also a few more data concerning the

size and the shape of the cluster. Like the datasets used in data points to separate, cluster center denotes the attributes that are used to explain the domain, but since the clustering algorithm manages the cluster center; it may or may not be represented in the dataset. The shape and size parameters decide the expansion of clusters in the various directions in the essential field. The FCM algorithm has been applied to many clustering problems in the past, and it is basically an iteration technique which is used to divide a dataset. The main aim of FCM partition is to evaluate the cluster centers and produces the class membership matrix. For initialization, “K-center” technique is used. The fuzzy clustering divides the system data such as the objects, their distance, and mean samples into K overlapped clusters according to the obtained minimizes of the fuzzy least squares function.

The steps involved in fuzzy c-means clustering are as follows:

The degree of relationship between the present object and the various clusters is represented by the initial matrix of FCM algorithm.

Let $x = \{x_1, \dots, x_n\}$, and every x_k has m attributes, $x_k = \{x_{k1}, \dots, x_{km}\}$.

1. Transformation of the eigenvalue takes place.
2. Weight is assigned to all attributes, where the weight represents the importance of the attributes, respectively.
3. Relative importance of the samples is obtained by assigning $y = x$ and $w = (y_1, \dots, y_n)$.
4. Based on the real time needs, the distribution of y is fixed. Here it is said that the samples fit the even distribution. In few cases, with idiographic instances possibly will not meet the requirement; hence, a few optimum distribution functions should be chosen for satisfying the realistic needs.
5. Based on the real-time needs, the distribution set of all clustering center is fixed. This step assumes that every individual section satisfies the normal distribution.

3.4 Expected Outcome

The result obtained by applying the proposed approach will be validated by comparing it with existing technique. Also, clustering algorithm will be employed for identifying refactoring in order to improve the design of software system. Identifying which refactoring method to apply will increase the cohesion of software and the effectiveness of system. Restructuring using clustering increases the reliability and decreases the maintenance cost of the software.

4 Conclusion

Code refactoring is an important area of research which is used to improve the software maintenance. In this paper, we put forward a novel method to develop well-designed object-oriented software system using refactoring which will increase the cohesion of the code. Rising cohesion of a class results in an eventual reduction in coupling due to refactoring. There are various methods available for the same. Clustering is one of the approaches used to restructure the software system to overcome code defect which helps in growing reliability and falling maintenance effort of the software. This work will propose multiple refactoring methods in source code to get a better design of software system.

Future work can study the other learning technique to improve the software. Another future research work can use other similarity measures to evaluate the software system.

References

1. Hudli R, Hoskins C, Hudli A (1993 Jan) Software metrics for object-oriented designs. *IEEE Trans. Electron Devices* 2:314–319
2. Mens T (2004) A survey of software refactoring. *IEEE Trans Softw Eng*, 126–139
3. Rathee A, Chhabra JK (2017) Restructuring of object-oriented software through cohesion improvement using frequent usage patterns. *ACM SIGSOFT Softw Eng Notes* 42(3):1–8
4. Fowler M (1999) Refactoring—improving the design of existing code. Addison Wesley
5. Harman M, Tratt L (2007) Pareto optimal search based refactoring at the design level. In: Proceedings of the 9th annual conference on genetic and evolutionary computation, pp 1106–1113
6. Hegedűs P, Kádár I, Ferenc R, Gyimóthy T (2018) Empirical evaluation of software maintainability based on a manually validated refactoring dataset. *Inf Softw Technol* 95:313–327
7. Kebir S, Borne I, Meslati D (2017) A genetic algorithm-based approach for automated refactoring of component-based software. *Inf Softw Technol* 88:17–36
8. Seng O, Stammel J, Burkhardt D (2006) Search-based determination of refactorings for improving the class structure of object-oriented systems. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp 1909–1916
9. Ducasse S, Pollet D (2009) Software architecture reconstruction: a process-oriented taxonomy. *IEEE Trans Softw Eng* 35(4):573–591
10. Kwok T, Smith K, Lozano S, Taniar D (2002 Aug) Parallel fuzzy c-means clustering for large data sets. In: European conference on parallel processing, pp 365–374. Springer, Berlin
11. Czibula IG, Czibula G (2010) Hierarchical clustering for adaptive refactorings identification. In: IEEE international conference on automation, quality and testing, robotics, pp 1–6
12. Serban G, Czibula IG (2007) Restructuring software systems using clustering. In: 22nd international symposium on computer and information sciences, ISCIS 2007, pp 1–6
13. Han AR, Bae DH, Cha S (2015) An efficient approach to identify multiple and independent move method refactoring candidates. *Inf Softw Technol* 59:53–66
14. Fokaefs M, Tsantalis N, Chatzigeorgiou A, Sander J (2009) Decomposing object-oriented class modules using an agglomerative clustering technique. In: Proceedings of the 25th international conference on software maintenance. Edmonton, Canada, pp 93–101

Chapter 42

Analysis of System Logs for Pattern Detection and Anomaly Prediction



Juily Kulkarni, Shivani Joshi, Shriya Bapat and Ketaki Jambhalı

1 Introduction

Log data is a definitive record of what's happening in every business, organization, or agency and its often an untapped resource when it comes to troubleshooting and supporting broader business objectives [1]. Asking a virtual personal assistant for help in debugging a production system may seem like a far-fetched idea, but the idea of using a machine learning approach is actually very feasible and practical. Machine learning algorithms are very useful in recent years at solving complex problems in many fields. From computer vision to autonomous cars to spam filters to medical diagnosis, machine learning algorithms are providing solutions to problems and solving issues where once expert humans were required [2]. System logs record the states of the system at various stages and important events at various critical points to help to understand performance issues and failures. This log data is universally available in all computer systems. As system logs record events from actively running processes, they become an important resource for anomaly detection.

Anomaly detection is an important task if we want the system to be secure and efficient. As the systems get more advanced and complex, anomaly detection becomes an essential task.

Our work proposes an intelligent system which uses parsed and filtered system logs to detect anomalies using deep learning and artificial intelligence approaches. Patterns are classified and detected from these system logs, and these patterns are used for error detection. Based on the anomaly detection performed, remedial measures are taken, wherein an alert message is sent to the user and system takes appropriate remedial measures. This work helps in detecting and fixing the critical situations which may arise in the future so as to avoid the loss of time, memory, etc.

J. Kulkarni (✉) · S. Joshi · S. Bapat · K. Jambhalı
Department of Computer Engineering, PVG's COET, 411009 Pune, India
e-mail: Juilykulkarni@gmail.com

2 Related Work

According to our survey, there exist certain systems which perform frequent pattern mining and anomaly detection. Each system has its unique features and methodologies. We have tried to improve upon these systems by predicting future anomalies and performing auto-remediation for the same.

Min Du, Feifei Li, Guineng Zheng, Vivek Srikumar have proposed the DeepLog system for HDFS and OpenStack log datasets. This method uses a deep neural network-based approach. It performs anomaly detection at per log entry level. This paper uses both deep learning approach using LSTM algorithm and classic mining methods such as clustering [3].

S. VijayaKumar, A. S. Kumaresan, U. Jayalakshmi proposed a system to predict new interesting patterns from Web server access log files. They have applied Apriori algorithm for matching new interesting patterns from the log data and applied support and confidence to calculate the measures of the found patterns [4].

3 System Architecture

See Fig. 1.

Architecture consists of the following components:

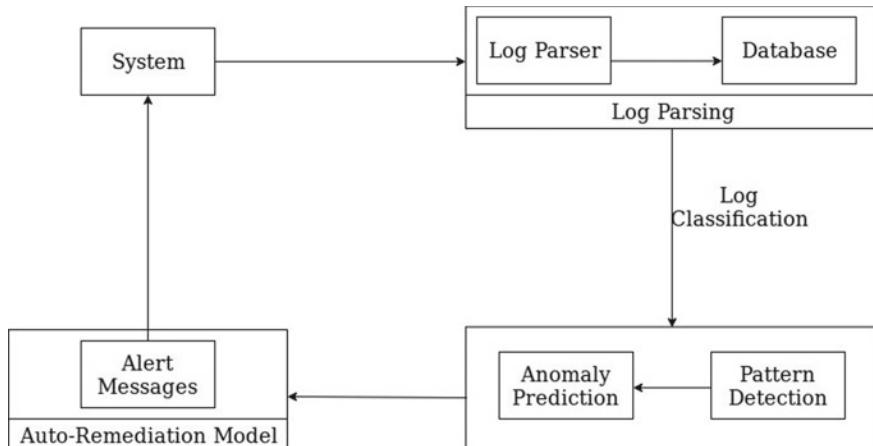


Fig. 1 System architecture

3.1 Log Parsing

System logs are parsed using Logstash, and parsed logs are passed to Elasticsearch. Elasticsearch is used as database to store and view logs in JSON format. Postman is used for visualization of logs.

3.2 Pattern Detection and Anomaly Prediction

Logs stored on Elasticsearch are classified, and data mining and machine learning algorithms are used to detect patterns from the logs stored in Elasticsearch. From these patterns, any errors or anomalies in the system are detected.

3.3 Auto-remediation

In this module, alert messages are passed to the action-mapping mechanism and remedial measures are taken by the system.

4 Proposed System

The proposed system is a system for analyzing system logs to detect anomalous patterns and predicting future anomalies. Auto-remediation is done based on these predicted anomalies.

The proposed system is implemented as follows:

4.1 Log Parsing

System logs are used here. Format of syslog is shown in Fig. 2.

Logstash configuration is made up of three parts:

- (1) An input file that contains system and authentication logs

```
30 2019-06-28T13:09:15.262051+05:30 admin-PC systemd[1]: Stopping LSB: disk temperature monitoring daemon...
30 2019-06-28T13:09:15.264275+05:30 admin-PC thermald[804]: Terminating ...
30 2019-06-28T13:09:15.264617+05:30 admin-PC systemd[1]: Stopping Thermal Daemon Service...
30 2019-06-28T13:09:15.266613+05:30 admin-PC systemd[1]: Stopping LSB: Record successful boot for GRUB...
30 2019-06-28T13:09:15.267082+05:30 admin-PC dispatcher[1]: Stopping Dispatcher daemon for systemd-networkd...
```

Fig. 2 System logs

```

input {
  file {
    path => "/var/log/syslog"
    type => syslog
    start_position => "beginning"
    codec => multiline {
      pattern => "^\\["
      negate => true
      what => previous
    }
  }
}

```

Fig. 3 Logstash input

- (2) A Grok filter, that is used to parse its contents to make a structured event
- (3) An Elasticsearch output.

4.1.1 Logstash Input Plugin

Here path of the log file is taken as input. The type of the log messages is specified as syslog. The start position of the log file is specified as ‘Beginning’ as the log messages are read from beginning. In this module, multiline codec is used to merge the multiple log lines into a single event as per requirement. In multiline codec pattern, option specifies a regular expression. Lines that match the regular expression are considered as either continuation of the previous line or start of the new multiline event. The ‘what’ option in multiline codec takes two values previous and next. The previous value specifies that lines that match the value in pattern option are the part of the previous event/line and the next value specifies that lines that match the value in pattern option are part of the following/next line/event. Negate option applies to the multiline code that does not match the regular expressions specified in the pattern option (Fig. 3).

4.1.2 Logstash Filter Plugin

Filters are used to take the raw log data and use it to parse them. Logstash has lots of such plugins, and in this project, we are making use of dissect filter, Grok filter, and date filter.

```

filter []
  dissect
  {
    mapping => { "message" => "%{priority} %{ts} %{host} %{process_name}: %{msg}" }
  }
  grok
  {
    match => { "path" => "%{GREEDYDATA}/%{GREEDYDATA:type}" }
  }
  if [type] == "syslog" {
    grok {
      match => { "message" => "<%{POSINT:syslog_pri}> %{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}?%{GREEDYDATA:syslog_message}" }
    }
  }
}

```

Fig. 4 Logstash filter

1. Dissect filter is used to extract the structured fields from unstructured line. Mapping for time stamp (ts), hostname (host), program name (program), log type (loglevel), process ID (id), and log message (msg) is done using dissect filter.
2. Grok filter is used to map the logline into the following fields: time stamp, log level, hostname, process id, program name and then the rest of the message.
3. Date filter is used to parse the date fields from the logs and convert them into the Logstash event time stamp format which is useful for searching and sorting (Fig. 4).

4.1.3 Logstash Output Plugin

Output plugin sends event data to a particular destination. Elasticsearch output plugin is used to store all the log events on the Elasticsearch. Hostname of the Elasticsearch (localhost) and port number is specified (9200), and index name (log_index4) is specified. Using the index name, the parsed logs can be seen on Elasticsearch in JSON format (Fig. 5).

Searching and querying take the format of:

[http://localhost:9200/\[index\]/\[type\]/\[operation\]](http://localhost:9200/[index]/[type]/[operation])

For example, in our project the following url consists of the parsed log records stored on Elasticsearch:

```

output
{
  elasticsearch{
    hosts => ["localhost:9200"]
    index => "log_index1"
  }
  stdout { codec => rubydebug }
}

```

Fig. 5 Logstash output

http://127.0.0.1:9200/log_index1/_search?size=100

Here [index] = log_index1

4.2 Pattern Detection and Anomaly Prediction

This module deals with the detection of patterns leading up to error messages and then the prediction of anomalies from this historical data. Before beginning pattern detection, the message part and the priority field are extracted from the entire parsed log. These two fields are required for further analysis. The type of the message is found using priority field. From this, facility and severity are extracted using the formula:

$$\text{facility} = \text{priority}/8 (\text{Here the whole number part is the facility})$$

$$\text{severity} = \text{priority} - (\text{facility} * 8)$$

The severity found gives us the type of the message. Here we take into consideration error messages.

4.2.1 Pattern Detection

A pattern could be described as something that appears frequently in a database [5]. Pattern detection is done using Apriori algorithm. After extracting the message part from the entire log, sequential unique mapping of the log messages is done. This sequence of messages is then divided into chunks of messages. These chunks are made such that a chunk ends at the first occurrence of an error message and the next chunk starts from there till the next occurrence of an error message.

These chunks are then passed on to the Apriori algorithm. Apriori algorithm is one of the most used and successful algorithms for frequent pattern mining. It makes use of frequent itemsets where the item with the minimum support is considered.

Apriori algorithm does the work of finding frequently occurring patterns from these chunks. Apriori algorithm makes use of support which indicates the number of times a particular pattern occurs through the file and length which indicates the length of each found pattern. Here, we take into consideration only the patterns which end with an error message as these will be the patterns detected due to which an error message will occur. After detection of all such patterns, patterns with a support greater than 1 and length greater than 2 are extracted and passed on to anomaly prediction model for further analysis.

For example, (see Fig. 6).

6	Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).
6	Listening on UUID daemon activation socket.
6	Started ACPI Events Check.
6	Started CUPS Scheduler.
3	PKCS#7 signature not signed with a trusted key

Fig. 6 Example of pattern found using Apriori algorithm

4.2.2 Anomaly Prediction

Anomaly prediction is done using long short-term memory (LSTM) algorithm. The function *split_sequence()* splits the data into dataX and dataY. Log messages in the patterns before the occurrence of the error message are stored in dataX, and the error message is stored in dataY.

Then LSTM network model is created and fitted using the sequential model. The model expects the input shape to be three-dimensional with *[samples, timesteps, features]*; therefore, we are reshaping (the array) the single input sample into a format expected by the LSTM networks, that is *[samples, time steps, features]* before making the prediction. The shape of the input for each sample is specified in the *input_shape* argument on the definition of the first hidden layer. Before training the model, we are configuring the learning process, which is done using the *compile* method. It receives the following arguments:

1. An optimizer – Adam optimizer.
2. A loss function. Mean-squared error loss function as it is a good match for our chosen error metric of RMSE.

The model is then fitted for 3000 epochs. After the model is fit, we can use it to make a prediction. Once the model is fit, we are estimating the performance of the model on the train and test datasets using *model.predict*.

Prediction of the next value in the sequence is done by providing the input: *x_input = array([a,b,c])*

where *a, b, c* contain the pattern in X array leading to an error message. For example, *a = 40, b = 41, c = 42*. And expect the model to predict: 43 (Fig. 7)

4.3 Auto-remediation

In this module, based on the anomalies detected alerts are sent to the system and remedial measures are performed accordingly. Remedial measures are taken based on the type of anomaly predicted. For example, if the predicted anomaly indicates a system failure, then remedial measures like system restart will be taken in advance in order to avoid such a type of future critical situation.

```

n_features = 1
n_seq = 2
n_steps = 2
X1 = X.reshape((X.shape[0], n_seq, n_steps, n_features))
print("reshape over")
# define model
model = Sequential()
model.add(
    TimeDistributed(Conv1D(filters=64, kernel_size=1, activation='relu'), input_shape=(None, n_steps, n_features)))
model.add(TimeDistributed(MaxPooling1D(pool_size=1)))
model.add(TimeDistributed(Flatten()))
model.add(LSTM(50, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
# fit model
model.fit(X1, y, epochs=2000, verbose=0)
# demonstrate prediction

```

Fig. 7 LSTM algorithm

5 Tools Used

5.1 Logstash

Logstash is a lightweight, open-source, server-side data processing pipeline that allows you to collect data from a variety of sources, transform it on the fly, and send it to your desired destination.

Logstash is a popular choice for loading data into Elasticsearch.

5.2 Elasticsearch

Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases [6]. Elasticsearch is scalable up to petabytes of structured and unstructured data and can be used as a replacement of MongoDB. It is open-source and available under the Apache license version 2.0. It can store, search, and analyze big volumes of data quickly and in real time. It is generally used as the underlying technology for applications with complex search features and requirements. It is optimized for needle-in-haystack problems rather than consistency or atomicity.

5.3 TensorFlow–Keras

TensorFlow is a computational framework for building machine learning models [7].

Keras models can be developed with a range of different deep learning backends. You can train a model with one backend and load it with another (e.g., for deployment) [8]. Available backends include

The TensorFlow backend (from Google)

Keras API comes packaged in TensorFlow as tf.keras
Keras is an open-source library written in Python for solving neural network problems.

6 Conclusion

This paper presents the model for analyzing system logs and erroneous pattern detection as well as anomaly prediction. The proposed system parses the machine logs into different fields. Log classification will be carried out based on the priority field. Frequent patterns leading to erroneous logs are found out using Apriori algorithm. These patterns will be further used for future error prediction which makes use of LSTM algorithm and taking remedial measures to avoid further risks.

Outcomes

Based on the sequence of the log messages or the pattern of log messages, future error is predicted.

For example,

Pattern of syslog messages:

- (1) action ‘action 1’ suspended (module ‘builtin:omfwd’), retry 0. There should be messages before this one giving the reason for suspension. [v8.32.0 try <http://www.rsyslog.com/e/2007>]
- (2) action ‘action 1’ resumed (module ‘builtin:omfwd’) [v8.32.0 try <http://www.rsyslog.com/e/2359>]

Predicted error message based on the above pattern:

omfwd: socket 12: error 101 sending via udp: Network is unreachable [v8.32.0 try <http://www.rsyslog.com/e/2354>]

If the probability of occurrence of log messages in a pattern is more than 50%, the alert message is sent.

Acknowledgements We thank Mr. Samir Sood (Harman Connected Services Corp. India Pvt. Ltd.) for his support, help, and guidance without which this research would not be what it is.

References

1. https://www.splunk.com/en_us/solutions/solution-areas/log-management.html
2. <https://logz.io/blog/machine-learning-log-analytics/>
3. Du M, Li F, Zheng G, Srikumar V (2017) DeepLog: anomaly detection and diagnosis from system logs through deep learning
4. Vijaya Kumar S, Kumaresan AS, Jayalakshmi U (2015 Oct) Frequent pattern mining in web log data using apriori algorithm. Int J Emerg Eng Res Technol 3(Issue 10)

5. Savio MND (2016) Predicting user ‘s future requests using frequent patterns. San Jose State University, SJSU ScholarWorks (Fall 12-19-2016)
6. <https://www.elastic.co/products/elasticsearch>
7. <https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit>
8. <https://mc.ai/tensorflow-playing-with-tensors/>

Chapter 43

Phishing Detection: Malicious and Benign Websites Classification Using Machine Learning Techniques



Sumit Chavan, Aditya Inamdar, Avanti Dorle, Siddhivinayak Kulkarni and Xin-Wen Wu

1 Introduction

Phishing is a technique in which personal information of the user is gathered using emails and websites. The goal of this technique is to make user believe that the emails that they have received or the websites which they are currently using are legitimate. It is 2019, and most users have an understanding about some basic types of phishing scams. According to the 2015 McAfee survey [1] there were around 97% of users who were not able to correctly recognize phishing emails, which means that even though users have basic understanding of phishing emails they are unable to avoid them in practice. The positive aspect is that even fewer Internet users have been clicking on malicious website URL and fake email links. As per the Internet

Sumit Chavan, Aditya Inamdar, Avanti Dorle—contributes equally.

S. Chavan (✉) · A. Inamdar

Department of Computer Engineering, MITCOE, Pune, India

e-mail: chavansumit13@gmail.com

A. Inamdar

e-mail: adityapinamdar@gmail.com

A. Dorle

Department of Information Technology, PICT, Pune, India

e-mail: avanti.dorle10@gmail.com

S. Kulkarni

Department of Computer Engineering, MIT-WPU, Pune, India

e-mail: siddhivinayak.kulkarni@mitcoe.edu.in

X.-W. Wu

Department of Mathematical and Computer Sciences,

Indiana University of Pennsylvania, Indiana County, USA

e-mail: xwu@iup.edu

security company Cofense [2], there was a 2% decrease in the authoritative susceptibility rates in the span of the year 2016–2017, which was actually then reduced to 10.8%. The best way to tackle the tendency of the user employees to get caught in the trap of these malicious websites is by providing better exposure for identifying and understanding phishing emails. There has been an observed decline in the susceptibility rate of repetitive phishing simulations, especially those that are based on relevant, upcoming threats. According to the survey of Wombat [3], there has been following trends seen in impact of phishing to organizations: There has been 27% of malware infection in 2016 whereas 49% of malware infection in 2017. There have been 17% of compromised accounts in 2016 whereas 38% of compromised accounts in 2017. There has been 7% loss of data in 2016 whereas 13% loss of data in 2017. The statistics clearly tells us the impact of phishing is high in spite of the consumers being aware of the phishing websites. Machine learning is an effective intelligent method to detect phishing and reduce the impact of it on organizations. Machine learning has supervised and unsupervised learning algorithms, out of which the problem of phishing detection can be thought of as a classification problem in supervised machine learning model.

Usually there are five categories of phishing. The first one is known as ‘Vishing’ that is voice phishing. It is in reference to the phishing attacks related to telephone conversations. The next category is ‘Smishing’ which is SMS phishing. This is an easy type of phishing. One of the categories being ‘Search Engine phishing’ which is in reference to the design of fraudulent webpages that aims at certain keywords to wait for user search and arrive on the fake webpage. The last two categories are ‘Spear phishing’ and ‘whaling’. In spear phishing, the email is send to over a million users whereas in whaling the targeted group becomes more specific. For all the categories of phishing that are present there are different types of phishing attacks that are possible. Some of them include clone Phishing attack, image phishing attack, CEO fraud attack and scripting.

Real time prediction of the type of website is performed by the models at the time when the user browses any new webpage. This result is communicated to the end-user. The most important part in the development of automated anti-phishing ML models is the website features in the dataset and availability of sufficient number of websites to create reliable predictive models. The Kaggle dataset that was considered in this study of phishing detection consists of 19 features and 1782 instances. The machine learning algorithms applied include logistic regression, decision trees, random forests, K-nearest neighbours and SVM along with the deep learning model of ANN. This paper has the following structure: Sect. 2 presents the literature survey and details of work done in this field. Section 3 describes the dataset and delves deep into the data pre-processing and feature engineering steps that were performed before applying different models. The Sect. 4 lists out the models that were employed to perform the classification and analyses the results of these models. The general flow of phishing as a classification process has been shown in Fig. 1.

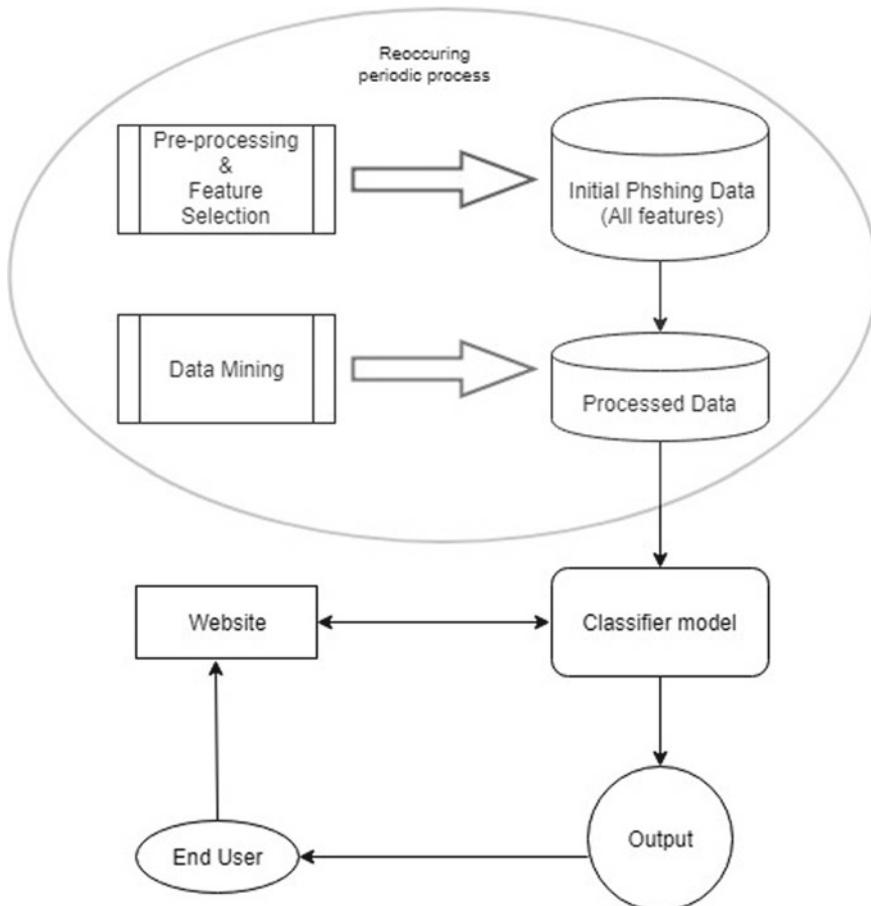


Fig. 1 Phishing as a classification process

2 Related Work

In this section, the existing work performed for detection and classification of phishing and the different methodologies used for this purpose are addressed. Phishing, which is being used for stealing of personal information and credentials, is also used against the banking system. So, it becomes crucial to prevent the phishing attack on banking system.

The work done in [4] implements a fuzzy hybrid system in order to detect phishing attack on Iranian e-banking. Here around 50 instances have been considered for detection, and after the feature engineering process, the system is able to detect a phishing attack with an accuracy of around 88%. The process of phishing attack occurs in the following manner: URL of a legitimate website is used by the attacker,

who replaces or adds a few characters to that URL so that a fake website is created which serves the attacker's purpose. Hence, URL of the website is one of the most important features used for classification of website as legitimate or not legitimate. In [5], the author lexically analyses the different types of phishing and legitimate URLs and finds the relation between the URL tokens. The classification accuracy using URL as the most important feature is about 97%. In [6], the comparison of many features was performed by the author using data mining algorithms. The various machine learning that are used are naive Bayes, regression tree, KNN and SVM. The results reveal the accuracy that could be attained using these algorithms for the lexical attributes.

Similarly, in [7] the dataset used contains thirty features and more than 11,000 examples in which each example is labelled as legitimate or phishy website. The data samples have been assembled from PhishTank and Millersmiles archives. The results of eight types of ML algorithms to detect fraudulent activities were estimated. From the eight different algorithms used the decision trees, Bayes net and SVM performed comparatively well, but the very large amount of information given by the decision trees and models of Bayes net and SVM are difficult to manage and comprehend by the users. In [8], the comparative study of accuracies of different machine learning algorithms is done on the different approaches. It considers different features for different algorithms. Some of those features include website traffic, text, URL, login form, certificate, etc. Because of the growing popularity of machine learning in every possible domain to solve the real life problem and to detect phishing, machine learning algorithms are used.

The paper [9] presents a machine learning algorithm which performs phishing detection using the lexical and domain attributes. In the methodology undertaken in [10], the URLs were classified involuntarily as either fraudulent or legit using supervised learning that used features which were lexical as well as host-based. The work done in [10] gives an approach to classify URLs automatically as either malicious or benign based on supervised learning on both lexical and host-based features. Also, the work done in [11] has used a data set that consists of 2889 malicious and benign emails for their study. The lowest error rate was obtained for random forests classifier as 43 attributes were employed for the training and testing of classifier models. The other algorithms employed in the comparative analysis included logistic regression (LR), classification and regression trees (CART), Bayesian additive regression trees (BART), support vector machines (SVM), random forests (RF) and neural networks (NNs). This shows that when appropriate data pre-processing and feature engineering is performed, acceptable results can be expected from the predictive models.

3 Methods and Dataset

(1) Dataset Description

This dataset was obtained from Kaggle website. It consists of 19 features and 1782 instances in the training example. The classification of websites as malicious or legit was performed using the dataset [12]. Table 1 presents a list of features and corresponding description.

(2) Data Preparation

Upon examination of the dataset, it was observed that the feature URL has all unique values (1781 unique values). The description of the dataset hinted at the fact that the

Table 1 Detailed feature description

Feature name	Feature description
URL	Identification of URL
URL_LENGTH	Instances of characters in URL
NUMBER_SPECIAL_CHARACTERS	Special characters identified in URL
CHARSET	Character encoding standard
SERVER	Server operative system given by packet response
CONTENT_LENGTH	Content size of HTTP header
WHOIS_COUNTRY	Name of the country attained from the server acknowledgement
WHOIS_STATEPRO	Name of the states attained from the server acknowledgement
WHOIS_REGDATE	Server registration date
WHOIS_UPDATED_DATE	The user from whom the last upgraded date was received by the server
TCP_CONVERSATION_EXCHANGE	The total packets that send and received to/from client and server
DIST_REMOTE_TCP_PORT	The total count of detected ports that are distinct to TCP
REMOTE_IPS	The total count of IP to honeypot connections
APP_BYTES	Amount of bytes transferred
SOURCE_APP_PACKETS	The packets received by the server from the honeypot
REMOTE_APP_PACKETS	The total packet count at the server reception
APP_PACKETS	The honeypot-server communication generated IP packet count
DNS_QUERY_TIMES:	The honeypot-server communication generated DNS packet count
TYPE	Category of web page denoted (1—malicious, 0—benign)

feature URL seems to be a sort of mapping key that does not describe the URL at all. Hence, the URL column was dropped for the training purpose. Regarding the null values, the following features were found to contain null values:

SERVER: 1 null value

CONTENT_LENGTH: 812 null values

DNS_QUERY: 1 null value

The CONTENT_LENGTH variable was imputed by interpolation. As it formed the major chunk of the data, it could not be dropped. Even after interpolation, the variable SERVER was found to have one null value; hence it was decided to drop the null value. The dataset contained categorical variables and hence they had to be processed before different machine learning algorithms could be applied to them. One-hot encoding is one of the best techniques of doing so as it has the benefit of weighing the value improperly. The method of one-hot encoding is supported by many libraries but we choose the ‘pandas.get_dummies()’ method to attain our goal. The data was made discrete for model creation after performing this step. The dataset was then separated into training and testing set (70–30%) and a random state value was assigned which can help anyone recreate our training/testing results.

(3) Feature Engineering

The process of choosing variable and modifying them while constructing a predictive model using machine learning is known as feature engineering, also called as data pre-processing. This Kaggle dataset consisted of 19 features, out of which some were used as is and some were transformed into something more useful.

The feature importance for all the features was observed in a descending order and it was observed that ‘SOURCE_APP_BYTES’ and ‘REMOTE_APP_PACKETS’ had the highest importance. But this was unusual since, the SERVER and origin of the place of server response should have been assigned greater importance. The unusual results could have been due to calculation of importance after converting the categorical variables into dummy variables. Hence, the feature importance was again calculated and then it was found that as expected the ‘SERVER’, ‘WHOIS_STATEPRO’, ‘WHOIS_COUNTRY’, ‘WHOIS_STATEPRO’, ‘WHOIS_REGDATE’, ‘WHOIS_UPDATED_DATE’ are among the features having highest importance. Figure 2 shows those features with the highest importance among all the features.

Hence, we decided to focus on these features. Regarding ‘SERVER’, there are a lot of types. That is why it was decided to reduce the dimensionality; an assumption will have to be made which turned out to be beneficial for the accuracy as it will be shown later. The assumption was that the server types with only one unique count will be assigned ‘RARE_VALUE’ for classification. For the registration date and updated date also, there were many unique values which increased the dimensionality. Therefore, it was decided to perform the extraction of year from these dates. All the different date formats were taken into account and accordingly the extraction was performed for both the features. As seen in Fig. 3, these steps considerably reduced the number of different values for those features.

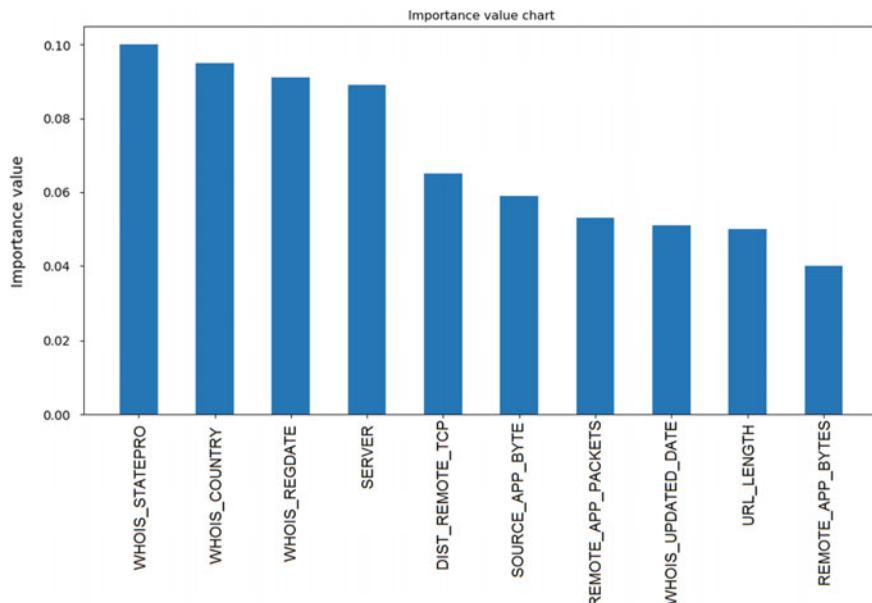


Fig. 2 Importance value of features

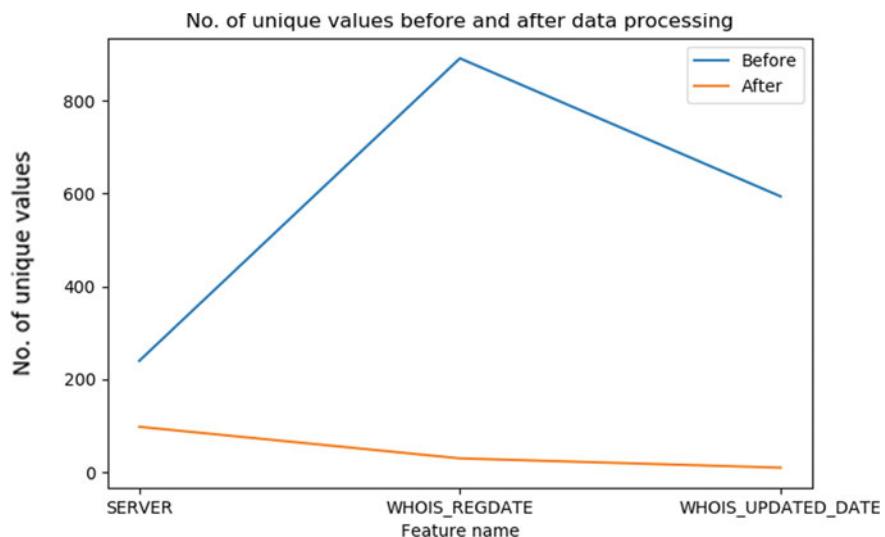


Fig. 3 Number of unique values before and after data processing

Similarly, having values for country without the state did not make much sense. Hence, the rows with no state values but having country values were filled with other state values of the same country.

4 Results and Experiments

Scikit-learn library was used to perform the experiments on this dataset. Different prediction models were deployed to classify the websites as malicious or benign. The Keras library was used to build the artificial neural network model. As shown in Fig. 4, deep component includes the feed forward network which consists of four hidden layers and one output layer. As shown in Table 3, it consisted of five hidden layers including the input layer. The number of dense layer units as well as the activation function used has been shown in Table 2.

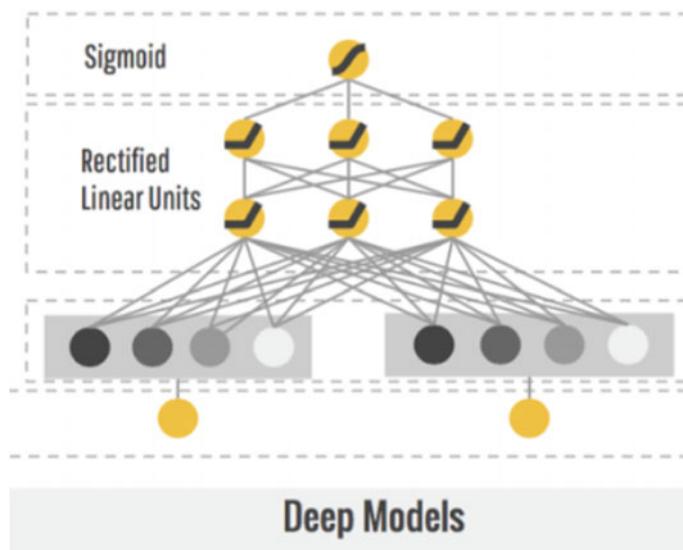


Fig. 4 Deep models

Table 2 ANN Layers

S. No.	Dense layer units	Activation function
1	15	Rectified linear unit
2	15	Rectified linear unit
3	15	Rectified linear unit
4	15	Rectified linear unit
5	1	Sigmoid

As shown in Fig. 5, the highest accuracy was obtained for decision tree model (96.82%) and random forests model too (95.70%). Other machine learning classification models such as logistic regression, K-nearest neighbours and SVM showed relatively low accuracy. The deep learning model of artificial neural network also showed good accuracy of 94.36%. The imputation strategy of mean was used in the ANN model and feature scaling was performed on the dataset. As the accuracy of such models is highly dependent on the number of training instances available, it could be improved by training the ANN for more number of epochs and a larger enough dataset. The various machine learning models and their respective performance metrics are shown in Table 3.

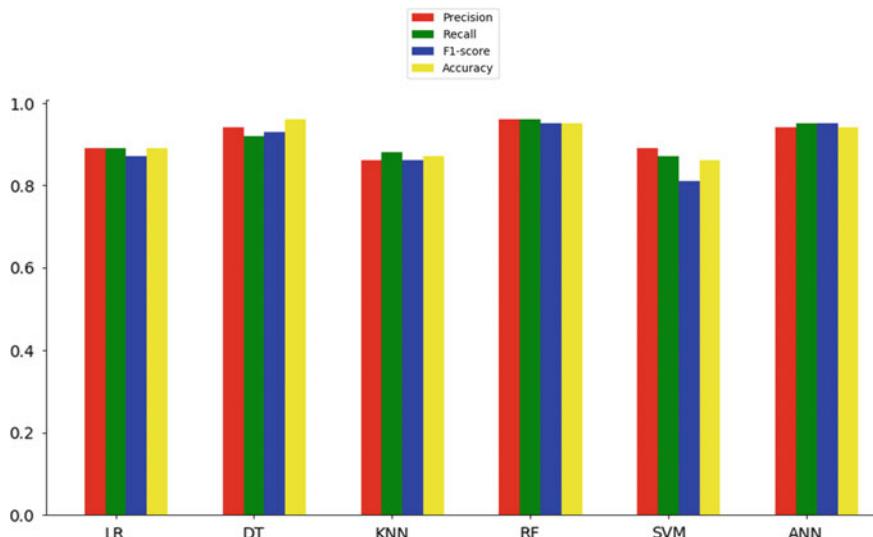


Fig. 5 Results of the experiment

Table 3 Comparative analysis of the classification model

S. No.	Classification model	Precision	Recall	F1-score	Accuracy (%)
1	Logistic regression	0.89	0.89	0.87	89.35
2	Decision tree	0.94	0.92	0.93	96.82
3	K-nearest neighbours	0.86	0.88	0.86	87.80
4	Random forests	0.96	0.96	0.95	95.70
5	SVM	0.89	0.87	0.81	86.92
6	Artificial neural network	0.94	0.95	0.95	94.36

5 Conclusion and Future Scope

The application of machine learning to cyber security problems has always given rise to robust solutions. A comprehensive analysis of different techniques and their comparison has been shown in the paper in detailed graphical form. The data pre-processing is a crucial part of any model building process and same has been achieved here. Deep learning model like ANN can also be employed to generate high prediction accuracy which will only increase as the number of training instances increase. Such studies will not only help the industry to avoid high losses but also helps in keeping intact the credibility of user information.

Further scope lies in exploring other factors in the phishing problems which are concerned with the individual user credentials and making the user aware of such activities. This will help us to take required contingency actions to prevent huge loss of data in the future.

References

1. Phishing statistics: retrieved from <https://www.comparitech.com/blog/vpn-privacy/phishing-statistics-facts/>
2. Phishing statistics: retrieved from <https://cofense.com/wp-content/uploads/2017/11/Enterprise-Phishing-Resiliency-and-Defense-Report-2017.pdf>
3. Phishing statistics: retrieved from <https://info.wombatsecurity.com/hubfs/2018%20State%20of%20the%20Phish/Wombat-StateofPhish2018.pdf?submissionGuid=2ecea77c-aa0d-404a-b0f4-030732e60a3a>
4. Montazer GA, ArabYarmohammadi S (2015) Detection of phishing attacks in Iranian e-banking using a fuzzy-rough hybrid system. *Appl Soft Comput* 35:482–492
5. Khonji M, Iraqi Y, Jones A (2011) Lexical URL analysis for discriminating phishing and legitimate websites. *CEAS*
6. James J, Sandhya L, Thomas C (2013) Detection of phishing URLs using Machine Learning techniques. In: 2013 international conference on control communication and computing (ICCC), pp 304–309
7. Abdelhamid N, Thabtah FA, Abdel-jaber H (2017) Phishing detection: a recent intelligent Machine Learning comparison based on models content and features. In: 2017 IEEE international conference on intelligence and security informatics (ISI), 72–77
8. Jain AK, Gupta BB (2016) Comparative analysis of features based Machine Learning approaches for phishing detection. In: 2016 3rd international conference on computing for sustainable global development (INDIACom), 2125–2130
9. Chu W, Zhu BB, Xue F, Guan X, Cai Z (2013) Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. In: 2013 IEEE international conference on communications (ICC), 1990–1994
10. Ma J, Saul LK, Savage S, Voelker GM (2009) Beyond blacklists: learning to detect malicious web sites from suspicious URLs. *KDD*
11. Abu-Nimeh S, Nappa D, Wang X, Nair S (2007) A comparison of Machine Learning techniques for phishing detection. *eCrime Researchers Summit*
12. Urcuqui C, Navarro A, Osorio JF, García M (2017) Machine learning classifiers to detect malicious websites. *SSN*

Chapter 44

Automation of Paper Setting and Identification of Difficulty Level of Questions and Question Papers



Ayesha Pathan and Pravin Futane

1 Introduction

A question paper can be called a good question paper if it possesses the following essential characteristics [1].

1. Validity
2. Reliability
3. Objectivity
4. Usability

Test making process is a very challenging task for test setter [2]. Therefore, computerized process of making question paper can reduce the time for setting question paper. But how to assess students' knowledge, traditionally examinations kept based on written examinations. But nowadays different assessment approaches like MCQ-based online assessments etc. being used. In all these assessments methods, question difficulty is very important part while creating question paper. Question difficulty helps to develop critical thinking and higher order cognitive abilities in students.

Benjamin bloom in 1948 created taxonomy, which is popularly known as Bloom's taxonomy, which is later revised by Lorin Anderson. There are different levels in Bloom's taxonomy. Difficulty gets increases from lower level to higher level as shown in Fig. 1. The verbs which different Bloom's taxonomy levels uses are

A. Pathan (✉) · P. Futane

Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India
e-mail: akp201996@gmail.com

P. Futane
e-mail: pravinfutane9@gmail.com

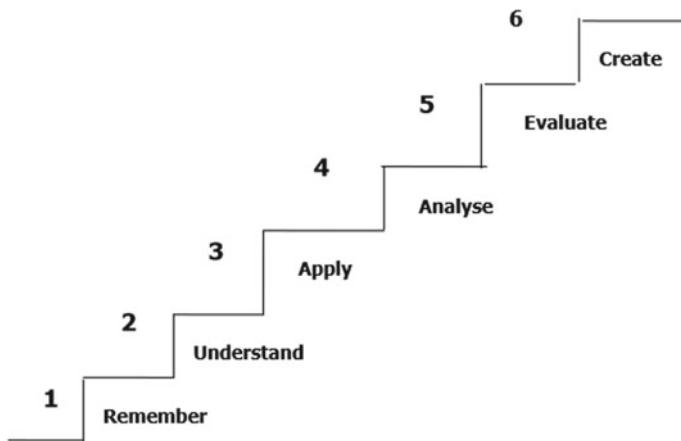


Fig. 1 Bloom's taxonomy levels [6]

- (1) Remember level align with keywords like Arrange, Describe, Name, Memorize, Order.
- (2) Understand level align with keywords like Explain, Summarize, Infer, Paraphrase, Discuss etc.
- (3) Apply level align with keywords like Apply, Choose, Modify, Discover, Diagram, Show etc.
- (4) Analyze level align with keywords like Break down, Calculate, Model, Subdivide etc.
- (5) Evaluate level align with keywords like Critique, Judge etc.
- (6) Create level align with keywords like Generate, Plan, Produce etc.

The proposed automated paper generation system set paper according to users need. User has to set schema i.e. format of question paper. For this study, the Savitribai Phule Pune University exam paper has been used. The proposed system divided into three parts. First is identification of individual question's difficulty. Using pre-processing techniques of natural language processing and by executing Latent Dirichlet Allocation, which is used to give the topics from sentence.

Second is automation of paper setting according to users need. And third is identification of difficulty level of question paper which is given in text file to system. For the third module also NLP pre-processing and LDA algorithm has been used.

2 Literature Survey

In this section, study 4 reference papers, to study different methods used for finding difficulty level of question paper and automation of paper setting.

A. Question Paper Generator System

Authors: Surbhi Choudhary, Abdul Rais Abdul Waheed, ShrutiKA Gawandi, Kavita Joshi Department of Computer Science and Engineering Theem College of Engineering, Boisar, University of Mumbai, Sep–Oct 2015 [3].

In this paper, an automatic question paper generator system is implemented. All the feature from login of admin to sending question paper through mail. Authors use the Bloom's taxonomy as to convert any question into proper question according to Bloom's taxonomy. For setting difficulty level of question author only decide questions difficulty, which is our try to remove this drawback in proposed system. For testing purpose, authors have used different sets of question in huge amount [3].

B. Enabling Fine Granularity of Difficulty Ranking Measure for Automatic Quiz Generation

Authors: Sasitorn Nuthong, Suntorn Witosurapot, Department of Computer Engineering, Prince of Songkla University Songkhla, Thailand [4].

This paper is about measuring difficulty of question for creation of automatic quizzes. Based on similarity index of answer options to question difficulty of quiz can be found. Based on ontology, generation of answer options is done. Author has used the five-level difficulty, which results in precise outcome of difficulty level of question. If the difficulty score of question is high, then it is considered as difficult question likewise categorization of difficulty level of question is done. The limitation of this system is that it gives difficulty level only for quiz type questions [4].

C. Automated Analysis of Exam Questions According to Blooms Taxonomy

Authors: Nazlia Omara, Syahidah Sufi Harisa, Rosilah Hassana, Haslina Arshada, Masura Rahmata, Noor Faridatul Ainun Zainala Rozli Zulkifli University Kebangsaan Malaysia 2011 [5].

This paper is based on Bloom's taxonomy method. Natural language processing (NLP) technique is used for knowing semantic meaning of question. Firstly, text file questions input is given to NLP module. In that stopwords removal is done then parts of speech tagging will happen, verbs, adverbs, etc. categorization will happen. If any verb or adverb matches with Bloom's categorization keyword, then question will be considered in corresponding Bloom's category. The rule development will happen, and categorization will be done. For rule development, 70 questions of programming subject are considered for training set. The drawback of this system is that difficulty level of question has not considered [5].

D. Question Difficulty How to Estimate Without Norming, How to Use for Automated Grading

Author: Ulrike Pado, Hochschule fur Technik Stuttgart Schellingstr. 24 70174 Stuttgart, Germany [6]

This paper uses the Bloom's taxonomy as well as the answer given by students for a question; these methods are used for finding difficulty level of question. Depending upon difference between student write the answer and get marks and actual difficulty

of question the questions difficulty gets find. For this work, author has used German corpus dataset, which contains 31 question-answer set.

Hence, from literature survey above discussed it reveals that Bloom's taxonomy method along with natural language processing is best method for automatic question generation system [6].

3 Proposed Methodology

In this study, automation of paper setting using natural language processing and identifying of difficulty level of question paper is done.

A. *Question paper generation*

The question paper generation will reduce the time spent by teachers on making question paper. And this generated question paper's quality is high since question setting can be done with standardized format.

Identifying individual questions difficulty level: firstly, user will log in system with mail id and password. Then user will enter any question for which user wants to find difficulty of question. After entering question, natural language processing module will start.

- (A) First stopwords removal takes place.
- (B) Then tokenization of words will happen.
- (C) Stemming will be done on question.
- (D) Lemmatization will be done.

After pre-processing Latent Dirichlet Allocation algorithm will start. Latent Dirichlet Allocation (LDA) is used for getting topics from document or sentences.

After getting the topics from sentences, system will categorize the questions according to Bloom's level mapping in easy, medium or hard level. There are total six Bloom's levels. First two lower levels i.e. remember and understand are mapped with easy level then next two levels i.e. apply and analyze are mapped with medium and last evaluate and create are mapped with create. The system will identify with the topics get and Bloom's keyword and identify the category of that question.

Sometimes there is conflict condition occur when question contains more than two categories topics/words; in that case, system will consider the highest level of Bloom's as question category and mapped with easy/medium or high accordingly.

Creating database: this is the important module of the system. In this, first user will add questions of computer engineering courses subjects as this work is limited only to Savitribai Phule Pune University's BE computer engineering course. Admin will login using credentials. Then he/she will enter the question. Then select the year from first year/second year/third year or last year. Then select unit ranging from 1 to 6. Then enter the subject name of that question and will store it in MySQL database. The question gets categorized in either easy, medium or hard automatically. For testing purpose, in this study 300 questions have been stored.

Generating question paper according to user's schema: this module will work if system has already store database. Firstly, admin will login with credentials. Then select from FE, SE, TE or BE for which he/she wants to generate the paper and having already stored questions in database. Then select subject from already stored database. Then user can select how question's difficulty should be there for each question. In this system, the format for FE and SE and for TE and BE kept same. In FE/SE, there are 16 question from which 8 are kept compulsory, and for TE/BE there are 18 questions from which 9 are kept compulsory. Question paper's difficulty setting is totally based on user how he/she select question.

After setting the schema when user enters generate question paper, the question will get retrieved from stored database according to year, subject, unit and difficulty level given by user. For that session, no question will get repeated since two separate columns are maintained. One from already stored data and another from retrieved data. The retrieved data gets deleted every iteration of paper generation.

B. Identifying difficulty level of question paper

This is the second part of the system. If user wants to know any random question paper's difficulty, then proposed system (Fig. 2) will identify it. For this work, Savitribai Phule Pune University's computer engineering question papers has been used. User will import the text document of question paper from system. Then pre-processing is done on each question.

After pre-processing, Latent Dirichlet Allocation algorithm will start. Latent Dirichlet Allocation (LDA) is run to get topics from documents and it will be compared with Bloom's keywords. After getting the topics from sentences, system will categorize the questions according to Bloom's level mapping in easy, medium or hard level. There are total six Bloom's levels. First two lower levels i.e. remember and understand are mapped with easy level then next two levels i.e. apply and analyze are mapped with medium and last evaluate and create are mapped with hard . The system will identify with the topics get and Bloom's keyword and identify the category of that question.

For the proposed system, we considered if there are 16 or 18 questions then

- (1) if 60% of 16 or 18 questions are easy, 30% of 16 or 18 questions are medium and 10% of 16 or 18 questions are hard then level of difficulty of question paper is considered as easy level.
- (2) if 40% of 16 or 18 questions are easy, 40% of 16 or 18 questions are medium and 20% of 16 or 18 questions are hard then level of difficulty of question paper considered as medium level.
- (3) if 30% of 16 or 18 questions are easy, 40% of 16 or 18 questions are medium and 30% of 16 or 18 questions are hard level then level of difficulty of question paper considered as hard level.

C. Architecture

After giving exact schema, the paper setting generation module will start and it gives the generated paper output as shown in Fig. 5.

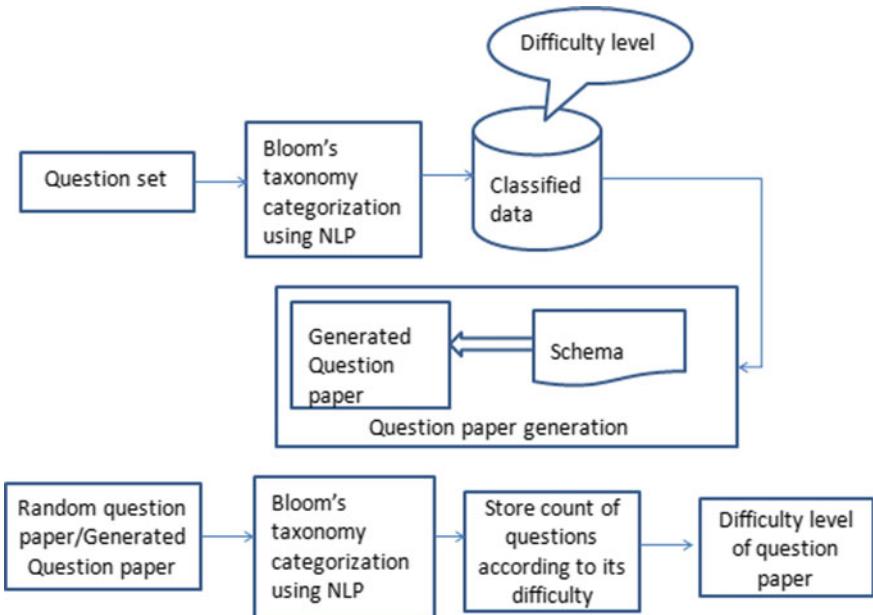


Fig. 2 Proposed system

Lastly one additional thing also user can do if he/she takes any question paper and give to this system then proposed system will find the overall difficulty level of question paper using NLP and Bloom's taxonomy as shown in Fig. 6.

D. Algorithms

- (1) Bloom's taxonomy categorization:
 - (a) Remove stopwords from the question given.
 - (b) Apply POS tagging to the output of the Step a.
 - (c) if (Word == NOUN or VERB or ADVERB) Go to step d; else if (Word == End of String) Go to step f; else Go to Step c;
 - (d) Compare the word with keywords of blooms taxonomy levels.
 - (e) Append the Bloom's taxonomy level to the result String. Go to Step c;
 - (f) Print result String [7].
- (2) Latent Dirichlet Allocation:
 - (a) First step in algorithm is pre-processing of text. This includes the following processing:
Tokenization: In this, each sentence is tokenized or cut into single word. All the stopwords are removed and tokenized words lowercased.
Lemmatization: In this, all the words in past participle form convert into simple present form.

Stemming: In this, words are converted to their original form.

- (b) Converting text to bag of words.
- (c) Running Latent Dirichlet Algorithm: The LDA algorithm is topic modelling algorithm. It finds each topic in document to work on it.

E. Mathematical Model

Let S be the whole system, which consists: $S = \{IP, Process, OP, IP\}$ is the input of the system.

- A. Process is the process applied to the system to process the given input.
- B. OP is the output of the system.

(a) **Input:**

$$IP = \{REG, LOG, LD, QUE\}$$

where,

1. REG is registration of user.
2. LOG is login of user into the system.
3. LD is load classified data of question in database.
4. QUE is Question set which is stored in database.

(b) **Process:**

$$PRO = \{CLAS, SCH, BT, PRE\}$$

1. CLAS is classification of question based on easy, medium and hard. CD is classified data using decision tree algorithm will store all classified questions.
2. SCH is schema for generating question paper.
3. BT is Bloom's taxonomy for categorize the questions into easy, hard and medium.
4. PRE is predict overall question difficulty.

(c) **Output:** $OP = AQS$

AQS—Automatic question paper setting with identification of its difficulty level.

(F) Comparative study

This system is more focused on outcome-based education process, which uses Bloom's taxonomy measure to categorize questions. Current state-of-the-art systems mainly focus on automation of paper setting rather than quality of question. This system uses the natural language processing techniques, LDA algorithm to categorize questions, which current systems have not used. This system improves quality of question papers produced in examinations. This system will also give the overall difficulty level of question paper if user gives any randomly generated paper to the system. This can be said as a novel feature of this system.

4 Result and Discussions

A. Experimental Setup

1. Simulator: Eclipse
2. Language: JAVA
3. Operating System: Window 7 OS or above version

B. Data Used

For working project, Savitribai Phule Pune University's computer engineering courses question papers are taken.

The system will classify the output based on difficulty level of question and according to it paper setting for the exam can be done. Firstly, the module is giving the output as difficulty level of question. As shown in Fig. 3 the output displayed. For instance, the example entered was to describe the different box specification in classroom engineering and the output obtained difficulty level of this question is easy. Since the mapping of question is with Bloom's taxonomy's first level i.e.

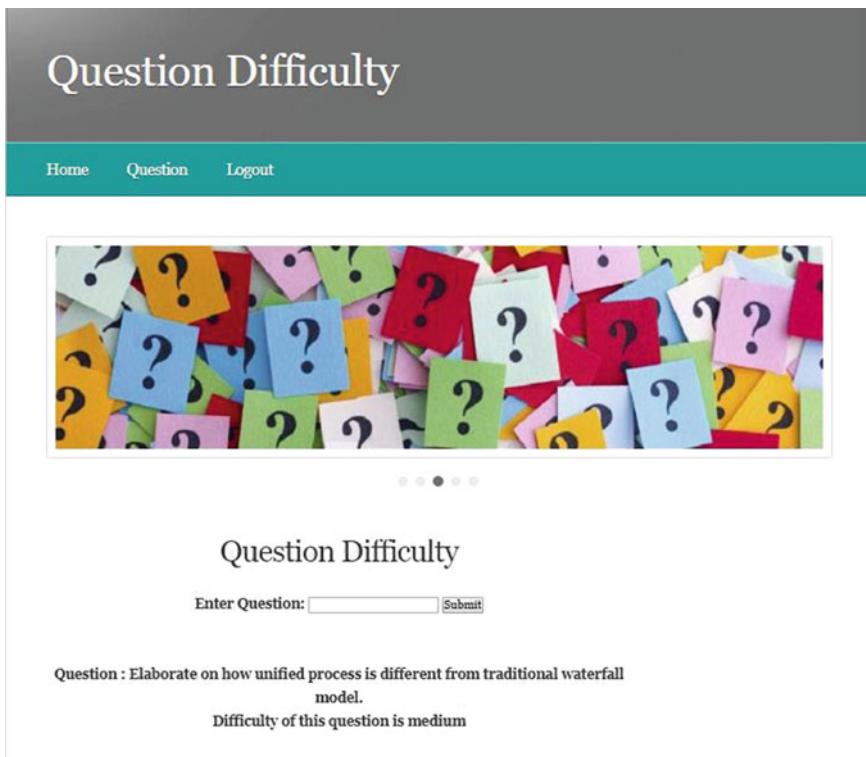


Fig. 3 Finding individual questions difficulty level

The screenshot shows a web-based application titled "Question Difficulty". The top navigation bar includes links for "Home", "Add Questions", and "Logout". Below the navigation is a decorative banner featuring a collage of colorful sticky notes with the text "Ask us a question". The main content area is titled "Admin Home". It contains a form with the following fields:

- Enter Question :** A text input field containing the question: "Draw a class diagram for online shopping system. Assume the scope."
- Unit :** A dropdown menu set to "1".
- Year :** A dropdown menu set to "TE".
- Subject :** A dropdown menu set to "Software modeling and design".
- Submit**: A button to submit the form.

Fig. 4 Creation of database

remember. Further the database creation of classified data will be done (Fig. 4). After getting the classified data, the implementation will go towards the automation of paper generation. For sample purpose, the second year engineering subject has been taken and overall generated papers difficulty has been set to easy. The generated paper is shown in Fig. 5. The comparative study of different approaches has been done and from that Bloom's taxonomy is best method can be adopted for finding difficulty of question (Fig. 6). For testing purpose, the 500 questions have been stored in database and categorized in different level of difficulty (Fig. 7). By using the automation generation system, the lengthy process of making question paper can be done in few minutes.

For verifying this work, the comparison between the output given by proposed system and subject matter expert is done. And system almost giving the correct answer same as given by subject matter expert. The comparison is given in Fig. 7. Also, the accuracy of results is shown in Fig. 8.

Subject : OSD

Time : Two Hours

Maximum Marks : 70

Instructions to the candidates :-

- (i) Answer to the questions (Q. NO. 1 or Q. NO. 2, Q. NO. 3 or Q. NO. 4, Q. NO. 5 or Q. NO. 6, Q. NO. 7 or Q. NO. 8,)
- (ii) Assume suitable data, if necessary.
- (iii) Draw neat labeled diagram wherever necessary.
- (iv) Figures to right indicate full marks.

Q1a) Explain the race condition in assigning Modes. (8)

Q1b) Why is the principle of locality crucial to the use of virtual memory? Explain with example (8)

Q1c) Explain in short - BIOS, MBR and init() process. (4)

OR

Q2a) What is disk Mode? State the difference between disk Mode and in-core inode. (4)

Q2b) Explain Kernel Structure. With neat diagram. (8)

Q2c) Explain with neat diagram process states and transition. (8)

Q3a) Give the details of U-area field. (8)

Q3b) What is deadlock? Explain necessary conditions to occur the deadlock? (8)

OR

Q4a) Explain the term signal and elaborate the various circumstances under which signals of the various classes are used. (8)

Q4b) Write short notes on: i) Tunis System. ii) Shared memory. (8)

Q5a) Write a short note on: i) Mork Manager. ii) Shim Manager. (8)

Q5b) Explain with example Linux utilities - grep, egrep, fgrep and sort. (8)

OR

Q6a) What is make utility? Explain it with example. Consider your own makefile. (8)

Q6b) Explain in detail how to make USB bootable with any open source tool/ utility? (8)

Q7a) Write short notes on: Fail soft operation. (9)

Q7b) Write a short note on: i) Multiprocessor scheduling. ii) Real time scheduling. iii) Linux scheduling. (9)

OR

Q8a) Write a short note on: i) Palm OS. ii) Google Android. iii) Windows Mobile. (9)

Fig. 5 Generated question paper



Fig. 6 Difficulty level of question paper: medium

Sr. No	Difficulty level	Difficulty analysis by expert	Difficulty analysis by system
1	Easy	300	295
2	Medium	150	144
3	Hard	50	46

Fig. 7 Experimental result

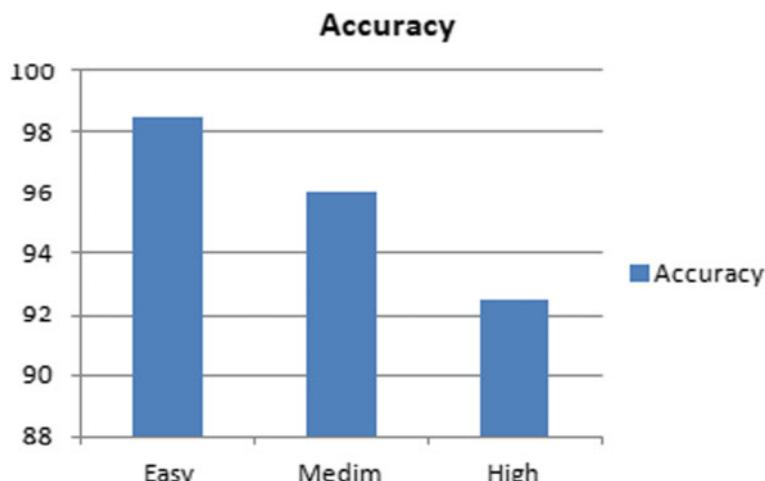


Fig. 8 Accuracy

5 Conclusion

This study is conducted with the aim of improving process of setting question paper according to Bloom's taxonomy which is a categorization method for differentiating question according to its complexity. The design of question difficulty based on

Bloom's taxonomy has been studied and analysis of different approaches has been studied. The experimental results show that Bloom's taxonomy enhances the systems effectiveness. The results compared with subject expert and proposed system show that its good approach to adapt in colleges or schools. By this approach, overall 94% accuracy got. There is need to standardize the process of question paper setting which is done by proposed system. Automatic question generation system has been studied. The automation of paper setting with knowing difficulty level of question will improve the assessment of students while giving exam.

References

1. <http://www.Gpnagpur.Ac.In/GpnagpurNew/PDF/2018/Question>
2. Naik K et al (2014) Automatic question paper generation system using randomization algorithm. Int J Eng Tech Res (IJETR) 2(12):192–194
3. Choudhary, Surbhi et al (2015) Question paper generator system. Int J Comput Sci Trends Technol 3(5)
4. Nuthong S, Witosurapot S (2014) Enabling fine granularity of difficulty ranking measure for automatic quiz generation. In: 2017 9th international conference on information technology and electrical engineering (ICITEE). Sci World J 2014
5. Omar N et al (2012) Automated analysis of exam questions according to Bloom's taxonomy. Procedia-Soc Behav Sci 59:297–303
6. Pad U (2017) Question difficulty how to estimate without norming, how to use for automated grading. In: Proceedings of the 12th workshop on innovative use of NLP for building educational applications
7. Krishna G, Sai et al (2017) Automatic question paper generator system with blooms taxonomy categorization. Int J Eng Technol Comput Res 5(2)

Author Index

A

Arora, Sakshi, 215
Asarkar, Shruti V., 181

B

Bapat, Shriya, 427
Bahel, Vedant, 149
Bhagat, Chaitanya, 361
Bhagwat, Laxmi B., 15
Bhirud, Bhagyashri, 329
Birajdar, Udayan, 245
Bobde, Sarika, 419

C

Chaudhari, Sheetal, 283
Chavan, Sumit, 437
Chikodikar, Shreyas, 245
Chiwhane, Shwetambari, 245
Chopade, Rupali, 99, 109
Choudhary, Brijesh, 33, 237
Chougule, Shreyanka B., 169

D

Dadhich, Shubham, 245
Dani, Ankit, 33, 237
Das, Ratan, 57
Deshmukh, Pratiksha R., 411
Deshmukh, Saurabh, 295
Deshmukh, Sudhir, 369
Dhavale, Sunita, 369
Dhavale, Sunita V., 49
Doke, Niket, 319
Dorle, Avanti, 437

E

Ekbote, Onkar, 127

F

Futane, Pravin, 447

G

Gadhave, Sanket, 245
Golhar, Archana, 99
Gorave, Asmita, 203
Gupta, Ayushi, 41, 139
Gutte, Aditya, 33, 237

H

Hooda, Neha, 57

I

Inamdar, Aditya, 437
Inamdar, Vandana, 127

J

Jadhav, Shivajirao M., 227
Jadhav, Sneha, 403
Jagtap, Vandana S., 33, 343
Jambhali, Ketaki, 427
Janvir, Sakshi, 99
Javale, Deepali, 381
Jayawardena, Chandimal, 265
Joshi, Deepali, 319
Joshi, Shivani, 427

K

- Kadam, Poonam, 23
 Kadam, Vinod J., 227
 Kale, Rohini, 119, 157
 Kanade, Vijay A., 91
 Karanje, Rucha, 403
 Kaur, Hardeep, 81
 Keskar, Devyani, 41, 139
 Khandekar, Varsha S., 67
 Khedkar, Sujata, 391
 Kodag, Vikas, 255
 Kokitkar, Nikita, 255
 Kulkarni, Juily, 427
 Kulkarni, Parth, 255
 Kulkarni, Pradnya V., 119
 Kulkarni, Siddhivinayak, 437
 Kumar, Anil, 81
 Kumari, Priya, 309
 Kumar, Neelesh, 57
 Kumbhare, Rohit, 109
 Kurdukar, Atharv A., 227

L

- Ladole, Kshitij, 203
 Lambor, Shilpa, 403

M

- Mane, Deepak, 361
 Md. Tanvir Uddin Haider, 309
 Misra, Srinibas, 203

N

- Nagtilak, Saraswati, 157
 Nair, Aswin Ramachandran, 353
 Nimbalkar, Shivali, 109

P

- Pachghare, V. K., 99, 109, 329
 Padir, Omkar, 203
 Pahade, Aatish, 381
 Palwe, Sushila, 41, 139
 Paranjape, Shweta, 3
 Pathan, Ayesha, 447
 Patil, Anirudha, 203
 Patil, Balaji M., 15
 Patil, Mohit, 381
 Patil, Shriya, 195

Patra, Sudhansu

- , 353
 Pendsey, Eshani, 3
 Pendsey, Shreya, 3
 Peshkar, Atharva, 149
 Phalnikar, Rashmi, 411, 419
 Phatak, Madhura V., 181
 Pophale, Chinmay, 33, 237
 Potdar, Akshay, 381
 Punjabi, Nikhil, 283

R

- Rai, Sunil, 119, 157
 Raj Mohan, M., 353
 Rodrigues, Elton, 23
 Roge, Swapnil, 23
 Rupani, Kunal, 283

S

- Sankhe, Pratik, 23
 Saykhedkar, Harshad, 255
 Sen, Sachin, 265
 Shamdasani, Mohnish, 283
 Shete, Shambhavi, 295
 Shinde, Subhash, 391
 Singal, Rushikesh, 381
 Singh, Sugandha, 149
 Sonawani, S. S., 237
 Srinath, Pravin, 67
 Sudha, K. L., 169

T

- Takalikar, Mukta, 255
 Thorat, Pranav, 343
 Tongaonkar, Raajas, 343

V

- Varsha, H. S., 169
 Verma, Divya, 403
 Vighnesam, N. V., 169

W

- Wani, Insha Majeed, 215
 Warhade, Krishna K., 195
 Watpade, Anuja, 255
 Wu, Xin-Wen, 437