

Rajiv Misra · Nishtha Kesswani ·
Muttukrishnan Rajarajan ·
Veeravalli Bharadwaj ·
Ashok Patel *Editors*

Internet of Things and Connected Technologies

Conference Proceedings on
5th International Conference
on Internet of Things and Connected
Technologies (ICIoTCT), 2020

Advances in Intelligent Systems and Computing

Volume 1382

Series Editor

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

Advisory Editors

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

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

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

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

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


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

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

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

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

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

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

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

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

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

Indexed by DBLP, EI Compendex, INSPEC, WTI Frankfurt eG, zbMATH, Japanese Science and Technology Agency (JST).

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

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

Rajiv Misra · Nishtha Kesswani ·
Muttukrishnan Rajarajan ·
Veeravalli Bharadwaj · Ashok Patel
Editors

Internet of Things and Connected Technologies

Conference Proceedings on 5th International
Conference on Internet of Things
and Connected Technologies (ICIoTCT), 2020

 Springer

Editors

Rajiv Misra
Department of Computer Science
and Engineering
Indian Institute of Technology Patna
Patna, Bihar, India

Muttukrishnan Rajarajan
City University of London
London, UK

Ashok Patel
Department of Computer Science
Florida Polytechnic University
Lakeland, FL, USA

Nishtha Kesswani
Department of Computer Science
Central University of Rajasthan
Rajasthan, India

Veeravalli Bharadwaj
Department of Electrical and Computer
Engineering
National University Singapore
Singapore, Singapore

ISSN 2194-5357

ISSN 2194-5365 (electronic)

Advances in Intelligent Systems and Computing

ISBN 978-3-030-76735-8

ISBN 978-3-030-76736-5 (eBook)

<https://doi.org/10.1007/978-3-030-76736-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2021

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

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

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

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book presents the proceedings of the 5th International Conference on Internet of Things and Connected Technologies (ICIoTCT), held during July 3–5, 2020, at Indian Institute of Technology (IIT), Patna, India. The Internet of Things (IoT) promises to usher in a revolutionary, fully interconnected “smart” world, with relationships between objects and their environment and objects and people becoming more tightly intertwined. The prospect of the Internet of Things as a ubiquitous array of devices bound to the Internet could fundamentally change how people think about what it means to be “online.”

The ICIoTCT 2020 conference provided a platform to discuss advances in Internet of Things (IoT) and connected technologies, such as various protocols and standards. It also offered participants the opportunity to interact with experts through keynote talks, paper presentations and discussions, and as such stimulated research. With the recent adoption of a variety of enabling wireless communication technologies, like RFID tags, BLE, ZigBee, embedded sensor and actuator nodes, and various protocols such as CoAP, MQTT and DNS, IoT has moved on from its infancy. Today smart sensors can collaborate directly with machines to automate decision making or to control a task without human involvement. Further, smart technologies, including green electronics, green radios, fuzzy neural approaches and intelligent signal processing techniques, play an important role in the development of the wearable healthcare devices.

August 2020

Rajiv Misra
Nishtha Kesswani
Muttukrishnan Rajarajan
Veeravalli Bharadwaj
Ashok Patel

Organization

Program Chairs

Rajiv Misra
Nishtha Kesswani

IIT Patna
Central University of Rajasthan, India

Program Committee

Aakanksha Sharaff
Nishtha Kesswani
Rajiv Misra

National Institute of Technology Raipur, India
Central University of Rajasthan, India
IIT Patna

Contents

IoT Based Solar Smart Tackle Free AGVs for Industry 4.0	1
Subhranil Das, P. Arvind, Sourav Chakraborty, Rashmi Kumari, and S. Deepak Kumar	
A Hybrid Positive-Unlabeled Learning Method for Malware Variants Detection	8
Alle Giridhar Reddy, Pushkar Kishore, Swadhin Kumar Barisal, and Durga Prasad Mohapatra	
Air Quality Monitoring and Disease Prediction Using IoT and Machine Learning	18
Mahima Jayaraj	
A Mobile Based Market Information System	33
Adebayo Abayomi-Alli, Sanjay Misra, Mojisola Dada, Christian Yetunde Alonge, John Bosco Agbaegbu, Oluwasefunmi ‘Tale Arogundade, and Ravin Ahuja	
Reduction of Sidelobe Levels in OFDM Radar Signal Using Two Samples Sliding Window Adder (TSSWA) Algorithm	45
M. P. Raghu Srivatsa and C. G. Raghavendra	
A Review and Case Study on Attacking and Security Tools at Application-Layer of IoT	51
Ankit Sinha, Sachin Kumar, Preeti Mishra, Umang Garg, and Arpit Agwarwal	
Accident Prevention of Automobile Using Real-Time Tracking System	64
S. B. Rudraswamy, M. G. Pruthvi, Sameera Fatima, Sneha Jangamashetti, and M. Sathya	

Attention LSTM for Time Series Forecasting of Financial Time Series Data	74
Yedhu Shali, Banalaxmi Brahma, Rajesh Wadhvani, and Manasi Gyanchandani	
Approximating Communication Cost for NFV-Enabled Multicasting . . .	85
Yashwant Singh Patel, Shivangi Kirti, and Rajiv Misra	
Delay Analysis for P2P Systems Using LPWAN	96
Shivendu Mishra and Rajiv Misra	
Data Mining Techniques in IoT Knowledge Discovery: A Survey	105
Beza Mamo Rabdo and Asrat Mulatu Beyene	
Literature Review on Answer Processing in Community Question Answering System	121
Saman Qureshi and Sri. Khetwat Saritha	
Shallow over Deep Neural Networks: A Empirical Analysis for Human Emotion Classification Using Audio Data	134
Chandresh S. Kanani, Karanjit Singh Gill, Sourajit Behera, Anurag Choubey, Rohit Kumar Gupta, and Rajiv Misra	
<i>k</i> Stacked Bidirectional LSTM for Resource Usage Prediction in Cloud Data Centers	147
Yashwant Singh Patel, Rishabh Jaiswal, Savyasachi Pandey, and Rajiv Misra	
Offline Bengali Handwritten Sentence Recognition Using BiLSTM and CTC Networks	158
M. A. Muhaimin Sakib, Omar Sharif, and Mohammed Moshuiul Hoque	
Robotics Vehicle System Using Simple Hand Gesture	169
Sanjay Misra, Modupe Odusami, Olusola Abayomi-Alli, Olaoluwa Oseni, Robertas Damasevicius, and Ravin Ahuja	
An Intelligent Recommendation System Based on Collaborative Filtering and Grid Structure	182
Animesh Chandra Roy and Mohammad Shamsul Arefin	
SMOTE Based Weighted Kernel Extreme Learning Machine for Imbalanced Classification Problems	193
Roshani Choudhary and Sanyam Shukla	
Bluetooth 5 and Docker Container: Together We Can Move a Step Forward Towards IOT	201
Mohammad Equebal Hussain and Rashid Hussain	
Statistical Analysis Based Feature Selection for Detection of Breast Cancer Using Thermograms	211
Vartika Mishra, Monika Lilhare, and Santanu Kumar Rath	

Distributed Computing Solution for Steganography Using Visual Cryptography and Genetic Algorithm 225
 Harsh Maru, P. Pranav, Yashwanth Miryala, and Bhawana Rudra

Wearable Device Design for Cattle Behavior Classification Using IoT and Machine Learning 235
 Fatema Ahmed, Bholanath Roy, and Saritha Khetawat

Fuzzy Edge Image Steganography Using Hybrid LSB Method 249
 Debina Laishram, Themrichon Tuithung, and Tayenjam Jeneeta

Universum Based Nonparallel Kernelized Extreme Learning Machine for Binary Classification 259
 Akansha Mangal and Sanyam Shukla

A New Image Watermarking Scheme Based on Block Conversion and DWT-SVD Approach 267
 Ranjeet Kumar Singh

A Concept of E-Medical Record Storage and Sharing Based on Blockchain Technology 278
 Adla Sanober and Shamama Anwar

Comparative Analysis of 6T, 9T and 10T SRAM Cells Using 18 nm FinFet Technology 287
 Aksh Rastogi and V. K. Tomar

Hybrid Cloud: The Next Generation of EAI 297
 Manish Kumar Dash, Supriya Devidutta, Bhabendu Kumar Mohanta, and Debasish Jena

Comparative Analysis of High Speed, Low Power FIR Multirate Polyphase Filter 306
 Rajendra Rewatkar and S. L. Badjate

Simplified and Secure Session Key Sharing for Internet of Things (IoT) Networks 319
 Krishan Pal Singh, Shubham Tomar, Usha Jain, and Muzzammil Hussain

Machine Learning Based Network Slicing and Resource Allocation for Electric Vehicles (EVs) 333
 Rohit Kumar Gupta, Anurag Choubey, Shlok Jain, R. R. Greeshma, and Rajiv Misra

A Peak Bulk Deal Tracing Based Comparative Analytical Study for Optimize Investment Strategy 348
 Amit Suthar, Hiral Patel, and Satyen M. Parikh

Path Planning Algorithms for Different Scenarios 354
 Saurabh Singh and Namita Tiwari

IPL: From Lens of Data Science	366
Rahul Pradhan, Drashti Maheshwari, Mayank Aggarwal, Ankur Chaturvedi, and Dilip Kumar Sharma	
Outlier Detection in Wireless Sensor Networks with Denoising Auto-Encoder	379
Bhanu Chander and Kumaravelan	
Malicious Webpage Classification	389
Kushagra Krishna, Jaytrilok Choudhary, and Dhirendra Pratap Singh	
Ontology-Based Modeling of Cloud Application Using Security Patterns	400
Ashish Kumar Dwivedi, Shashank Mouli Satapathy, and Aakanksha Sharaff	
Deep-Learning Based Mobile-Traffic Forecasting for Resource Utilization in 5G Network Slicing	410
Rohit Kumar Gupta, Amit Ranjan, Md Ashraf Moid, and Rajiv Misra	
Software Defined Radio Based Multi-band Audio Broadcasting System for Drone Based Communications	425
Yaswanth Chalamalasetti, Sudhir Kumar Sahoo, Barathram Ramkumar, and M. Sabarimalai Manikandan	
Genetic Algorithm Based Feature Selection for Software Reliability Prediction Using Multi-layer Perceptron	436
Priyanka Kumari, Kulamala Vinod Kumar, and Durga Prasad Mohapatra	
Fault Prediction Using Deep Neural Network	447
Avishikta Chatterjee, Kulamala Vinod Kumar, and Durga Prasad Mohapatra	
Head Pose Classification Based on Deep Convolution Networks	458
Sadia Afroze and Mohammed Moshiul Hoque	
A Robust Multi-Server Two Factor Remote User Authentication Scheme Using Smartphone and Biometric	470
Hasan Muhammad Kafi, Md. Al-Hasan, Mohammad Hasan, and Md Mamunur Rashid	
Packet Error Probability Model for IEEE 802.15.6 MAC Protocol in Wireless Body Area Network	481
M. Ambigavathi and D. Sridharan	
Semantic Meaning Based Bengali Web Text Categorization Using Deep Convolutional and Recurrent Neural Networks (DCRNNs)	494
Md. Rajib Hossain and Mohammed Moshiul Hoque	

Stability and Power Analysis of Read Decoupled 8T SRAM Cell 506
Saloni Bansal and V. K. Tomar

**Analysis of Higher Stable 9T SRAM Cell for Ultra Low
Power Devices 517**
Harekrishna Kumar and V. K. Tomar

Author Index 527



IoT Based Solar Smart Tackle Free AGVs for Industry 4.0

Subhranil Das¹(✉), P. Arvind¹, Sourav Chakraborty¹, Rashmi Kumari²,
and S. Deepak Kumar³

¹ Department of Electrical and Electronics Engineering, Birla Institute of Technology Mesra, Ranchi, India

² Department of Computer Science and Engineering, Birla Institute of Technology Mesra, Ranchi, India

³ Department of Production Engineering, Birla Institute of Technology Mesra, Ranchi, India
deepakkumar@bitmesra.ac.in

Abstract. The emergence of Industry 4.0 has made a breakthrough by providing state of the art services when it comes to the manufacturing and material handling sectors. Continuous research and development are being made to ascertain efficiency. In spite of technological advancements, several bottlenecks still exist that require its mitigation to a great extent. Hence, in this paper, a strategy has been suggested which encompasses the fully automated Autonomous Guided Vehicles (AGV), Green energy, Automatic Storage and Retrieval Systems (ASRS) and the Internet of Things (IoT) to coordinate the status of the operation by acquiring data through a supervisory control in order to optimize vehicular paths with the help of a dynamic routing algorithm. Based on the operating area, the ZigBee protocol is seen to be best suited for this purpose. Thus, by formulating a prescribed working environment, manufacturers can conserve energy, reduce costs, eliminate machine downtime, and increase operational efficiency.

Keywords: AGV · ASRS · Industry 4.0 · IoT · Path routing · Solar photovoltaics · ZigBee

1 Introduction

A primitive man's simple life has given way to a modern man's more complex life. Wants have increased a hundredfold and have become more diversified in kind, thereby leading to automation [5]. Various operations are necessary which requires repetition and manifold material handling with high precision and accuracy, this has led to the development of automatic computer numerically controlled machines which can achieve the required tasks in shorter time period, hence opening more areas of research in the field of automatic machines which will lead to higher productivity and the ability of machines to complete more complex tasks [6]. Thus, the introduction of robotic manipulators in industries, such as AGVs, ASRS are widely used for tasks involving positioning, logistics etc. Parallel robots are lesser utilized in industries because of their lower popularity.

Extensive research has been carried out in the last 15 years on six degree of freedom robots which are generally called Parallels. These manipulators are very efficient in moving heavy masses at higher speeds [7]. Thus, for a number of such manipulators, a need to coordinate correct pathways comes into play so as to ensure that no two manipulators interfere with each other's operation. Besides that, the fully automated vehicles should have the ability to drive safe and smooth in traffics. Path tracking, steering, obstacle avoidance and traffic laws must be considered in the driving goals [8–10]. Also, it includes positioning of vehicle or equipment onboard the relative vehicle to other objects. The docking stage was implemented by introducing a vital target of the current mobile robot position and the final target location [11]. Thus in this paper, an improved approach to path tracing for automated guided vehicles have been proposed that takes into account the position velocity and the presence of other vehicles in proximity along with the size of the obstacle. The objectives of the paper revolves around the implementation of a coordinated vehicular movement with the avoidance of hurdles in the path, elucidation of use in storage and retrieval purpose, the implementation of self-docking for battery charging without the dip in performance of the entire industrial functioning and lastly the aggregation of solar energy-based power supply to encourage a green and clean powering of material handling.

2 Automated Guided Vehicles (AGV)

Automated guided vehicles (AGV) are kind of transportable robots which follow a certain kind of marked lines to follow the path defined by them. AGV are mainly used in flexible management systems (FMS) for movement of materials from one place to another place. For path optimization, a number of different works has been implemented. One of the works proposed by Wenrong Lu et.al. which is mainly focused on order picking that has being applied in warehousing operations. A dynamic path routing algorithm has been implemented with multiple pick up points inside the factory settings [1].

Hybrid genetic algorithm has been implemented in AGVs in order to make integrated scheduling and detailed routing paths. The fitness function which is of multi-objective has been used for adaptive weight updates so that the weights can be assigned to each objective function for every generation [2].

A new approach has been made that mainly focused on integration of coordination between machine and scheduling problem of automated guided vehicle (AGV). A genetic algorithm in combination with Dijkstra algorithm has been proposed in order to solve the problems such as required number of minimum numbers of AGVS, minimum transportation time and conflict-free routing problem (CFRB) in flexible manufacturing systems (FMS) [3, 4].

Role of Autonomous Guided Vehicles (AGV) in Industry 4.0 has made sustainable development in reducing the cost of applicability and enhancing flexibility of the system. Regarding the safety standards, AGVs turn out to be a prominent figure head in enhancing the productivity consequently results in profitability. The most significant role of AGV in industry is to increase number of required additional AGVs as to evade from high investment costs.

However, in many industries reported a significant number of problems such as fixed path following path which has become a significant problem. This problem is still

persisting from last decade and exact solution has not been available yet. For creating fully efficient vehicles with minimum investment cost, two solutions are being proposed to make fully automated vehicles.

3 Problem Description

In this paper, Flexible Manufacturing Services (FMS) has been considered where more than one AGV are being implemented here. At a particular decision point, there are about 8 different paths are being applied. These AGVs has been well operated through a single algorithm known as S P tackle free algorithm which can be applied for multiple operations or tasks assigned to these vehicles. The most interesting part of AGVs is, these vehicles are identical to each other. In starting point, multiple operations can easily be done with multiple AGVs so that there will be no collisions of AGVs. In addition to that, following assumptions are being made:

- Each AGV has been assigned a particular work for a particular time
- There are standby AGVs if any fails to work
- Single operation at a particular path for selected AGV
- AGVs are numbered on a particular task assigned on their priority basis
- AGVS are considered to have different velocities
- The path followed by AGVs may be linear as well as nonlinear curves

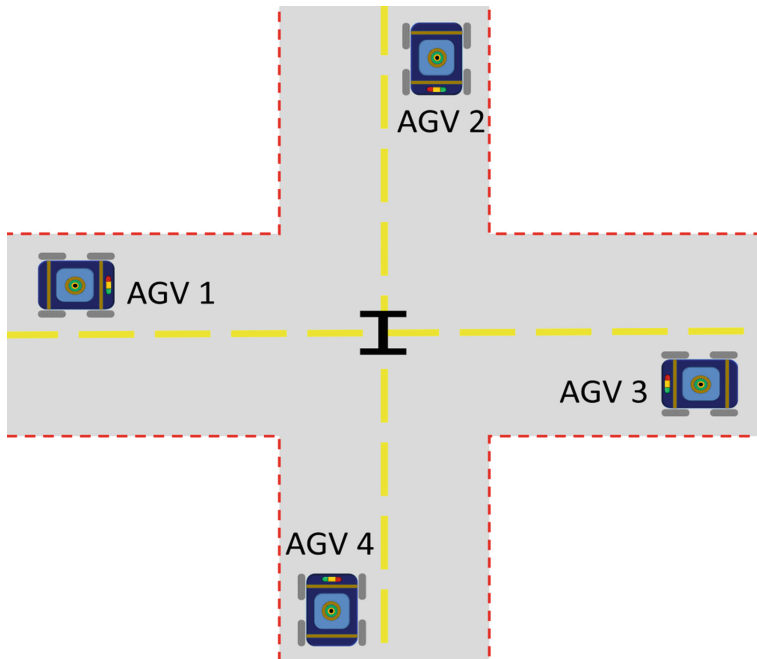


Fig. 1. Pictorial representation of 4 -point Decision of AGV

Here, both classical algorithm and optimization are applied to every AGV in order to get minimum time for any particular task so that the efficiency of the work can be enhanced and wastage of power can be minimized.

3.1 Algorithm for 4-point Decision for Free Collision of AGVs

Let V_1, V_2, V_3, V_4 symbols represents are vehicle 1, vehicle 2, vehicle 3 and vehicle 4 respectively (Fig. 1).

Let $I =$ intersecting point of all the vehicles.

The path followed by the vehicle is of bidirectional where the materials are transported from one end to other end. The yellow lines indicate the path of the vehicle traversed whereas red lines indicate the minimum number of sensors are to be used.

STEP1: Considering the vehicle V_1 , check whether the distance of V_2 to decision point (I), V_4 to decision point(I).

STEP2: If $V_2I=V_1I=V_4I$

then the input controller of V_1 is increased w.r.t. V_2 and V_4

Endif

STEP3: Considering the vehicle V_2 , check whether the distance of V_1 to decision point (I), V_3 to decision point(I).

STEP4: If $V_3I=V_2I=V_1I$

then the input controller of V_2 is increased w.r.t. V_1 and V_3

Endif

STEP5: Considering the vehicle V_3 , check whether the distance of V_4 to decision point (I), V_2 to decision point.

Step6: If $V_3I=V_1I=V_2I$

then the input controller of V_3 is increased w.r.t. V_2 and V_4

Endif

STEP7: Considering the vehicle V_4 , check whether the distance of V_1 to decision point(I), V_3 to decision point.

Step8: If $V_4I=V_1I=V_3I$

then the input controller of V_4 is increased w.r.t. V_1 and V_3

Endif

Step9: We will be considering all the steps involved from step 1 to step 8 into a single step by considering all the positions of all vehicles V_1, V_2, V_3 and V_4

4 State Space Modeling of Automatic Guided Vehicles

The states of 4 AGVs are considered to be time dependent. The initial states and predicted states of vehicles are $X_{t|0}, P_{t|0}$ at time $t = 0$. The previous state and previous predicted states are given by $X_{t-1|0}, P_{t-1|0}$.

The generalized state space equation in terms of velocity is represented by

$$\dot{x}_{pn} = A x_{pn} + B a_{xn} + N \quad (1)$$

where X = State matrix for i, j dimensions

a_{xn} = Control Variable Matrix

N = State noise matrix

n = No of vehicles ranging from 1, 2, 3, 4,, N - 1.

Since, there is no interference in the relative positions of vehicle. As a result of this, N = 0.

The measurement of state that determines the position of the vehicle which is given by the equation

$$Y = C X_{pn} \quad (2)$$

Here, some notations are to be made before proceeding for solution for above two equations,

ΔT = Time for the relative positions of the vehicle for an interval of 1 s

X_0 = initial state of the vehicle

\dot{X} = velocity of the vehicle

\ddot{X} = acceleration of the vehicle.

Therefore, the state matrix is represented by position, velocity and acceleration variables with respect to time given by following equation

$$X = X_0 + \dot{X}t + \frac{1}{2}\ddot{X}t^2 \quad (3)$$

Let x_{px} = position of vehicle in x direction

x_{py} = position of the vehicle in y direction

x_{pz} = position of the vehicle in z direction

\dot{x}_{px} = velocity of the vehicle in x direction

\dot{x}_{py} = velocity of vehicle in y direction

\dot{x}_{pz} = velocity of vehicle in z direction

a_{x1} = acceleration of the vehicle in x direction

a_{y1} = acceleration of the vehicle in y direction

a_{z1} = acceleration of the vehicle in z direction

For AGV1,

$$\begin{bmatrix} \dot{x}_{px1} \\ \dot{x}_{py1} \\ \dot{x}_{pz1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{px1} \\ x_{py1} \\ x_{pz1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \end{bmatrix} \begin{bmatrix} a_{x1} \\ a_{y1} \\ a_{z1} \end{bmatrix} \quad (4)$$

For AGV2,

$$\begin{bmatrix} \dot{x}_{px2} \\ \dot{x}_{py2} \\ \dot{x}_{pz2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{px2} \\ x_{py2} \\ x_{pz2} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \end{bmatrix} \begin{bmatrix} a_{x2} \\ a_{y2} \\ a_{z2} \end{bmatrix} \quad (5)$$

For AGV3,

$$\begin{bmatrix} \dot{x}_{px3} \\ \dot{x}_{py3} \\ \dot{x}_{pz3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{px3} \\ x_{py3} \\ x_{pz3} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \end{bmatrix} \begin{bmatrix} a_{x3} \\ a_{y3} \\ a_{z3} \end{bmatrix} \quad (6)$$

For AGV4,

$$\begin{bmatrix} \dot{x}_{px4} \\ \dot{x}_{py4} \\ \dot{x}_{pz4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{px4} \\ x_{py4} \\ x_{pz4} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \end{bmatrix} \begin{bmatrix} a_{x4} \\ a_{y4} \\ a_{z4} \end{bmatrix} \quad (7)$$

5 Integration with IOT

Internet of things (IOT) plays a crucial role in integrating the AGV vehicle to enhance the communication between them. IOT interfaces the object to the cyberspace which implies that each and every AGVs has the virtual identifiable entity of itself in cyberspace. A personal area network employing ZigBee comprises of a router, routing protocols and entities or objects to be interfaced. The underlying objective to employ ZigBee protocols relies on the fact that its data transfer rate requirement is low thereby making it energy efficient. In order to ensure the smooth functioning of the system the assemblage should be vendor independent. To ensure the real time control the embedded processor should be able to clock an appreciable speed such that the entire entities data can be collected for a short processing span, which implies that any change in data should be successfully captured by the system in a nutshell, the system have the capability to operate in real time, beside the compatibility data security is also a key factor hence ZigBee comes with robust AES (advanced encryption standard) for data encryption and security. The data is analyzed and transfer to the cloud from the production control. The intelligent gateway harmonizes the communication among various data sources. Moreover, this interface can be applied in both directions [12].

6 Conclusion

Although the significant developments have taken place in the fields of manufacturing areas, but there are several areas where certain issues such as data security, cost of implementing the systems still exists. The agglomeration of routing algorithm of AGV and IOT will lend a helping hand to resolve some of the potential areas of improvement not only in the manufacturing systems but also in the different areas of applications such as warehousing operations.

References

1. Lu, W., McFarlane, D., Giannikas, V., Zhang, Q.: An algorithm for dynamic order-picking in warehouse operations. *Eur. J. Oper. Res.* **248**(1), 107–122 (2016)
2. Umar, U.A., Ariffin, M.K.A., Ismail, N., Tang, S.H.: Hybrid multi-objective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. *Int. J. Adv. Manufact. Technol.* **81**(9–12), 2123–2141 (2015)
3. Corr ea, A.I., Langevin, A., Rousseau, L.M.: Scheduling and routing of automated guided vehicles: a hybrid approach. *Comput. Oper. Res.* **34**(6), 1688–1707 (2007)
4. Lyu, X., Song, Y., He, C., Lei, Q., Guo, W.: Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems. *IEEE Access* **7**, 74909–74924 (2019). <https://doi.org/10.1109/ACCESS.2019.2919109>
5. Critical Manufacturing Blog about MES, Industry 4.0. <https://criticalmanufacturing.com/en/newsroom/blog/posts/blog/labor-management-made-better>. Accessed 02 Oct 2019
6. Lim, D., Park, C., Kim, N.H., Kim, S.-H., Yu, Y.S.: Fall-detection algorithm using 3-axis acceleration: combination with simple threshold and hidden Markov model. *J. Appl. Math.* **2014**, 1–8 (2014). <https://doi.org/10.1155/2014/896030>
7. Tsai, L.W.: *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley, Hoboken (1999)
8. Diankov, R.: *Automated construction of robotic manipulation programs* (2010)
9. Antonelli, G., Arrichiello, F., Chiaverini, S.: The null-space-based behavioral control for autonomous robotic systems. *Intel. Serv. Robot.* **1**(1), 27–39 (2008)
10. Connell, J.H.: A behavior-based arm controller. *IEEE Trans. Robot. Autom.* **5**(6), 784–791 (1989)
11. Bostelman, R., Bostelman, R., Hong, T.: *Review of Research for Docking Automatic Guided Vehicles and Mobile Robots*. US Department of Commerce, National Institute of Standards and Technology (2016)
12. Keertikumar, M., Shubham, M., Banakar, R.M.: Evolution of IoT in smart vehicles: an overview. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 804–809. IEEE, October 2015



A Hybrid Positive-Unlabeled Learning Method for Malware Variants Detection

Alle Giridhar Reddy^(✉), Pushkar Kishore, Swadhin Kumar Barisal,
and Durga Prasad Mohapatra

Department of Computer Science, National Institute of Technology Rourkela,
Rourkela, Odisha, India
{518CS1002,durga}@nitrkl.ac.in

Abstract. Malware are capable of evolving into different variants and conceal existing detection techniques, which relinquishes the ineffectiveness of traditional signature-based detectors. There are many advanced malware detection techniques based on machine learning and deep learning, but they cannot fulfill the real issues in industries. Malware variants are evolving at a rapid pace and labelling each of them is not practical and feasible. So, industries are considering a lot of the unlabeled samples as benign, while only a few are labelled. Consequently, the authentic malware samples are mislabelled. Bias created by mislabelling the samples severely restrains the accuracy. Also, the user is unsatisfied with malware detection system, since there is poor negotiation between the speed and accuracy.

In this research article, we propose a hybrid positive-unlabeled learning technique for malware detection that can address some important challenges. Here, we use an ensemble model comprising of Logistic regression (cost-sensitive boosted), Random Forest and Support vector machine, to detect the malware variants. Along with that, we demonstrate that features in the form of a triplet vector are optimal while training a model. Experimental outcomes show that our proposed model attains 91% malware detection accuracy having a false alarm rate less than 0.005, while the earlier state-of-art approaches can only achieve 76.4% to 89% accuracy. The detection speed of our approach is 0.003 s.

Keywords: Ensemble model · Positive-unlabeled learning · Machine learning techniques · Malware

1 Introduction

Malware is an extensive threat that covers computers as well as Internet of Things (IOT) devices [1]. Whenever software under study has coding or configuration error and wrongly sensed as anomalous, it generates a false positive data [7, 11, 14]. For the machine learning (ML) or deep learning (DL) based techniques to work, we must label several legitimate and benign binary executables for training. Labeling each sample is inimical as it takes a lot of time and requires

high labour costs. For this reason, the volume of the unlabeled sample is smaller compared to labeled one. For tackling this issue, companies label the remaining unlabeled malware samples as benign.

Now, the unlabeled binary executable is considered as benign one (negative samples), but the authentic malicious binary executable (positive samples) are mislabeled. Immense amount of malicious samples in the unlabeled binary executable will reduce the efficiency of the malware detection model. So, it generates bias in the decision boundary and is illustrated in Fig. 1, which leads to an inaccurate malware detector. We term this problem as positive-unlabeled learning, where a 2-class classifier is trained using a dataset containing negative data (benign), positive data (malicious) and unlabeled data (presumed as benign).

In this research work, our primary goal is to enhance the accuracy of the detector when it is trained using the dataset created from positive-unlabeled binary executable. Besides, detecting malware variants, we use system calls which is collected using NITRSCT [2] tool and demonstrate that the vector of consecutive 3 system calls as a feature, will be effective and optimal for malware detection. Whenever a dataset is created, we optimize the logistic regression with cost-sensitive boosting. We propose an ensemble model comprising optimized logistic regression, random forest and support vector machine, which will accurately detect malware variants and will have least detection time.

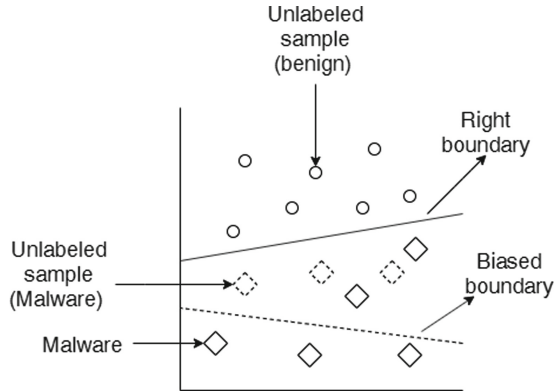


Fig. 1. Biased boundary misclassifies the samples [20]

We organize the remaining part of this article as follows. Section 2 consists of related works, Sect. 3 presents our proposed malware detector’s methodology. Section 4 exhibits the experimental results. Section 5 presents the comparison with related work. Section 6 discusses over the threats to validity and Sect. 7 concludes the paper with future directions.

2 Related Works

In this section, we discuss some existing work linked to our approach.

2.1 Malware Variants Detection

Fan et al. [3] recommended subgraphs construction of API calls which describes the prevalent behaviour of binary executable malware of the same family. But, the extraction of API calls from binary executable fails sometimes. Zhang et al. [4] embedded opcode and API sequential calls using Convolutional neural networks (CNN) and Back Propagation Neural Networks (BPNN) and the mix of these networks is used for training a malware variant detector having hybrid features. Zhang et al. [5] used topological features of the opcode graph to identify android binary executable malware. Stringhini et al. [8] created a graph depicting the file delivery networks and trained using semi-supervised Bayesian label propagation. Here, reputation of the acknowledged files is sent to all nodes of the graph. But these files are behind-time, meaning that whenever malicious binary executable is sensed, then up to that time, countless copies of the malware has already done the damage.

Canzanese et al. [6] represented binary executable using system call n-gram and used SVM for malware variants detection. Raff et al. [19] used CNN and bytecode n-grams for malware variants detection. Presence of noise in bytecodes is much higher compared with opcodes, resulting in a depreciation of accuracy. Kang et al. [18] used Naive Bayes technique towards catching the 2-opcode vectors of binary malicious executable. But, the assumption that Naive Bayes considers features as independent, depreciates the malware detector's accuracy. Puerta et al. [10] used SVM for detecting malware and represented binary executable using opcode frequencies. Simplicity of features depreciates the accuracy due to the lack of the adequate information in the features.

All the above discussed approaches rely on dataset having known positive and negative labels. If they will design detector with dataset having semi-labeled instances, then the accuracy is acutely hampered.

2.2 Positive-Unlabeled Learning

Some scholars preferred positive-unlabeled learning techniques for training the malware detector whenever they encounter positive-unlabeled data. Liu et al. [12, 13] designed a SVM with some bias and upcoming steps were used to find obvious positive samples. Xiao et al. [16] employed K-means to discover positive as well as negative instances in dataset which are unlabeled. But, these techniques were influenced by the recognized negative training samples. Malware detection results' accuracy will be disastrous if the negative data is inaccurately detected. Elkan et al. [17] trained a malware detection classifier on positive as well as unlabeled instances for estimating weights of validation set samples and modeled a weighted Support Vector Machine. Xu et al. [9] generalized solitary positive class of binary executable into various positive classes. Gong et al. [15]

identified the margin among positive and potentially negative samples and proposed a label-calibrated SVM. However, their method is confined to SVM and is inappropriate for other ML methods. Zhang et al. [20] suggested a cost-profound boosting technique for positive-unlabeled learning for malware detection. But they have used opcode representation, which is only the static analysis of the binary executable.

3 Proposed Methodology

In this section, we propose the methodology for detecting malicious variants using system calls and ML. First, we discuss the architecture of our proposed model. Second, we discuss the representation of system calls and conclude by presenting the positive unlabeled learning, which we use in our approach.

3.1 Architecture of Our Approach

We present the architecture of our malware detection model in Fig. 2. First, a model is initialized and trained using a customized cross entropy loss function and termed as optimized Logistic Regression. We use the cross-entropy function as it will output the binary label in the terms of probability. Then the training is done using the optimised LR, Support Vector Machine and Random Forest. After training is completed by the three different approaches, we ensemble them and use the hard voting classifier to predict the final label on the test dataset. In Ensemble model first, we will sum the predictions made by each model and then we will predict the class label with the most votes.

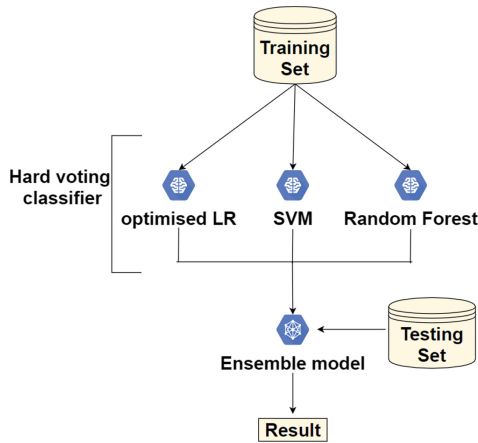


Fig. 2. Proposed architecture of our approach

3.2 Representation of System Calls

Features are extracted using n-gram model representing an adjacent sequence of n consecutive system calls derived from a selected sequential system call. We analyze three ML models’ true positive rate (tpr), by varying the length of the vector of system calls. The tpr of each detector is shown by fixing the false positive rate at 10^{-5} . For $n = 1$, the highest tpr is 0.32, while for $n = 2$, the highest tpr is 0.85. When we select $n = 3$, the highest tpr is 0.91, and it decreases to 0.81 whenever $n = 4$ is considered. So, we can infer that taking the value of $n = 3$ will be optimal for the length of all feature’s vector. Here, n is the number of consecutive system calls in a vector.

3.3 Positive-Unlabeled Learning

Nowadays, industries mark a limited portion of the binary malicious executable as positive sample and leftovers are “unlabeled legitimate data” containing malware variants. Upon training malware detector with “semi-labeled data” (unlabeled samples along with malware samples) using ML, the detection accuracy will be inadequate. In this article, we propose an ensemble method comprising of logistic regression (cost-sensitive boosted), random forest and SVM, that can boost the accuracy of positive unlabeled learning.

First, we optimize the logistic regression with cost-sensitive boosting. The idea behind this technique is assigning separate weights to unlabeled binary executable for differentiating between true legitimate binary executable and false legitimate binary executable (malware variants) in it. We calculate loss function using Eq. 1, where $Loss_{CEL}$ is the Cross Entropy Loss, $h(\mathbf{x})$ is the confidence of \mathbf{x} (model’s yield) and y_i is the label of a sample in the t^{th} iteration.

$$Loss_{CEL} = -\left(\sum_{i=0}^n \frac{h(x_i)}{\sum_{j=0}^{i-1} h(x_j)} \cdot (y_i \cdot \log(h(x_i)) + (1 - y_i) \cdot \log(1 - h(x_i)))\right) \quad (1)$$

The major problem is allocating proper weights to the samples. While training during each iteration, we allocate the weights bestowing to the confidence $h(\mathbf{x})$. Let \mathbf{x} be an instance of a dataset of binary executable, \mathbf{x}_m is the malware instance and \mathbf{x}_b is the benign one. Since \mathbf{x}_m in unlabeled datasets and malware datasets are analogous, $h(\mathbf{x}_m)$ hovers between 0 and 1, while $h(\mathbf{x}_b)$ of x_b in unlabeled data sets will advance towards 1. Accordingly, $h(\mathbf{x})$ of benign will be huge than $h(\mathbf{x})$ of malicious binary executables. This amplifies the value of genuine legitimate binary executable and dampens the cost of false legitimate binary executable, while training with unlabeled data. The algorithm for our proposed hybrid positive-unlabeled learning malware variants detector is presented in Algorithm 1.

4 Experimental Results

We present the experiment for demonstrating that our proposed ensemble model can boost the accuracy with commendable detection time. At first, we present

Algorithm 1: Hybrid positive-unlabeled learning malware variants detection model

Input: A data set \mathbf{X}
Output: A detection model which classifies samples into malware and benign.

- 1 Initialize a model $ML(x)$;
- 2 Set $avg = 0$ as initialization ;
- 3 **Function Training**($\mathbf{X}, ML(x)$):
- 4 **for** $i = 0 : n$ **do**
- 5 Train x using $ML(x)$;
- 6 calculate $h(x)$ from $ML(x)$;
- 7 Get $avg = (avg \cdot i + h(x)) / (i + 1)$;
- 8 Update $ML(x)$ according to Equation 1 ;
- 9 Apply Random Forest and SVM on \mathbf{X} ;
- 10 $ML(x) =$ Ensemble $ML(x)$ (Optimized Logistic regression), Random forest and SVM using hard voting ;
- 11 **return** $ML(x)$;
- 12 **end function**

the experimental setup, the dataset, the hyper-parameter settings and at last we conclude with performance evaluations.

4.1 Setup, Dataset and Hyper-parameters

We carry out entire experiments on one system. The variant of the CPU is Intel i5-3470 @ 3.20 GHz, the RAM is 16.0 GB and the OS is Windows 10. We have implemented our approach using Python language in which the matrix computations are dependent on numpy.

The dataset¹ considered for performance evaluation and training is developed using NITRSCT², which was developed by us. We evaluate the performance of our proposed model during training with the unlabeled datasets, which contain 20% malware variants and 80% benign ones.

For preparing a dataset, we have collected benign executables from 20 hosts in offices, laboratories and isolated testbeds. The malware used for experimental purpose is collected from VirusTotal³. The hyper-parameters which are fixed by us have a considerable effect on the performance. We display the hyper-parameters used in our way in Table 1.

4.2 Performance Analysis of Malware Detection

The parameters which we use for performance analysis of our proposed model are classification accuracy, detection false positive rate, detection true negative

¹ <https://github.com/pushkarkishore/NITRSCT/blob/master/data1.rar>.

² <https://github.com/pushkarkishore/NITRSCT/blob/master/Debug.rar>.

³ <https://www.virustotal.com/>.

Table 1. The hyper-parameter settings of our experiment

Detector	Hyper-parameter	Value
Optimized Logistic Regression (OLR)	Learning rate	0.001
Optimized Logistic Regression (OLR)	Number of iterations	5000
Random Forest (RF)	Number of estimators	100
Random Forest (RF)	Random State	0
Random Forest (RF)	Number of Jobs	2
Support Vector Machine (SVM)	Kernel	Sigmoid
Proposed model	Weight	SVM(1), OLR(1), RF(1)

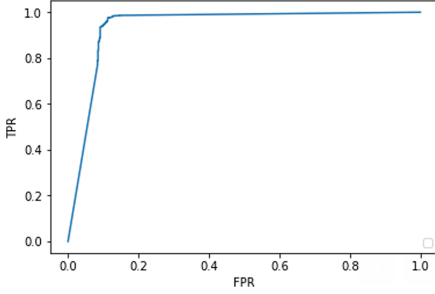
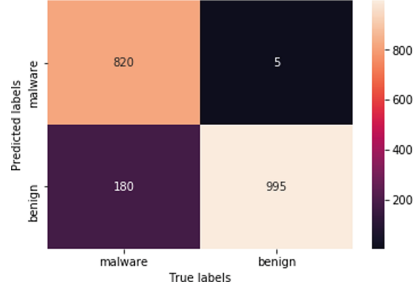
rate, detection false negative rate, detection precision, detection recall, F1-score, training time cost, detection time cost and area under curve. The classification accuracy is evaluated using Eq. 2. The recall of the model is the true positive rate, where True Positive (TP) is the number of correctly classified malware samples and False Negative (FN) implies malware samples misclassified as the benign one. TNR is the true negative rate, where False Positive (FP) is the number of benign samples misclassified as malware binaries and True Negative (TN) is the number of benign samples which are correctly classified. FPR represents false positive rate, FNR represents false negative rate, Precision represents malware detector’s precision and F1-score is computed using Precision and Recall.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (2)$$

The performance evaluation of our model is presented in Table 2. The ROC curve of our model is shown in Fig. 3. The area under the receiver operating characteristics (ROC) curve of our proposed model is 0.94. The confusion matrix of our model is presented in Fig. 4. For the combination of 1000 benign and malicious testing instances, we get 995 True negatives, 5 False positives, 180 False negatives, 820 True positives. Our model’s False positive rate is 0.005 which is considerably low as compared to other works. A malware detector is practically useful when it has low FPR.

5 Comparison with Related Work

We have compared the performance of our model with that of several state-of-art methods and shown in Table 3. By comparing with the other state-of-the-art methods, we observe that our approach significantly improves the classification accuracy, precision, the (1-FPR), F1-score and training time while retaining the detection speed. Accuracy is better than all the considered models, making it useful for industrial malware detection. Precision is 99.3%, which is higher than the precision evaluated by using the SVM method. Recall is 82.0%, which is lower than logistic (optimization) and CNN (optimization) methods. It means that 82.0% of the total relevant results are correctly classified by our proposed

**Fig. 3.** Roc curve of our model**Fig. 4.** Confusion matrix of our model**Table 2.** Performance evaluation of our model

Sl. no.	Performance parameters	Value
1	Accuracy (%)	91.0
2	Recall (%)	82.0
3	TNR (%)	99.5
4	FPR (%)	0.5
5	FNR (%)	18.0
6	Precision (%)	99.3
7	F1-score (%)	89.6
8	Detection Time (s)	0.003
9	Training Time (s)	160

model. Considering the problem under consideration, we give the highest priority to either precision or recall. In general, we use a simple metric, F1-score, which is the harmonic mean of precision and recall. Our model proves its vitality when we consider F1-score, which is 89.6%. Specificity is equivalent to “1-FPR”, which implies that benign samples being labeled benign is 99.5%. Its lower value will only block the benign process, so we consider it as an auxiliary parameter. Our extraneous objective of blocking of benign executable on the hosts will be minimal as specificity is higher. SVM [18] method has a higher precision, but it has lower F1-score, accuracy and recall. As accuracy and F1-score is lower than our proposed model, we cannot use this model. CNN (optimization) [20] method has higher recall than our proposed model, but lags in F1-score, accuracy and precision. Logistic (optimization) [20] method has also higher recall than our model, but has least accuracy, precision, specificity, F1-score, detection time as well as training time. Considering the above models, there is a poor trade-off between precision and recall, thus, their F1-score is minimal. Comparing with the above stated parameters, we observe that our proposed model is more suitable for malware detection.

Table 3. Comparison of performance of our approach with existing state-of-art approaches

Method	Accuracy	Precision	Recall	1-FPR	F1-score	Detection time	Training time
Logistic (optimization) [20]	89.0%	87.3%	91.2%	86.7%	89.2%	0.007 s	16,053.0 s
Softmax (optimization) [20]	80.0%	82.7%	75.8%	84.1%	79.1%	0.006 s	16,241.0 s
CNN (optimization) [20]	86.0%	86.2%	85.7%	86.3%	85.9%	0.053 s	107,519.0 s
CNN [19]	84.3%	91.8%	75.2%	93.3%	82.7%	0.053 s	109,633.0 s
SVM [18]	76.4%	92.9%	58.9%	95.5%	67.6%	0.006 s	676.0 s
Our approach	91.0%	99.3%	82.0%	99.5%	89.6%	0.003 s	160.0 s

6 Threats to Validity

For all the methods to work with system calls, they have to capture the system calls during runtime with the help of sandboxes, which makes it tough and costly in terms of resources in preparing dataset. Modern malware variants tend to hide their malicious behaviour whenever they detect themselves running in sandboxes. Our model can eliminate this issue to some level. We have analyzed every act of malware inactiveness or sleepy behaviour by making sandbox dynamically changing its time settings to deceive malware and stimulate its execution.

7 Conclusion and Future Work

In this paper, we proposed an ensemble malware detection method for positive-unlabeled learning adopted to detect numerous malware variants. In industries, malware detection model is trained with positive-unlabeled datasets, which severely limits the accuracy. Our approach addresses this issue by providing a novel ensemble malware variants detection model. Besides, we have demonstrated that a vector of three consecutive system calls, when considered as a feature in dataset will be optimal for malware detectors. The experimental results convey that our model achieves 91.0% accuracy with false alarm rate less than 0.005, while the other techniques achieve up to a maximum of 89%, when the unlabeled dataset contain many “mislabelled” data (positive data).

In the future, we can improve the cost of dataset creation and can detect newer sandbox-evading malware by enhancing the features of the sandboxes. We can consider a static analysis of the binary executable like API calls, opcode, etc. and design an ensemble detector to improve malware detection accuracy.

References

1. Meng, Y., Zhang, W., Zhu, H., Shen, X.S.: Securing consumer IoT in the smart home: architecture challenges and countermeasures. *IEEE Wirel. Commun.* **25**(6), 53–59 (2018)
2. Kishore, P., Barisal, S.K., Vaish, S.: NITRST: a software security tool for collection and analysis of kernel calls. In: *IEEE Region 10 Conference (TENCON)*, pp. 510–515 (2019)

3. Fan, M., et al.: Android malware familial classification and representative sample selection via frequent subgraph analysis. *IEEE Trans. Inf. Forensics Secur.* **13**(8), 1890–1905 (2018)
4. Zhang, J., Qin, Z., Yin, H., Ou, L., Zhang, K.: A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding. *Comput. Secur.* **84**, 376–392 (2019)
5. Zhang, J., Qin, Z., Zhang, K., Yin, H., Zou, J.: Dalvik opcode graph based Android malware variants detection using global topology features. *IEEE Access* **6**, 51964–61974 (2018)
6. Canzanese, R., Mancoridis, S., Kam, M.: System call-based detection of malicious processes. In: *Proceedings of IEEE International Conference on Software Quality, Reliability and Security* (2015)
7. Barisal, S.K., Nayak, G., Naik, B.: Enhanced type safety in Java. *Int. J. Comput. Appl.* **47**(24), 12–16 (2012)
8. Stringhini, G., Shen, Y., Han, Y., Zhang, X.: Marmite: spreading malicious file reputation through download graphs. In: *Proceedings of the 33rd Annual Computer Security Applications Conference (AC-SAC)* (2017)
9. Xu, Y., Xu, S.C., Xu, C., Tao, D.: Multi-positive and unlabeled learning. In: *International Joint Conferences on Artificial Intelligence*, pp. 3182–3188 (2017)
10. De la Puerta, J.G., Sanz, B.: Using Dalvik opcodes for malware detection on Android. *Logic J. IGPL* **25**(6), 938–948 (2017)
11. Barisal, S.K., Behera, S.S., Godbole, S., Mohapatra, D.P.: Validating object-oriented software at design phase by achieving MC/DC. *Int. J. Syst. Assur. Eng. Manag.* **10**(4), 811–823 (2019)
12. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: *IEEE International Conference on Data Mining*, pp. 179–186 (2003)
13. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially supervised classification of text documents. In: *International Conference on Machine Learning*, pp. 387–394 (2002)
14. Barisal, S.K., Dutta, A., Godbole, S., Sahoo, B., Mohapatra, D.P.: MC/DC guided test sequence prioritization using firefly algorithm. In: *Evolutionary Intelligence*, vol. 95, pp. 1–14. Springer (2019)
15. Gong, C., Liu, T., Yang, J., Tao, D.: Large-margin label-calibrated support vector machines for positive and unlabeled learning. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 3471–3483 (2019)
16. Xiao, Y., Liu, B., Yin, J., Cao, L., Zhang, C., Hao, Z.: Similarity-based approach for positive and unlabeled learning. In: *International Joint Conferences on Artificial Intelligence*, pp. 1577–1582 (2011)
17. Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In: *International Conferences on Knowledge Discovery and Data Mining*, pp. 213–220 (2008)
18. Kang, B., Yerima, S.Y., McLaughlin, K., Sezer, S.: N-opcode analysis for android malware classification and categorization. In: *Proceedings of International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)* (2016)
19. Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., Nicholas C.: Malware Detection by Eating a Whole EXE, *Proceedings of arXiv:1710.09435* (2017)
20. Zhang, J., Khan, M.F., Lin, X., Qin, Z.: An optimized positive-unlabeled learning method for detecting a large scale of malware variants. In: *IEEE Conference on Dependable and Secure Computing (DSC)*, pp. 1–8 (2019)



Air Quality Monitoring and Disease Prediction Using IoT and Machine Learning

Mahima Jayaraj^(✉)

Department of Information Technology,
Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India

Abstract. Air pollution is one of the new civilized world's major concerns, which has a serious impact on human health and the environment. The main areas that are affected by toxic pollutants are the industrial areas and areas around it. Air quality prediction focuses mainly on these industrial areas. Industrial level usage of this project requires expensive sensors and an enormous amount of power supply. The World Health Organization (WHO) states that major air pollutants include particulate pollution, Carbon Monoxide (CO), Sulfur dioxide (SO₂) and Nitrogen Dioxide (NO₂). In addition to these mentioned gases, PM or Particulate Matter and VOC or Volatile Organic Compounds components also pose as grave threats. Long and short-term exposure to air suspended or air-borne toxicants has a different toxicological impact on humans. Some of the diseases include asthma, bronchitis, some cardiovascular diseases and long-term chronic diseases such as cancer, lung damage and in extreme cases diseases like pulmonary fibrosis.

In this proposed system, an IoT prototype of a large-scale system using high-end and expensive sensors that measures the different air pollutants in the atmosphere is designed. Gas sensors are used in this prototype to record the concentration of the various pollutants that are encountered in the air on a regular basis. The data collected is stored in a cloud environment and analysis is performed that predicts the possible diseases, illness or health issues related to a particular pollutant, along with other factors like time of exposure to a particular pollutant etc. The framework uses stored data to train the model using multi-label classification with Random Forest algorithm, XG Boost algorithm in the local system. The real time data obtained using the different sensors is tested and the results obtained would be used to predict the possibilities of diseases such as asthma, lung cancer, ventricular hypertrophy etc. and the Air Quality Index (AQI) is calculated. In addition to this, preventive suggestions are also provided which is merely a cautionary message displayed on our LCD display to vacuum clean the room or mop the room thoroughly.

Keywords: IoT · XG boost · Random forest algorithm · Multi-label classification · Ventricular hypertrophy

1 Introduction

Air pollution refers to the condition in which the existence of toxic substances in the atmosphere generated by various human activities and natural phenomena such as volcanic eruptions, results in detrimental effects on the welfare of human beings and the

living environment. Air pollutants are responsible for meticulous air pollution which hampers the human life. Air pollution may cause severe breathing problems in the respiratory system of human body, skin diseases, eye irritation etc. The pollution level in the air is measured using the Air Quality Index or AQI. AQI is a numerical value which tells us how polluted the air is. The higher the value of AQI, the more polluted the area is. Air pollution is most common in large cities where emissions from many different sources are concentrated. Large cities in developing nations tend to have more air pollution than in cities of developed nations. Some of the world's most polluted cities are Karachi, Pakistan; New Delhi, India; Beijing, China; Lima, Peru; and Cairo, Egypt, according to the World Health Organization (WHO). People experience a wide range of health effects from being exposed to air pollution. Smart cities can be developed with low carbon usage in a sustainable way. Effects can be broken down into short-term effects and long-term effects. Short-term effects, which are temporary, include illnesses such as pneumonia or bronchitis. Long-term effects of air pollution can last for years or for an entire lifetime. They can even lead to a person's death. People react differently to different types of air pollution. Like people, animals, and plants, entire ecosystems can suffer effects from air pollution. Reducing air pollution is considerably easy when its sources are identified.

Global air pollution is responsible for:

- 29% of all deaths and diseases are caused due to lung cancer.
- 17% of all deaths and diseases from acute lower respiratory infection.
- 24% of all deaths from strokes.
- 25% of all deaths from heart diseases
- 43% of all deaths and diseases from chronic obstructive pulmonary disease.

An estimated 4.2 million premature deaths globally are linked to ambient air pollution, mainly from heart disease, stroke, chronic obstructive pulmonary disease, lung cancer, and acute respiratory infections in children. Pollutants with the strongest evidence for public health concern, include particulate matter (PM), ozone (O₃), nitrogen dioxide (NO₂) and Sulphur-di-oxide (SO₂). The health risks associated with particulate matter of less than 10 and 2.5 microns in diameter (PM₁₀ and PM_{2.5}) are especially well documented. PM is capable of penetrating deep into lung passageways and entering the bloodstream causing cardiovascular, cerebrovascular and respiratory impacts. It is also the most widely used indicator to assess the health effects from exposure to ambient air pollution.

In children and adults, both short- and long-term exposure to ambient air pollution can lead to reduced lung function, respiratory infections and aggravated asthma. Maternal exposure to ambient air pollution is associated with adverse birth outcomes, such as low birth weight, pre-term birth and small gestational age births. Emerging evidence also suggests ambient air pollution may affect diabetes and neurological development in children. Considering the precise death and disability toll from many of the conditions mentioned are not currently quantified in current estimates, with growing evidence, the burden of disease from ambient air pollution is expected to greatly increase.

2 Related Work

In the past, many methods have been proposed for predicting air quality and their effects on people's health and a number of diseases caused. Akshata Tapashetti utilizes the data mining algorithms to detect the air pollution [1]. Air Cloud [12] has been proposed as a PM_{2.5} monitoring system using particulate matter monitors to infer PM_{2.5} concentration. S. Poduri et al. [13] used sky luminance to estimate air turbidity using mobile phones. The users are asked to select a small area of the sky following which the air quality is estimated by comparing the intensity of the selected sky area with the sky luminance model. Mao et al. [14] use color channels to detect foggy images and estimate the haze degree factor. Liu et al. [7] require manual selection of ROI in the image. However, ROI detection and selection are a non-trivial task. Detecting foggy images and estimating the haze degree factor [15] focuses on haze level estimation rather than actual PM_{2.5} or PM₁₀, they have to only keep 46 images with manually labeled haze level for verifying their model. With the rapid development of smartphone [20], directly estimating air pollution from images starts to gain the potential of being a convenient and less expensive approach because it can cover more areas in a crowd-sourcing manner. Somansh Kumar and Ashish Jasuja in their work- Air Quality monitoring using raspberry pi, uses a sensing unit, raspberry-pi board, Arduino uno and cloud environment to collect data to display on the laptop and phone application. A drawback of their system is that long-term pollution patterns are not discovered. Amita Biswal, J. Subhashini and Ajit Kumar Pasayat came up with a system that monitors air for indoor environment. Data was collected using different gas sensors and sent to the cloud associated with the prototype, but no analysis was done as well as there were no health concerns addressed. Rajana Gore and Deepa Deshpande published their work- Air Data Analysis for Predicting Health risks, which uses a classifier which takes the input of the AQI index at various timelines of a day and the possible health risks are predicted using the classifier. Considering all the work done previously pertaining to this design, it is concluded that Multiclass classifier is more efficient and accurate when compared to other machine learning algorithms.

3 Proposed Work

The Air Quality Index is measured and its impact on human health is predicted in industrial areas. Prediction is done using the Multi-label Classification method -Random Forest classifier. The real time data (i.e. measure of pollutants sensed by the sensors in the prototype) can cause more than one disease. So, multi-label classification is used to predict multiple diseases. The major function is to predict the health issues based on the real-time data observed by the gas sensors such as MQ-7 and MQ-135. Preventive measures are suggested to reduce the effects of pollutants. Figure 1 shows the proposed system architecture.

3.1 IOT Device Prototype

IoT mainly deals with connecting smart devices (embedded electronics devices) to internet by harnessing the advantage of OSI layered Architecture. IoT is a ubiquitous concept where physical objects are connected over the Internet and have unique identifiers

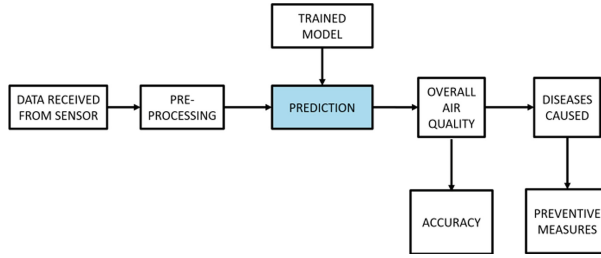


Fig. 1. Block diagram of proposed IoT based air quality prediction system

to facilitate their self-identification with other devices and the ability to generate and transmit data continuously over a network. Consequently, network security, data theft protection, integrity of sensor devices is an important concern in the IoT network due to high growth rate based on the exchanged data and sensor interconnectivity. Accordingly, the Internet of Things elicits significant challenges in the field that benefit the methods for potential realization. The combination of objects with Internet have the powerful analytic capabilities which promises for the transformation of the data from our way of living and work status. In the recent days, air pollution has become a growing issue due to the unchecked increase in the number of infrastructure and industrial plants. So, a plethora of diseases, particularly the ones involving respiratory system, can be ascribed to air pollution. Moreover, recent research has elucidated the importance of micro-level data on pollution to study the deterioration of human health, particularly, highlighting the influence of personal human exposure and external intense exposure to air pollutants. Diseases and health issues include asthma, a chronic and sometimes debilitating airway inflammatory disease caused by noxious gases and particulate matter; chronic obstructive pulmonary diseases, such as bronchitis and emphysema caused by cigarette smoke and car exhausts; lung cancer associated with long-term fine particulate matter exposure.

In this model, the Arduino Uno micro-controller board plays a significant role to which various sensors like MQ135, MQ7 and dust sensor are connected. The sensors are used to detect the level of carbon dioxide, carbon monoxide, temperature, humidity and dust particles present in the environment. Further, the sensor data are stored on a local server and excavated whenever necessary.

Figure 2 shows the Arduino Uno, which is used in this work. Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

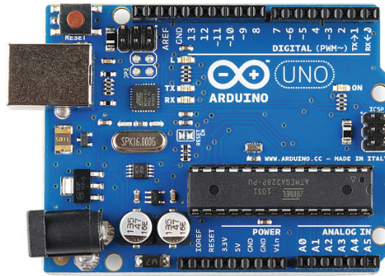


Fig. 2. Arduino Uno

Figure 3 shows the DHT11 sensor. DHT11 sensor is used to measure the temperature and humidity in a particular region, and it is connected to Pin 7 of the Arduino Uno module. This device gives the variation temperature and humidity in degree centigrade and percentage format, respectively.



Fig. 3. DHT11 sensor

Figure 4 shows the Dust Sensor - GP2Y1010AU0F. In this work, to know the level of dust particles in the air, GP2Y 1010AU0F is used. It detects the reflected light from dust particulate in air and it is especially effective in detecting very fine particles such as smoke from cigarette. Additionally, it can distinguish smoke from the house dust. It is mainly used in air purifier, air conditioner and air monitor.



Fig. 4. Dust sensor GP2Y1010AU0F

Figure 5 shows MQ-7. It is a Carbon Monoxide (CO) sensor, which is used in this work. It is suitable for sensing Carbon Monoxide concentrations (PPM) in the air. The MQ-7 sensor can measure CO concentrations ranging from 20 to 2000 ppm. It makes detections by method of cycle high and low temperature, and detect CO at low temperature. It is widely used in domestic CO gas leakage alarm, industrial CO gas alarm and portable CO gas detector.



Fig. 5. MQ-7 gas sensor

Figure 6 shows MQ-135. For monitoring the air quality, a gas sensor, MQ135 is used. It measures the level of NH_3 , NO_x , alcohol, Benzene, smoke, CO_2 in air. The resistance connected to MQ135 is different for various kinds of concentrated gases, so the sensitivity adjustment of components is necessary at time of using. The sensor has wide detecting scope, due to its fast response, high sensitivity, stability and long life. It is mainly utilized in office, buildings and homes for air quality control. Furthermore, the concentration is calculated by using the following formula, $R_s = V_c \times R_L / V_{out} - R_L$, (4) where $V_c = 5$, R_s is the sensor resistance, R_L is load resistance. Figure 7 shows the workflow of the prototype.



Fig. 6. MQ-135 gas sensor

Figure 8 shows the overall design of IoT prototype and the connections of gas sensors to the Arduino Uno board. Further, pre-processing is done and this data is used for training the model and also for testing.

3.2 Predictive Modeling

Machine Learning mainly deals with computational methods that enhance the execution of automating and securing of learning from encounter. The process of learning by a machine from complex set of data and solving critical problems, being more intelligent is what machine learning is all about. Just like, there's a regular weather forecasting done for the next day, in the same way the pollution forecasting model can be used so that people can take precautionary measures. We aim to accurately predict the Air quality and diseases caused by the abnormal concentrations of $\text{PM}_{2.5}$, PM_{10} , SO_2 , NO_2 and CO .

For training our model, we have used an authorized pollution dataset provided by the Tamil Nadu Pollution Control Board [23], which consists of gas pollutants data of $\text{PM}_{2.5}$, PM_{10} , SO_2 , NO_2 and CO , all collected in real-time.

The process of building the prediction models mainly deal with 2 steps shown on Fig. 9:

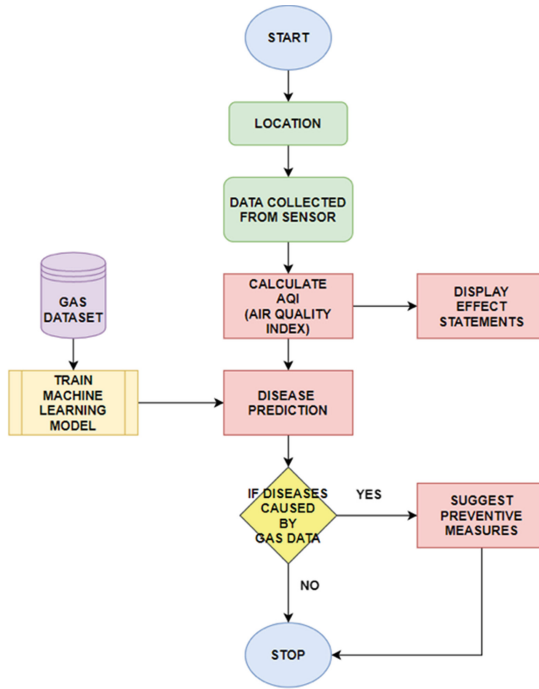


Fig. 7. Overall workflow of IoT prototype

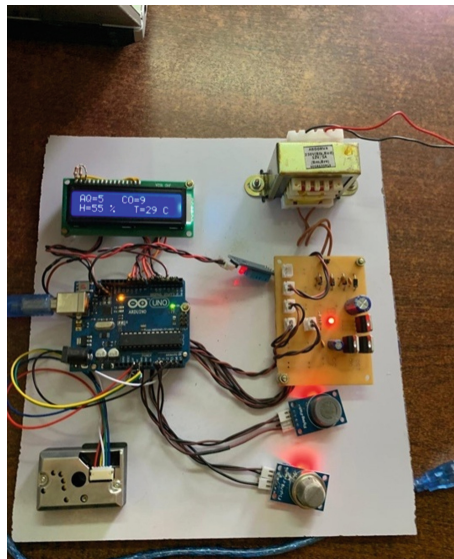


Fig. 8. IoT prototype

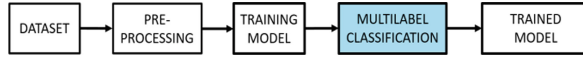


Fig. 9. Design of building a model

- 1) **Data Pre-processing:** The first step of building a prediction model is data pre-processing where data is cleaned, missing values are filled, outliers are removed and also data is arranged in a way to fit for the Machine Learning algorithm (SimpleImputer).
- 2) **Building Model:** Model is built to predict the future, i.e. on the unseen data based on the historical data. In training data known target variables are stored and used in all the algorithms.

For the testing part the developed models are cross-validated and evaluated. The model was evaluated using cross-validation techniques based on Accuracy Score. The models are implemented and their individual performance is evaluated in this work, which gives a qualitative measure of model's performance. For implementing the model in real-life the best features and prediction model will be used for the unseen data. In the volatile environmental change the tautology of training and testing and deploying will be done periodically. This is an iterative process which should be done to improve the model performance. There are numerable algorithms that can be implemented, but out of which Random Forest and XG Boost proved to be the most successful models. In this work, the target variables are multiple. So Multi-label classification is used.

3.2.1 Multi-label Classification

In machine learning, multi-label classification and the strongly related problem of multi-output classification are variants of the classification problem where multiple labels may be assigned to each instance. Multi-label classification is a generalization of multiclass classification, which is the single-label problem of categorizing instances into precisely one of more than two classes; in the multi-label problem there is no constraint on how many of the classes the instance can be assigned to. Formally, multi-label classification is the problem of finding a model that maps inputs x to binary vectors y . Several problem transformation methods exist for multi-label classification, and can be roughly broken down into:

Binary classification problems are the baseline approach, called the binary relevance method, amounts to independently training one binary classifier for each label. Given an unseen sample, the combined model then predicts all labels for this sample for which the respective classifiers predict a positive result. It is essentially different, because a single classifier under binary relevance deals with a single label, without any regard to other labels whatsoever.

A classifier chain is an alternative method for transforming a multi-label classification problem into several binary classification problems. It differs from binary relevance in that labels are predicted sequentially, and the output of all previous classifiers (i.e. positive or negative for a particular label) are input as features to subsequent classifiers.

It is first classifier is trained just on the input data and then each next classifier is trained on the input space and all the previous classifiers in the chain.

The label powerset (LP) transformation creates one binary classifier for every label combination present in the training set. In LP, the problem into a multi-class problem with one multi-class classifier is trained on all unique label combinations found in the training data. The random forest and XG Boost classifier models are implemented in this multi label classification problem. These developed models are cross-validated and evaluated. The model was evaluated using cross validation techniques based on Accuracy Score.

3.2.2 Machine Learning Algorithms

Random Forest is an adaptable, simple-to-utilize machine learning calculation that produces, even without hyper-parameter tuning, an awesome outcome more often than not. It is additionally a standout amongst the most utilized calculations, since its effortlessness and the way that it can be utilized for both grouping and relapse errands. In this context, it can be inferred the appropriateness and usability of random forest calculation and a few other vital things about it.

The pre-processed data is used to train this algorithm in multi-label model. Random forest can be used for both classification and regression. A forest is comprised of trees. It is said that, the more trees it has, the more robust the forest is. Random forests create decision trees on randomly selected samples, get predictions from each tree and selects the best solution by voting as seen in Fig. 10.

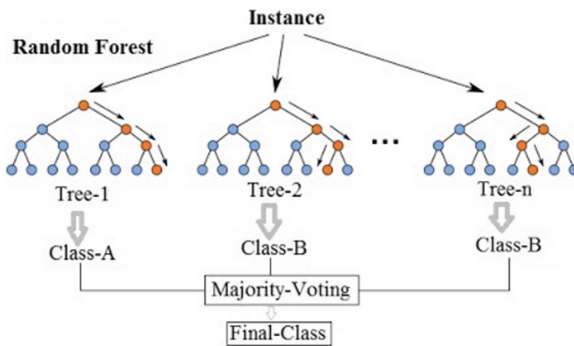


Fig. 10. Random forest classification

As we see in Fig. 11, the algorithm places the attribute of the dataset at the root of the tree. Initially it places the input attribute - PM2.5 at the root of the tree. The training sets get split into subsets. These subsets should be made in a way such that each subset contains data with the same value for a PM2.5 attribute. The above steps are to be repeated for each subset until the leaf node for all branches of the tree is found. Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is chosen randomly. If the problem is perfectly classified, Gini index would be zero. With the Gini index value and number of samples, the algorithm works

based on Fig. 11. Therefore, the decision trees are formed using the attribute PM2.5 as the root. Now, the algorithm uses the remaining attributes as root and compares each other based on Gini indices. The models are then stored to separate files so that they can be used later for predictions.

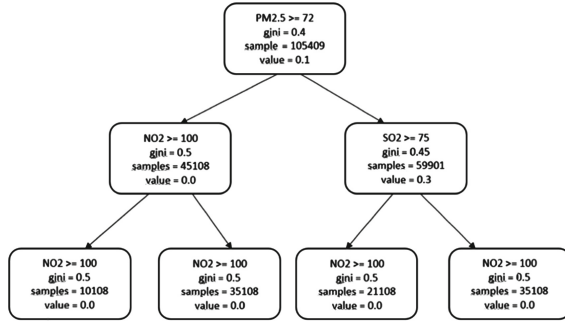


Fig. 11. Part of a random forest

XG Boost is another way to say “Extreme Gradient Boosting”. The expression “Inclination Boosting” is proposed as a Greedy Function Approximation: A Gradient Boosting Machine, by Friedman. Gradient boosting is currently one of the most popular techniques for efficient modeling of tabular datasets of all sizes. XG Boost is a very fast, scalable implementation of gradient boosting. XG Boost depends on this unique model. XG Boost is an implementation of gradient boosted decision trees designed for speed and performance. The Gradient Boosting Model (GBM) based on decision tree is a popular machine learning technique. Gradient boosting is a machine learning (ML) technique typically used for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, based on a decision trees.

XG Boost builds a sequential tree using the input attributes PM2.5, PM10, NO₂, SO₂ and CO. This is possible due to the interchangeable nature of loops used for building base learners, the outer loop that enumerates the leaf nodes (input attributes) of a tree, and the second inner loop that calculates the features (value ranges). The nesting of loops limits parallelization because without completing the inner loop, the outer loop cannot be started. Therefore, to improve runtime, the order of loops is interchanged using initialization through a global scan of all instances and sorting using parallel threads. This switch improves algorithmic performance by offsetting any parallelization overheads in computation. The models are then stored to separate files so that they can be later loaded during predictions.

4 Results and Discussion

From the context of this work, we have discussed the performance of the different models, after the data preprocessing. First part of this section deals with the data that has been generated by our device and the next part is the extension of our model on the

open-source dataset. The results obtained from this work are Air Quality Index (AQI) and diseases caused by real time data obtained from device.

An AQI is defined as an overall scheme that transforms weighted values of individual air pollution related parameters (SO₂, CO, visibility, etc.) into a single number or set of numbers. The result is a set of rules (i.e. most set of equations) that translates parameter values into a simple form by means of numerical manipulation (Fig. 12).

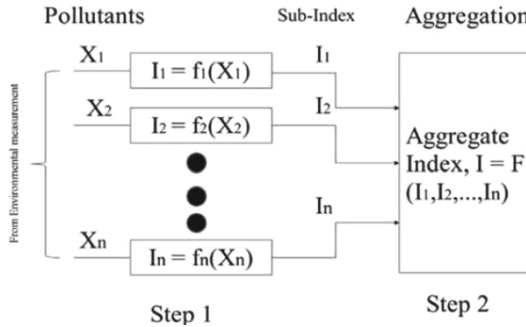


Fig. 12. Formation of air quality index

Air quality Index or AQI is formulated by aggregation of the Quality index of separate pollutants concentration. Primarily two steps are involved in formulating an AQI:

- (i) Formation of sub-indices (for each pollutant) and
- (ii) Aggregation of sub-indices to get an overall AQI.

Formation of sub-indices (I_1, I_2, \dots, I_n) for n pollutant variables (X_1, X_2, \dots, X_n) is carried out using sub-index functions that are based on air quality standards and health effects. Mathematically;

$$I_i = f(X_i), 1, 2, \dots, n \tag{1}$$

Each sub-index represents a relationship between pollutant concentrations and health effect. Aggregation of sub-indices, I_i is carried out with some mathematical function (described below) to obtain the overall index (I), referred to as AQI.

$$I = F(I_1, I_2, \dots, I_n) \tag{2}$$

From the aggregated AQI value, AQI is categorized based on the table Fig. 10. Based on the AQI category, suitable health effects statements for each pollutant are displayed for groups of people (Fig. 13).

The next result obtained from this work is diseases caused by observed data from device. Some diseases are Chronic Bronchitis, Asthma, Genetic mutation, Emphysema, Brain damage etc. These diseases are the target variables of Multi-label classification models. These are caused due to the abnormal value of pollutants in air.

Air Quality Index - Particulate Matter	
301 – 500	Hazardous
201 – 300	Very Unhealthy
151 – 200	Unhealthy
101 – 150	Unhealthy for Sensitive Groups
51 – 100	Moderate
0 – 50	Good

Fig. 13. Air quality index categories

Figure 14 shows the graph of variation of aqi values in different locations. x label contains the AQI values and y label contains the region count. It represents that the number of places having the AQI between the range. It helps to understand the variation of AQI values.

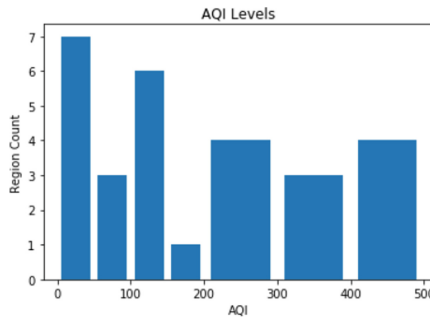


Fig. 14. AQI levels

Figure 15 shows the graph of Disease levels in Different Places. x label contains the diseases and y label contains the region count. It represents that the different diseases caused in different regions. From the data, it helps to understand the diseases caused in number of places. From the disease count levels, the preventive measures can be implemented and helps to avoid the serious effects of diseases on human. On broad scale, the device helps to maintain the stability of pollutants levels and also to avoid the effects of these industrial pollutants.

Figure 16 shows the output module that predicts diseases such as Chronic Bronchitis, Asthma, Genetic Mutation, Emphysema and Brain Damage that caused by pollutant's data obtained from sensors. The diseases are caused because the gas value exceeds the safe range. If PM_{2.5} value exceeds the mark of 72, then it can cause Chronic Bronchitis. Likewise, if PM₁₀ value crosses 150, it can cause Asthma and so on. The prediction is done by multi-label model which is stored as a file in the processing module. These diseases are caused due to abnormal value of pollutants in the air. In addition to this, preventive suggestions are also provided which is merely a cautionary message. These

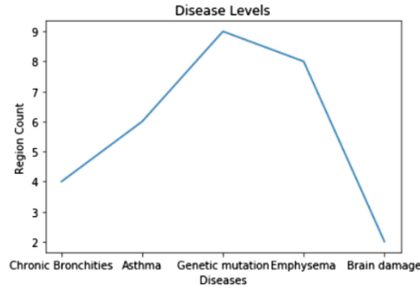


Fig. 15. Disease levels in different places

suggestions, if followed can help to reduce the effects of the pollutants on the human body.

PM2.5	PM10	NO2	SO2	CO	Chronic Bronchities	Asthma	Genetic mutation	Emphysema	Brain damage
154	91	91	132	89	1	0	0	1	0
158	93	93	132	98	1	0	0	1	0
162	89	89	127	92	1	0	0	1	0
166	93	93	128	92	1	0	0	1	0
166	98	98	129	97	1	0	0	1	0
162	103	103	132	99	1	0	1	1	0
162	108	108	128	102	1	0	1	1	1
166	105	105	123	99	1	0	1	1	0
171	109	109	123	94	1	0	1	1	0
166	109	109	127	90	1	0	1	1	0
169	113	113	131	89	1	0	1	1	0
170	116	116	133	87	1	0	1	1	0
165	117	117	130	89	1	0	1	1	0
170	120	120	131	84	1	0	1	1	0
175	120	120	135	81	1	0	1	1	0
176	115	115	134	82	1	0	1	1	0
173	114	114	134	78	1	0	1	1	0
175	112	112	133	74	1	0	1	1	0
176	110	110	137	72	1	0	1	1	0
171	108	108	136	72	1	0	1	1	0
175	113	113	135	70	1	0	1	1	0
176	109	109	135	71	1	0	1	1	0
80	39	43	89	53	1	0	0	1	0
77	36	45	94	48	1	0	0	1	0
74	39	46	95	51	1	0	0	1	0
72	36	51	99	49	1	0	0	1	0
68	39	46	99	49	0	0	0	1	0
66	41	49	95	46	0	0	0	1	0
65	38	50	94	51	0	0	0	1	0
68	35	52	95	50	0	0	0	1	0

Fig. 16. Sample output

Figure 17 shows the bar graph of AQI in Different Places. x label contains the different location and y label contains the AQI values. It represents that each location have its AQI value. It helps to detect whether the places and its corresponding AQI levels are good and helps to take the preventive measures to reduce the pollutant levels.

Figure 18 shows the comparison of the performance analyses of different machine learning algorithms. Random Forest model of Classifier Chain Multi-label method has proven to show the highest accuracy score of 99.65% and the accuracy count of 31,520 of 31,623. The precision and recall score shows the ability of model to find the True Positive samples. Thus, the Random Forest Classifier model is selected for predicting the diseases from the data picked up by the sensor.

5 Conclusion

Air quality is a critical issue that straightforwardly influences human wellbeing. Air quality information are gathered remotely from checking bits that are outfitted with a

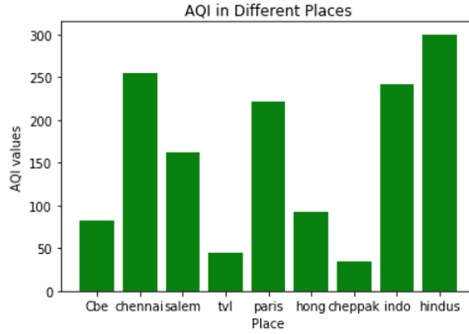


Fig. 17. AQI in different places

Multilabel Type	Algorithm	Accuracy Score	Accuracy Count	Precision	Recall	F1-Score	Fbeta-Score
Binary Relevance	Naïve Bayes	0.9606	30360	0.9938	0.9914	0.9926	0.9933
	Random Forest	0.9855	31020	0.9933	0.9938	0.9926	0.9954
	XGradient Boost	0.9901	31280	0.9974	0.9925	0.9968	0.9926
Classifier Chains	Naïve Bayes	0.9332	28789	0.9929	0.9932	0.9926	0.9938
	Random Forest	0.9965	31520	0.9975	0.9984	0.9996	0.9983
	XGradient Boost	0.9931	31372	0.9933	0.9926	0.9938	0.9983
Label Powerset	Naïve Bayes	0.9304	28007	0.9926	0.9935	0.9933	0.9914
	Random Forest	0.9901	31295	0.9918	0.9939	0.9929	0.9966
	XGradient Boost	0.9923	31301	0.9933	0.9925	0.9919	0.9938

Fig. 18. Algorithm performance analyses

variety of vaporous also, meteorological sensors. This information is investigated and utilized as a part of anticipating fixation estimations of contaminations utilizing savvy machine to machine stage. The stage comprises of ML based calculations to construct the estimating models by training from the gathered information. However, we can conclude that we can use gradient boosting method for prediction, preferably XG Boost because of its level-wise approach and helps in building a model which has low bias and low variance. The overall performance is pretty good as a forecasting model, what can be used as to make a daily forecast of pollutants level and its disease causing factor and the overall air quality in all industrial areas.

References

1. Tapashetti, A., Vegiraju, D., Ogunfunmi, T.: IoT-Enabled Air Quality Monitoring Device (2018)

2. Xu, D.: Mining PM_{2.5} and Traffic Conditions for Air Quality (2018)
3. Liu, X., Xu, P., Chen, X.: IOT-Based Air Pollution Monitoring and Forecasting System (2015)
4. Gómez, J.E., Marcillo, F.R., Triana, F.L., Gallo, V.T., Oviedo, B.W., Hernández, V.L.: IoT for Environmental Variables in Urban Areas (2017)
5. Xia, X., et al.: A Comprehensive Evaluation of Air Pollution Prediction Improvement by a Machine Learning Method (2015)
6. Raipure, S., Mehetre, D.: Wireless Sensor Network Based Pollution Monitoring System in Metropolitan Cities (2015)
7. Rybarczyk, Y., Zalakeviciute, R.: Machine Learning Approach to Forecasting Urban Pollution (2016)
8. Sumithra, A., Jane Ida, J., Karthika, K., Gavaskar, S.: A Smart Environmental Monitoring System Using Internet of Things (2016)
9. Shah, J., Mishra, B.: IoT Enabled Environmental Monitoring System for Smart Cities (2016)
10. Taneja, S., Sharma, N., Oberoi, K., Navoria, Y.: Predicting Trends in Air Pollution using Data Mining (2016)
11. Cairncross, E.K., John, J., Zunckel, M.: A novel air pollution index based on the relative risk of daily mortality associated with short-term exposure to common air pollutants. *Atmos Environ.* (2007)
12. Kan, H., et al.: Part 1 - A time series study of ambient air pollution and daily mortality in Shanghai, China. *Res. Rep. Health Eff. Inst.* (2010)
13. Zhou, N., et al.: Air pollution and decreased semen quality: a comparative study of Chongqing urban and rural areas (2014)
14. Chen, B., Kan, H.: Air pollution and population health: a global challenge. *Environ. Health Prev. Med.* (2008)
15. Molina, M.J., Molina, L.T.: Megacities and atmospheric pollution. *J. Air Waste Manag. Assoc.* (2004)
16. Air pollution: Consequences and actions for the UK, and beyond. *Lancet* (2016)
17. WHO. Database: Outdoor Air Pollution in Cities (2010)
18. Mawer, C.: Air pollution in Iran. *BMJ* (2014)
19. Lovett, G.M., et al.: Effects of air pollution on ecosystems and biological diversity in the eastern United States. *Ann. N.Y. Acad. Sci.* (2009)
20. Mellouki, A., George, C., Chai, F., Mu, Y., Chen, J., Li, H.: Sources, chemistry, impacts and regulations of complex air pollution: preface. *J. Environ. Sci. (China)* (2016)
21. Camargo, J.A., Alonso, A.: Ecological and toxicological effects of inorganic nitrogen pollution in aquatic ecosystems: a global assessment. *Environ. Int.* (2006)
22. Veras, M.M., Caldini, E.G., Dolhnikoff, M., Saldiva, P.H.: Air pollution and effects on reproductive-system functions globally with particular emphasis on the Brazilian population. *J. Toxicol. Environ. Health B Crit. Rev.* (2010)
23. Tamil Nadu Real time monitored data. <https://tnpcb.gov.in>



A Mobile Based Market Information System

Adebayo Abayomi-Alli¹(✉), Sanjay Misra², Mojisola Dada¹,
Christian Yetunde Alonge¹, John Bosco Agbaegbu¹,
Oluwasefunmi ‘Tale Arogundade¹, and Ravin Ahuja³

¹ Federal University of Agriculture, Abeokuta, Nigeria
{abayomia11ia, arogundadeot}@funaab.edu.ng

² Covenant University, Ota, Nigeria
sanjay.misra@covenantuniversity.edu.ng

³ ShriVishwkarma Skill University, Gurgaon, India

Abstract. Effective market information systems help to reduce information asymmetries, increase competitiveness, and improve efficiency in the marketing network. Thus, lack of dissemination of market information and the bargaining capability of the traders across the agricultural supply chain is a major concern among small holder farmers in Nigeria. The advent of mobile phones serves as a great tool for awareness and information dissemination to people. A Mobile Based Market Information System is presented in this study. It serves as a means of enhancing farmers marketing strategy by providing market visibility for transacting business both within and outside the region. The proposed system employs Google map API which shows the location of the registered markets. The system was implemented using JavaScript, PHP/MySQL and Phoneygap/Cordova. XAMPP database server was used for data management. The application was tested and validated by 50 respondents, the results returned high acceptance rate, high perception on usage and ease of use.

Keywords: Mobile application · Market · Information system · Agriculture · Mobile phones

1 Introduction

The introduction of Information Communication Technologies (ICT) have provided new ways, and techniques of disseminating information among people in the society and has led to developments in all sectors of Nigerian economy. Mobile phones as part of the inventions of ICT, has brought significant changes in the agriculture sector [1]. The use of mobile phones by farmers has helped them in getting recent information on ways to improve their agricultural techniques from crop plantation to marketing of farm produce to the consumers. Mobile technology has also aided the increase of network among larger communities most especially among farmers in a topical region on exchange of ideas and experience regarding agricultural farming. The rate of adoption of mobile phone technology in developing countries is increasing with attendant benefits to the society [2].

According to Poulsen [3], one of way of incorporating ICT into agriculture is establishment of Farmers' markets which serve as a means of bringing producers together with consumers under direct marketing alternatively known as Food Networks. This was done for several reasons: to allow producers to find markets for their produce and also retain the profits resulting from the sale of their products, and to enable the consumers get fresh products directly from the production unit at an affordable price [4]. This system has several advantages such as providing avenues for producers to sell their produce to consumers which in the process enhanced the production growth, minimizes food wastage and also increased income rate [5].

Thus, ICT offers the potential to increase the information flow among agricultural stakeholders hence increasing the transparency of agricultural exchange in agrarian economies such as Nigeria. It was observed that markets in most developing countries often fail because smallholder farmers who form the majority of agricultural producers are discouraged [6]. The failure of agricultural markets for farmers often results from lack of access to information between farmers and buyers which leads to setbacks in agricultural growth [7]. Consequently, majority of farmers sell their produce in local low-paying markets rather than traveling to distant markets with a good purchasing power [8]. One of the way of providing solution to such is the use of mobile phone by farmers to help them in increasing market participation through access to information on available markets and prevailing market prices. Also it helps the consumers to have the knowledge of available markets where agricultural produce or any items for human use can be purchased at a very reasonable price.

Considering all of the identified issues faced by farmers, this research proposes a Mobile Based Market Information System that will provide farmers with available agricultural markets within and outside their region. Not only will this make selling easier but it will also boost income. The proposed application is not limited to farmer's markets alone, but also for different kinds of goods that can be gotten in a market, (cosmetics, wares, electronics, IT, etc.). A Google Map and Geographical Information system (GIS) is embedded to the application which serves as a location based and positioning system for determining the location of users (traders).

The rest of the paper is organized as follows: the related works on agricultural marketing, mobile marketing and integration of ICT tools in agriculture are described in Sect. 2, Sect. 3 describes the research methodology of the proposed mobile application. In Sect. 4, the implementation process and evaluation of the proposed mobile application was described. The conclusion ends the paper in Sect. 5.

2 Related Works

The impact of ICT on Agriculture and marketing of Agricultural products is discussed in this section.

2.1 The Impact of ICT on Agricultural Market

Recent efforts to improve market access by smallholder farmers have been directed to mobile technology which has recorded rapid penetration in Africa [9]. The innovation

of mobile phones has facilitated access to information by farmers to help them increase their bargaining power and control over external events [10]. For instance, in Tanzania, the adoption of mobile phones, under the five project of the Agricultural Marketing Systems Development Programme (AMSDP), has virtually transformed agricultural business through the way producers access vital market information [11].

A study by [12] in Uganda on the impact of mobile phone coverage expansion on market participation revealed that there is great record of market participation, and higher income rate among banana farmers which resulted from the use of mobile technology.

Similarly, in [13] a study on the use of mobile phones by 134 younger agricultural entrepreneurs resulted into expansion in their market due to high and fast rate of accessing information which in turn led to profit increase after two years. Also, Aker [14] discovered that mobile phones has positive effects on the welfare of traders and consumers in Niger by increasing traders' profits up to 29% and reducing average consumer grain prices by 3.5%. [14] also reported that the use of mobile phones enabled traders to utilize markets extensively and to reach out to more customers.

Lastly, [15] carried out a study in a small community in Morocco, the result showed farmers are inspired by the growth of agriculture, increased market orientation, and diversification from low-value crops into higher-value with corresponding higher income rate because of the mobile phones usage. Similarly, it encouraged farmers to engage directly with wholesalers by getting better understanding of prices and penetrating larger and more distant markets.

2.2 Marketing Agricultural Products

In this section, some of the study on marketing of agricultural products and e-commerce was reviewed.

Yang [16] conducted a research on the live broadcast marketing management of e-commerce with the consideration of agricultural products. The author considered the growth of online commerce of agricultural products as related to China. It was stated that this was still being explored and developing with existing problems that should be addressed. Wang and Wang [20] considered agriculture as a major means of development for rural communities using the online-to-offline (O-2-O) marketing approach. Ambika [17] considered marketing and irrigation as a major problem of agriculture. The authors observed that farmers are unaware of recent trends and then presented a website for marketing agricultural produce without a mediator or middle man. The platform makes the leasing of agricultural tools easy as well as educates farmers on current trends in the field. Beyond the support given on marketing, farmers could get awareness on irrigation practices in order to increase yield.

Radosavljević [18] proposed an approach to efficiently place agro-food product as a major strategic objective for economic development in Serbia. They stated that quality food produce is on high demand and it requires a strategy for meeting them as quickly as possible. The authors conducted an in-depth analysis of several channels with which customer demands are met and stated that technology operations should be exhibited to ensure fast, flexible and efficiency in the agricultural supply chain.

Liu and Walsh [19] considered the improving standard of living in China and its immense interest in health, agriculture and nutrition. Products in these areas are considered highly demanded and profitable. The authors stated that despite the immense number of e-commerce websites for agricultural products, a number of them are difficult to use. They analysed the problems of developing e-commerce for agriculture products. They proposed that the legislative mechanism should be improved and more specialized systems need to be considered.

None of the study considered using mobile technology with geo-location technology to improve or advance agricultural business. This research puts some of the problems identified and employs effective approach for enhancing agricultural based marketing.

3 Methodology

Mobile Based Market Information System is an application that supports Android operating system. This application uses the GPS function and Google map, available in most of Android phones. With this application installed on mobile phone, first step is for the user to launch the application and register their details which is stored in the database server. This application will send co-ordinates from their mobile phones to Server using GPS to determine their location, and then check the database to determine the markets found within their region. The development of the mobile application is developed with the consideration of the following key entities:

Users

Users consist of registered traders which can also be farmers and an administrator. Traders are the end users that make use of the mobile phone to interact with the application through Internet.

Interface

The interface is the tool for different users to communicate with the system. This include android platform, text message (SMS), and Internet.

Middleware

This acts as an intermediary between the core system and the user interface. This includes SMS gateway, GPRS gateway, and cell tower. The SMS gateway enables sending and receiving of text message to users. The GPRS and cell towers enables sensing and detection of user's location from location target to the application server. This enables the application to show the user's location when it is being launched and then display the markets within the region.

Core System

The core system is the key elements in the entire design which includes the database server, application server, web server. The database server enables storage of individual users, and market information and scheduled web contents. Getting a timely message on market information requires the server to send automated alerts to users.

3.1 Design Process Modelling

The design process for the mobile application is illustrated graphically by showing the interactions between the users and the application. It shows how the information flow is conveyed in a systematic and standard manner. Some of the models used are case diagram, sequence diagram, class diagram.

Use Case Diagram

A use case diagram is a representation of a user's interaction with the application. It shows the relationship between the users and highlight the key activities to be performed by each users of the application system in a diagrammatic way which is represented either circles or ellipses. Figure 1 shows the use case diagram for the proposed mobile application. Figure 2 shows the class diagram for the proposed system. The sequence diagram that shows the steps of employment for the mobile application is shown in Fig. 3.

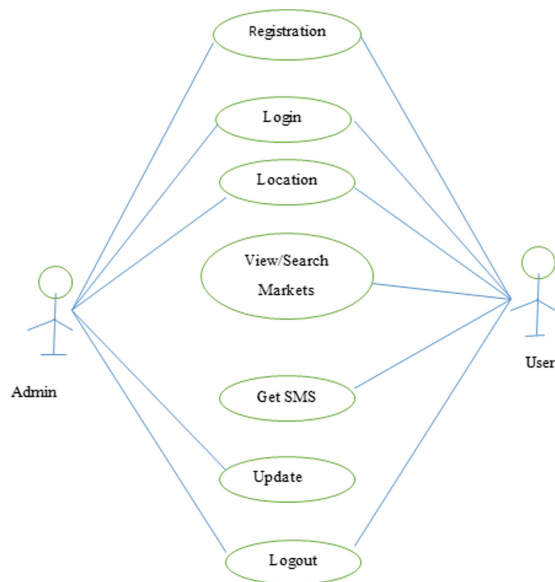


Fig. 1. Use case diagram for the proposed mobile application.

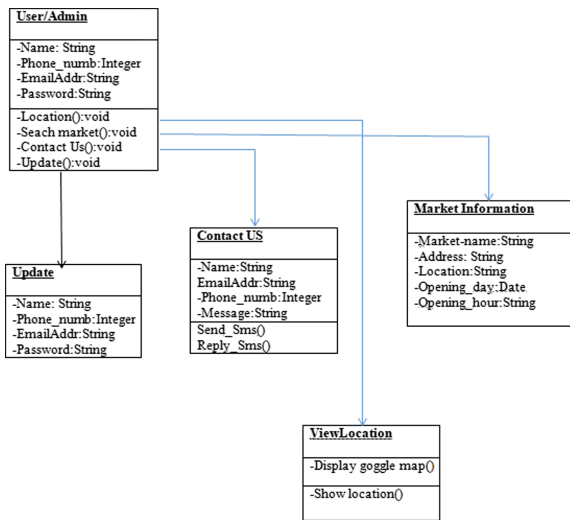


Fig. 2. Class diagram for the proposed mobile application

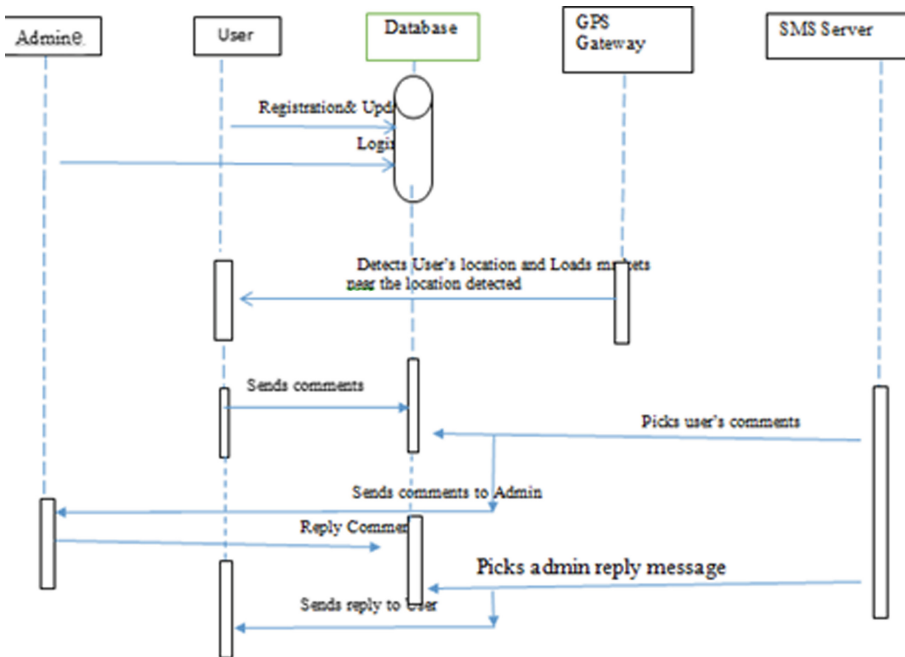


Fig. 3. Sequence diagram for the proposed mobile application

4 Result and Discussion

4.1 Implementation

The application is designed from a user point of view. This application supports graphical user interface which enables the users to interact and accomplish their task with ease. The design was simple and understandable. The database used is MySQL where all the information related to markets and users is stored. PHP was used as a server side scripting language to connect the application with the database and files located on the XAMPP Server. GPS and Google map was integrated into the application to enable locations and market search. Figure 4 shows the market search page of the mobile application. It enables the user to search for markets within and outside their region and then return the registered markets in Google map. Figure 5 shows the feedback page that enables users give their perception of the mobile application.



Fig. 4. Search page and result of the mobile application

4.2 Evaluation

The proposed application was evaluated using questions raised through user perception questionnaire based on the Technology Acceptance Model (TAM). Four factors from TAM were used, which are:

1. Perceived Usefulness (PU): To access users perception of the usefulness of the Mobile based Market Application System
2. Perceived Ease of Use (PEOU): To determine what degree of ease is associated with the use of the mobile application.

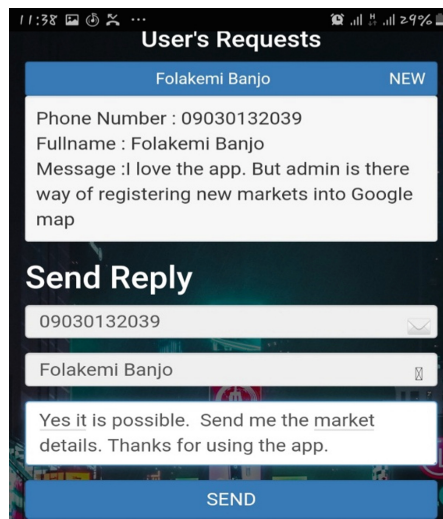


Fig. 5. Feedback page and result of the mobile application.

3. Attitude towards Using the Mobile Application (ATTD): This is to determine if the use of the mobile application is a good approach of making market search.
4. Behavioral Intention (INT): It is used to determine the degree to which a user plans using the mobile application in carrying out their trading activities.

Overall Usability Results

A total of 12 questions were formulated and measured on a 5-point Likert scale. This questionnaire was administered to 50 respondents which are 23 males and 27 females. The overall usability feedback of Mobile Based Market Information System from respondents is analyzed below in Table 1, which showed 4.52 mean score for perceived usefulness, 4.45 mean score for perceived ease of use, 4.67 mean score for attitude towards usage, and 4.59 mean score for behavioral intention coupled with the overall average mean score of 4.55. This indicated the overall acceptance of the application by both genders in the society. Table 1 shows the overall usability response and further analyzed in Fig. 6.

Table 2 shows the responses of users regarding the perceived usefulness of the mobile application with a mean of 4.52. Which means larger percentage of the respondents agree to the usefulness of the mobile application to carry out their trading activities.

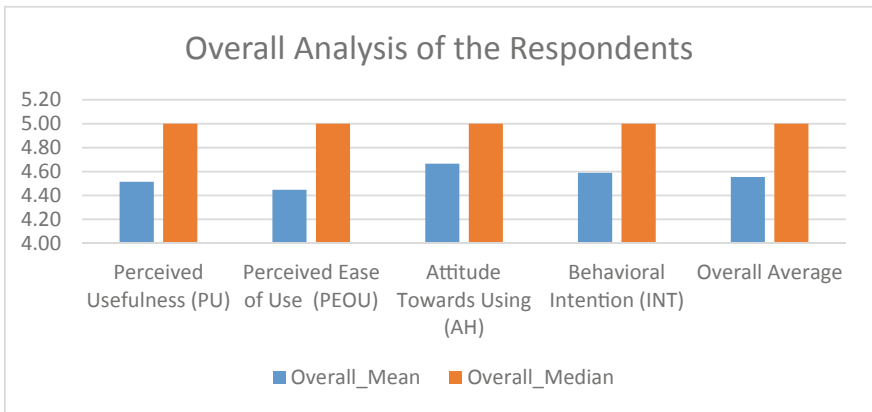
Three items were used to measure the Perceived Ease of Use (PEOU) TAM construct, as shown in the table above. Further investigation of participants' feedback for PEOU suggests that MSMS App is convenient to use (Table 3).

Attitude towards the usage of the app (ATTD) was measured through three questions; findings indicated agreement on participants' idea to use the application, with good mean score of 4.67 and the average of 5.00 (Table 4).

In conclusion, from the user's responses, it was observed that the Intention of Users (INT) about the MSMS application regarding the usage of the application was highly

Table 1. An overall usability result from the evaluation of the mobile application.

Statements	Overall mean	Overall median
Perceived Usefulness (PU)	4.52	5.00
Perceived Ease of Use (PEOU)	4.45	5.00
Attitude Towards Using the Application (ATTD)	4.67	5.00
Behavioural Intention (INT)	4.59	5.00
Overall Average	4.55	5.00

**Fig. 6.** Chart showing the user perception evaluation result of the mobile application.**Table 2.** Evaluation of users about the perceived usefulness of the application

Factors	Questions	Mean	Median
PU	1	4.46	5.00
	2	4.48	5.00
	3	4.52	5.00
	4	4.60	5.00
Overall (PU)		4.52	5.00

encouraging with mean score of 4.59 and average of 5.00 compared to other constructs. This indicated that the application is generally perceived as a useful tool to small holder farmers and other potential users as seen in Table 5.

Table 3. Evaluation of perceived ease of use

Factors	Questions	Mean	Median
PEOU	5	4.30	5.00
	6	4.42	5.00
	7	4.62	5.00
Overall (PEOU)		4.45	5.00

Table 4. Evaluation of user's attitude

Factors	Questions	Mean	Median
ATTD	8	4.64	5.00
	9	4.52	5.00
	10	4.84	5.00
Overall (ATTD)		4.67	5.00

Table 5. Evaluation of users intension

Factors	Questions	Mean	Median
INT	11	4.66	5.00
	12	4.52	5.00
Overall (INT)		4.59	5.00

5 Conclusion

This study describes the development of a mobile phone based market information system on the Android operating system. This mobile application serves as easy aid to access different food markets, it provides a platform for traders to get market related information such as where to carry out trading activities, connect to buyers and/or retailer, obtain other logistics for transportation, etc. This will help farmers and other traders predominantly in Nigeria to prevent the restriction being faced about marketing of products and inadequate market information. This further alleviates the limitation of farmers and traders to local markets only. It thus connects farmers to regional and global markets where food produces could be sold at wholesale directly and at competitive process determined by the farmers. The adoption of the application will help the farmers to know the markets to take their produce to for selling, likewise inform the trades of the available markets to do exchange at a cheaper rate and high quantity. This will increase the farmer's profits,

encourage the involvement of women and youths, and ensure food local and national food security. The application is designed to ensure good output, portability, productivities as well as reliability. It can be easily handled at high speed of operation and cost effectiveness. A well-developed menu driven and user friendly system for promoting an easy interaction between the user and the application was designed. This system from validation prove to be highly flexible, economical, efficient and hereby recommended for use.

References

1. Yahya, N.: Agricultural 4.0: its implementation toward future sustainability. In: Yahya, N. (ed.) *Green Urea. GET*, pp. 125–145. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7578-0_5
2. Deichmann, U., Goyal, A., Mishra, D.: Will digital technologies transform agriculture in developing countries? *Agric. Econ.* **47**(1), 21–33 (2016). <https://doi.org/10.1111/agec.12300>
3. Poulsen, M.N.: Cultivating citizenship, equity, and social inclusion? Putting civic agriculture into practice through urban farming. *Agric. Hum. Values* **34**(1), 135–148 (2017). <https://doi.org/10.1007/s10460-016-9699-y>
4. Zikiene, K., Pileliene, L.: Research of factors influencing customer switching behavior in farmers' markets in Lithuania. In: 22nd Proceedings of Annual International Scientific Conference-Research for Rural Development, New York, USA, vol. 2, pp. 150–157 (2016). <https://doi.org/10.2139/ssrn.2792011>
5. Malagon-Zaldua, E., Begiristain-Zubillaga, M., Onederra-Aramendi, A.: Measuring the economic impact of farmers' markets on local economies in the basque country. *Agriculture* **8**(1), 1–14 (2018). <https://doi.org/10.3390/agriculture8010010>
6. Salami, A., Kamara, A.B., Brixiova, Z.: Smallholder agriculture in East Africa: trends, constraints and opportunities. Working Papers Series, African Development Bank, Tunis, Tunisia, no. 105, pp. 1–52 (2010)
7. Nandi, R., Gowdru, N.V., Bokelmann, W.: Factors influencing smallholder farmers in supplying organic fruits and vegetables to supermarket supply chains in Karnataka, India: a transaction cost approach. *Int. J. Rural Manag.* **13**(1), 85–107 (2017). <https://doi.org/10.1177/0973005216689319>
8. Anadozie, C., Fonkam, M., Cleron, J.P., Kah, M.M.: The impact of mobile phone use on farmers' livelihoods in post-insurgency Northeast Nigeria. *Inf. Dev.* (2019). <https://doi.org/10.1177/0266666919886904>
9. Okello, J., Kirui, O.K., Njirani, G.W., Gitonga, Z.M.: Drivers of use of information and communication technologies by farm households: the case of smallholder farmers in Kenya. *J. Agric. Sci.* **4**(2), 111–124 (2012). <https://doi.org/10.5539/jas.v4n2p111>
10. Baumüller, H.: The little we know: an exploratory literature review on the utility of mobile phone-enabled services for smallholder farmers. *J. Int. Dev.* **30**(1), 134–154 (2018). <https://doi.org/10.1002/jid.3314>
11. Mumbero, M.: *Mobil Phones Means Letter Prices*. African Farming and Food Processing Magazine, p. 31. Alain Charles Publishers Ltd., London (2007)
12. Muto, M., Yamano, T.: The impact of mobile phone coverage expansion on market participation: panel data evidence from Uganda. *World Dev.* **37**(12), 1887–1896 (2009). <https://doi.org/10.1016/j.worlddev.2009.05.004>
13. Mwalukasa, N., Mlozi, M.R., Sanga, C.A.: Influence of socio-demographic factors on the use of mobile phones in accessing rice information on climate change adaptation in Tanzania. *Glob. Knowl. Mem. Commun.* **67**(8/9), 566–584 (2018). <https://doi.org/10.1108/GKMC-01-2018-0006>

14. Aker, J.C.: Does digital divide or provide? The impact of mobile phones on grain markets in Niger. Center for Global Development Working Paper 154, pp. 1–62 (2008)
15. Hagos, F.G., Hailelassie, A., Getnet, K., Gebregziabher, G., Bogale, A., Getahun, Y.: Marketing patterns of rained and irrigated systems: do they differ? LIVES Working Paper 14, International Livestock Research Institute (ILRI), Nairobi, Kenya, pp. 1–20 (2016). <https://doi.org/10.13140/RG.2.1.4266.8407>
16. Yang, Z.: Research on live broadcast marketing management under the background of rapid development of agricultural products e-commerce. In: 4th International Conference on Modern Management, Education Technology and Social Science (MMETSS). Atlantis Press (2019). <https://doi.org/10.2991/mmetss-19.2019.144>
17. Ambika, N., Meharoofa, F.T.M., Menezes, F.E., Deepika, M.K., Sandesha, S.P.K.: Agro-amigo: agro based web marketing and automated irrigation. *Perspect. Commun. Embed.-Syst. Signal-Process. (PiCES)* **2**(11), 286–290 (2019)
18. Radosavljević, K., Vučić, I., Plavšić, M.: Expansion of marketing channels and their influence on trade in agri-food products. *Int. Exp. Ekonomika preduzeća* **67**(5–6), 370–383 (2019). <https://doi.org/10.5937/EKOPRE1906370R>
19. Liu, X., Walsh, J.: Study on development strategies of fresh agricultural products e-commerce in China. *Int. Bus. Res.* **12**(8), 61–70 (2019). <https://doi.org/10.5539/ibr.v12n8p61>
20. Wang, Y.: Research on the application of online-to-offline mode in agricultural products e-commerce based on new retail. In: International Conference on Economic Management and Cultural Industry (ICEMCI 2019), pp. 272–275. Atlantis Press (2019). <https://doi.org/10.2991/aebmr.k.191217.050>



Reduction of Sidelobe Levels in OFDM Radar Signal Using Two Samples Sliding Window Adder (TSSWA) Algorithm

M. P. Raghu Srivatsa and C. G. Raghavendra^(✉)

Department of Electronics and Communication Engineering, M.S. Ramaiah Institute of Technology, Bangalore 560 054, India
cgraagu@msrit.edu

Abstract. This paper aims to reduce the sidelobe levels of a Multicarrier Complementary Phase Coded (MCPC) radar signal. MCPC radar signal takes advantage of the properties of orthogonal frequency division multiplexing (OFDM). A MCPC signal is obtained by phase modulating the N subcarriers and every subcarrier are spaced apart by $1/t_b$ duration, which forms an OFDM signal. The detection range for given radar is independent of the waveform which led to the implementation of MCPC signal using orthogonal transforms. The sidelobe level and Peak to Mean Envelope Power Ratio (PMEPR) are the problems which should be addressed to improve the performance of radar. Simulation result shows, the proposed method called Two Samples Sliding Window Adder (TSSWA) algorithm results in lower sidelobe levels.

Keywords: MCPC · PMEPR · OFDM · TSSWA · Autocorrelation

1 Introduction

There is a huge surge in demand for more information and the rapid growth of multimedia wireless applications, all points towards the need of a novel technology which aide very high speed data transmission rates. Similarly in the field of radar communications also there is a dire need for such new technique which balances the contrarian relationship between range and resolution detection capabilities. The range (or delay) resolution of a radar structure is inversely relative to the bandwidth of the signal transmitted. Therefore, the best evident way to enhance the range and resolution is to curtail the pulse width, because the pulse duration of each bit is inversely proportional bandwidth [1]. On the other hand, for the better detection capacity of radar, the system must transmit with greater signal energy, and this can be attained by smearing long duration pulses, because the radar transmitting systems are operated close to their peak power limitations. One suggested method is to follow the pulse compression method to counterbalance the differences between the resolution and range of the radar signal.

There may be a huge demand for bandwidth if single carrier methods are used for radar signal communication. But, taking the advantage of multitone or multicarrier

methods, i.e., OFDM for the transmission of the radar signal may help to reduce the bandwidth requirements [2]. OFDM is a method of multiplexing the modulated signals. Multiplexing in general represents combing the subcarriers signals, which is formed by various sources. In multicarrier signal the multiplexing is applied to independent (or subcarrier) signals, however, these subcarrier signals are a sub-set of the complex (or main) signal. In OFDM the signal is divided into subcarriers, modulated by data (or phase) and later it is recombined to obtain the final OFDM signal. The concept of OFDM can be applied in radar for detecting the target.

In 2000 Levanon [3] coined a concept called as multitone or multicarrier complementary phase coded (MCPC) radar signal which was developed by the use of digital phase coded [4, 5] method. The Peak to Mean Envelope Power Ratio (PMEPR) in the MCPC signal is supposed to be less. In the conventional method of MCPC signal it was observed, if an attempt is made to lessen the PMEPR value, the sidelobe levels increases and vice versa, this may result into the wastage of energy due to more sidelobes.

This paper tries to reduce sidelobe levels and further improve PMEPR values. The proposed method Two Samples Sliding Window Adder (TSSWA) [6] algorithm decreases the sidelobe levels by directly manipulating the autocorrelation function (ACF) which leads to clear distinction between mainlobe and sidelobes. In reducing the sidelobes levels few authors has attempted, namely, by using constellation adjustment method in [7], an improvement of of 35 dB is obtained using genetic algorithm and reported in [8] but computational complexity is high. All the above cited literatures will try to reduce sidelobe and focused towards data transmission problem. This paper concentrates on reducing the sidelobe level for radar signals.

The structure of this paper is as follows. Firstly, in Sect. 2, characteristics of MCPC Signal are illustrated. In Sect. 3, MCPC based on two samples sliding window adder (TSSWA) algorithm is discussed. In Sect. 4, a brief conclusion has been given which highlights the efficiency of the proposed technique.

2 Characteristics of MCPC Signal

The multifrequency phase coded radar signal is based on the theory of OFDM scheme. The MCPC signal employs N number of carriers which is composed of N subcarriers transmitted all in parallel, which are parted in frequency by inverse of each duration of t_c bit. This concept of splitting the frequency yields OFDM notion, which is well known in the field of communication. In the OFDM every carrier is phase modulated with N number of bits and each bit will be of t_c duration and these phases are developed with polyphase methods namely P3 and P4 [9]. A schematic block diagram for developing MCPC signal which is based polyphase P4 codes is shown in Fig. 1.

The phases for P4 digital phase modulation scheme with N phases is given in Eq. (1)

$$\phi_n = \frac{\pi}{N}(n-1)(n-N-1) \text{ where } n = 1, 2, 3, \dots, N \quad (1)$$

The advantage of following phase coding scheme in modulation yields fewer phase values exhibiting ideal periodic thumbtack autocorrelation function with peaky mainlobe and zero sidelobes.

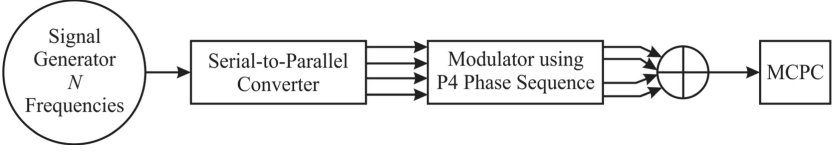


Fig. 1. Block diagram of MCPC-OFDM signal generator

The complex envelope of MCPC OFDM signal is given by Eq. (2).

$$s(t) = \sum_{v=1}^N \left[\sum_{u=1}^N \exp(j\phi_{u,v}) \exp \left\{ j \left[2\pi f_s t \left(\frac{N+1}{2} - u \right) \right] \right\} \right] x[t - (v-1)t_c] \quad (2)$$

$$\text{where } x(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq t_c \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

There are subcarriers and phase modulation chips with as the duration of each chip. The term $\exp \left\{ j \left[2\pi f_s t \left(\frac{N+1}{2} - u \right) \right] \right\}$ corresponds to the subcarrier u . The term $\exp(j\phi_{u,v})$ represents the v^{th} element of the u^{th} sequence modulating the subcarrier u . The term depends mainly on v . When $v = 1$, it is the unit pulse in the first chip duration. Similarly, for $v = N$, it is the unit pulse at N^{th} chip duration. $\phi_{u,v}$ denotes the v^{th} phase element of the u^{th} sequence. The term f_s denotes the frequency separation between any two adjoining sub-carriers.

In the MCPC signal, N sequences are transmitted on N subcarriers with each sequence containing N phase modulated chips. Thus N different subcarriers are modulated by N different phase sequences of length N . The frequency separation between any two successive subcarriers is the reciprocal of chip duration which defines an OFDM signal. The unique feature of this signal is the pulse to pulse diversity i.e., each pulse is complementary to one another. Also this signal provides a high spectral efficiency as the power spectrum is nearly rectangular with a spectral width of $\frac{N}{2t_c}$.

Figure 2(a) represents the real envelope of 7×7 MCPC signal for phase sequence order 1,234,567 which exhibits minimum PMEPR. For 7×7 MCPC signal, totally $7! = 5040$ ways of ordering is possible. But least PMEPR is obtained for continuous ordering of phase sequences such as 1,234,567, 2,345,671 and 7,123,456. Signals with least PMEPR provide larger sidelobe levels in ACF as shown in Fig. 2(b) which results in poor detection. Signals with high PMEPR result in smaller sidelobe levels. The sequence 1234567 is considered, as it provides the least PMEPR.

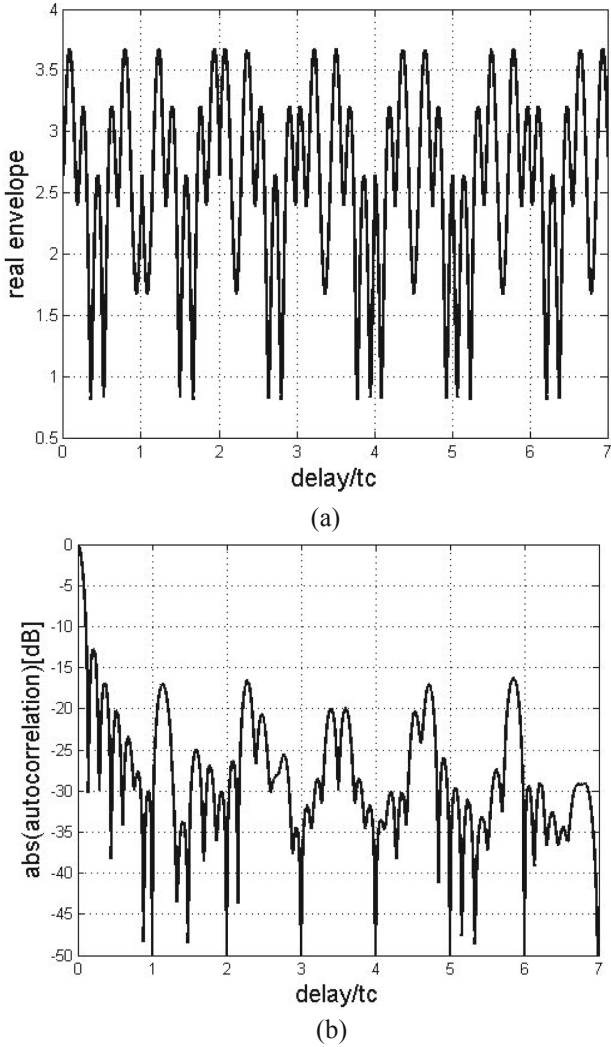


Fig. 2. (a) Real envelope of 7×7 MCPC signal for the order 1234567. (b) Autocorrelation function of 7×7 MCPC signal for the order 1234567

3 MCPC Based on Two Samples Sliding Window Adder (TSSWA) Algorithm

Two Samples Sliding Window Adder algorithm (TSSWA) is applied to single carrier polyphase coded radar signal to reduce the peak sidelobe (PSL) levels. Here an attempt is made to utilize the similar approach to reduce the sidelobe levels for multicarrier signals.

Figure 3 shows the function of TSSWA which is to segregate the input signal using two operations, first is to perform the autocorrelation function of the MCPC signal and

then to delay the autocorrelation function by t_c . Finally, these two signals are added in Stage 1. The above operations are carried out again on the adder output to obtain the final ACF in stage 2 with reduced sidelobe levels. The compressed width after two stages of TSSWA is $3t_c$.

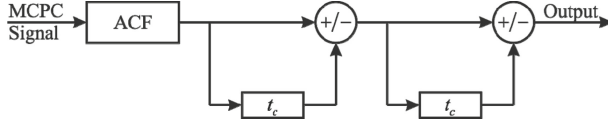


Fig. 3. Block diagram of TSSWA

Figure 4 shows the autocorrelation function of MCPC signals with 3, 5, 7 and 9 carriers using TSSWA. From the plot it can be observed that the sidelobe levels are significantly reduced.

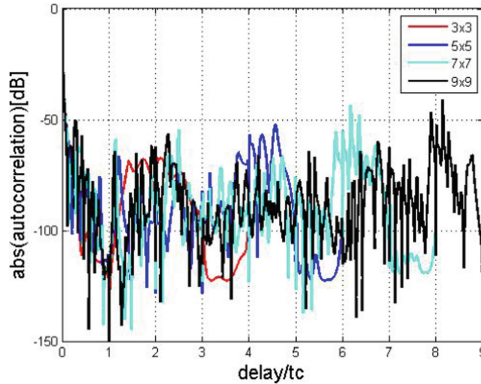


Fig. 4. ACF of 3×3 , 5×5 , 7×7 and 9×9 MCPC signals with continuous order using TSSWA

In TSSWA, operations are not performed on the complex envelope of MCPC signal, rather manipulation is directly done on the autocorrelation function of MCPC signal to obtain clearer distinction between the mainlobe and the sidelobes. Hence this algorithm does not have an impact on PMEPR. It is observed from Table 1 that the sidelobe levels quickly decrease as compared to conventional implementation with no change observed in complex envelope parameters namely PMEPR. The 5×5 MCPC signal developed using P4 polyphase, which is called as conventional, yields 13.70 dB of sidelobe, and similarly 52.37 dB of sidelobe is obtained for the MCPC signal which is developed based on TSSWA method. An improvement in sidelobe level can be observed in the MCPC signal developed using TWSSA technique.

Table 1. Comparison of sidelobe levels of TSSWA with conventional algorithm

Carriers	SL [in dB] (Conventional)	SL [in dB] (TSSWA)
3×3	10.44	67.56
5×5	13.70	52.37
7×7	12.78	43.53
9×9	12.96	40.89

4 Conclusion

In this paper an attempt has been made to optimize the MCPC radar signal performance. The MCPC signal has several merits with respect to bandwidth efficiency and pulse compression capability when compared to conventional radar signals which makes it more suitable for radar applications. In this literature an endeavour is made to improve the MCPC radar signal performance using TSSWA method in reducing sidelobe levels. Numerical results show an improvement in reduction of sidelobe levels using TSSWA based MCPC signal for 3×3 , 5×5 , 7×7 and 9×9 subcarriers. The enhanced complex envelope with zero null points and low autocorrelation sidelobes is obtained.

References

1. Levanon, N., Mozeson, E.: Radar Signals. Wiley, Hoboken (2004)
2. Hwang, T., Yang, C., Wu, G., Li, S., Li, G.Y.: OFDM and its wireless applications: a survey. *IEEE Trans. Veh. Technol.* **58**(4), 1673–1694 (2009)
3. Levanon, N.: Multifrequency complementary phase-coded radar signal. *IEEE Proc. Radar Sonar Navig.* **147**(6), 276–284 (2000)
4. Raghavendra, C.G., Bhat, N.K., Raghu Srivatsa, M.P., Murthy, R.N., Vignesh Nayak, P., Prasad, N.N.S.S.R.K.: A novel approach to generate OFDM radar signals. In: International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques, ICEECCOT 2016, December 2016, pp. 141–145. IEEE (2016)
5. Raghavendra, C.G., Sriranga, R., Rao, S., Nadig, S., Vinay, M., Prasad, N.N.S.S.R.K.: An improvement to the multicarrier complementary phase-coded radar signal using signal distortion technique. In: Sa, P.K., Bakshi, S., Hatzilygeroudis, I.K., Sahoo, M.N. (eds.) *Recent Findings in Intelligent Computing Techniques*, vol. 2, pp. 237–247. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8636-6_25
6. Sato, R., Sasase, I., Shinriki, M.: Time sidelobe reduction technique with small S/N loss for binary-coded pulse compression. In: International Conference on Radar Proceedings, October 2001
7. Li, D., Dai, X., Zhang, H.: Sidelobe suppression in NC-OFDM systems using constellation adjustment. *IEEE Commun. Lett.* **13**(5), 327–329 (2009)
8. Khan, H., Yoo, S.-J.: Active interference restriction in OFDM-based cognitive radio network using genetic algorithm. In: International Conference on Information and Communication Technology Convergence (ICTC), NY, pp. 840–842 (2015)
9. Krebstschmer, F.F., Lewis, B.L.: Doppler properties of polyphase coded pulse compression waveforms. *IEEE Trans. AES* 10–16 (1983)



A Review and Case Study on Attacking and Security Tools at Application-Layer of IoT

Ankit Sinha¹, Sachin Kumar¹, Preeti Mishra^{1(✉)}, Umang Garg², and Arpit Agwarwal¹

¹ Department of CSE, Graphic Era University, Dehradun, India

² Department of CSE, Graphic Era Hill University, Dehradun, India

Abstract. Internet of Things (IoT) is a revolution in our daily life ranging from tiny devices to large industrial setup. Since the last decade, the proliferation of IoT has raised major security concerns which are not often considered by the manufacturers and the end-users as well. IoT devices are more vulnerable to attacks due to weak passwords, lack of standard architecture, availability, and plug and play services, etc. Most of the existing IoT security literature primarily focus on network layer security. However, this paper mainly focuses on the application layer security of IoT for secure cloud-based sensor data handling. We provide a detailed taxonomy of various attacking and security tools at the application layer of IoT. We also provide a comparative analysis of these tools. Finally, a Denial of Service (DoS) case study is performed in the simulated IoT testbed environment integrated with Amazon Web Services (AWS)-IoT using open source tools. We hope that our work will be helpful to the researchers working in the area of IoT security.

Keywords: IoT security · DDoS attack · Attacking tools · Security tools

1 Introduction

The worldwide expansion of the internet and its availability to more than 4 billion end-users has given exponential rise to the IoT. The term IoT refers to anything which has sensors and actuators being managed through controllers over the internet. By 2020, around 50 billion IoT devices are expected to be deployed in the world for almost every possible service available [1]. Recently hackers leaked more than 5 billion passwords of routers, servers, or IoT devices [2]. It raises the strong concern for incorporating security solutions in the IoT environment. Various security vulnerabilities are associated with IoT such as data privacy, physical security, information security, and many more. Attackers exploit the vulnerabilities and launch potential attacks such as Distributed Denial of Service (DDoS), data theft, spam, traffic eavesdropping, jamming, etc. [3].

To perform various attacks, there exist attacking tools [4] such as Low Orbit Ion Canon (LOIC), High Orbit Ion Canon (HOIC), R-U-Dead-Yet (RUDY), GoldenEye, etc. which targets specific layer of IoT such as perception layer, network layer, and application layer [5]. These tools may target an IoT device and may try to increase its CPU utilization, memory consumption or the bandwidth depletion. In order to deal with such attacks, various security tools have been developed. However, there is no one security tool that can detect all possible IoT attacks. As IoT is expanded along with industry 4.0 and the fourth revolution of industries provide many security tools with AWS IoT Core, Azure IoT hub, ZAP, etc. [6] to protect, manage and monitor the IoT-based attacks.

The perception layer of IoT consists of sensors and actuators which gather information by interacting with the physical environment. Some powerful attacks can be executed by attackers at this layer such as physical node tempering, RFID tag cloning, de-packaging, reprogramming of Micro-controller, power analysis, side-channel attack [7]. The network layer is responsible to provide communication between the physical devices and cloud services or the end-user platform. The network layer security ensures security against traffic examining and passive monitoring. It is highly prone to man-in-the-middle attacks which can be followed by eavesdropping [8]. The Jamming attack, Sinkhole attack, Traffic analysis attack, MiTM attacks, etc. are the major attacking possibilities that can occur at this level [7].

The application layer can be considered as the front-end for the IoT system which processes the data retrieved from the cloud servers through the Internet. This layer does not have any common standard framework due to the heterogeneous behavior of IoT [9]. There are several attacks possible at the application layer such as HTTP flood, reverse engineering, DDoS, reprogramming, etc. [10]. To handle these attacks, application-layer security tools are designed to detect and prevent the re-occurrence of such attacks. Several research efforts have been made by researchers to provide security against IoT attacks.

In this paper, we provide a detailed taxonomy of various attacking and security tools for the application layer of the IoT security framework. We also provide a comparative discussion of these tools. At the end, a case study is performed for the HTTP-DoS attack using HULK [10], targeting IoT Bots in the IoT testbed environment. The sensors are simulated using Node-Red tool [11] which is sending data to AWS IoT Core service. The knowledge of attacking tools is very helpful for researchers to create the attack data-sets. In addition, knowledge about security tools is very helpful for understanding the existing solutions. The major contributions of the work can be summarizing as follows:

- To provide a taxonomy and detailed description of various application layer attacking tools.
- To provide a taxonomy and detailed description of various application layer security tools.
- To provide the comparative description of various application layer attacking and security tools.

- To demonstrate a case study of HTTP-DoS attack targeting IoT devices using open source tools.

The rest of the paper is organized as follows: Sect. 2 provides the detail of the related work that has been done in the field of IoT security. Section 3 gives the taxonomy of attacking and security tools. Section 4 and 5 provides a detail description of various attacking and security tools. Section 6 provides the case study for the attacking tool of IoT. Finally, Sect. 7 concludes the paper.

2 Related Work

Recently, the security and privacy issues for the IoT have attracted a lot of research interests. Serrano et al. [12] investigated several privacy issues and provided ranking to them based on their threat level. To provide ranking to the threats open web application security project (OWASP) RRM methodology adopted by authors which is a very popular method and applicable in various tools. Ziegeldorf et al. [13], provided a state of the art survey for threat analysis and privacy issues in IoT. According to the authors, threats can be categorized in seven ways like life-cycle transition, identification, localization, profiling, privacy violation interaction, inventory attack, and linkage. Deogirikar et al. [14] studied a different kind of attacks, classify them according to the layers of IoT, and countermeasures.

As in many other security application domains [15], data mining, and machine learning techniques have been widely adopted in the domain of IoT security and IoT traffic analysis. One of the key challenges is the data privacy problem, especially in collaborative and cloud-based learning scenarios. Several recent studies have proposed novel data privacy-preserving approaches for addressing the problem [16]. According to the best of our knowledge, there is no survey available on attacking and security tools at the application layer. So, we try to categorize different tools for the application layer in IoT. We also implement a case study and display the result that can illustrate the impact after the attack.

3 Taxonomy of Attacking and Security Tools at Application Layer

In this section, a taxonomy of various attacking and security tools used in the application layer has been discussed (as shown in Fig. 1). Application layer tools can be classified into two categories i.e. attacking tools and security tools. The tools which cause any disruption or damage to an application are placed in the attacking category whereas the tools used to detect or prevent malicious activities are placed in security tools. According to the taxonomy discussed, the attacking tools can be used to trigger application-layer attacks like HTTP floods, which send seemingly legitimate requests, but, in vast quantities. On the other hand, the security tools are used to detect this malicious behavior on the application layer. In the next section, we discuss these attacking and security tools with their functionalities.

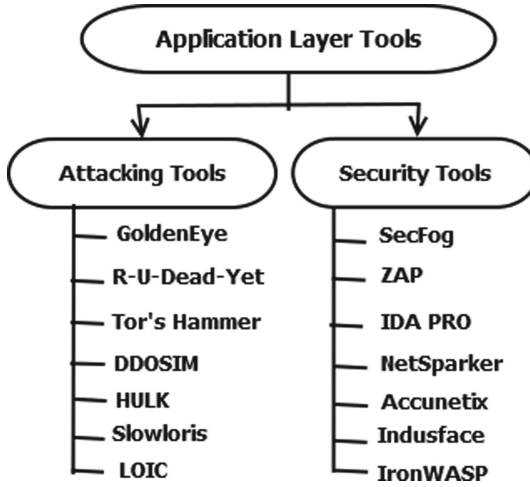


Fig. 1. Taxonomy of attacking and security tools

4 Attacking Tools on Application Layer

In this section, we have discussed various application layer attacking tools that are capable of compromising IoT devices shown in Table 1. These tools are used to attack servers or routers to get unauthorized access or exploit the resources of the target machine. So that the legitimate user can not access the required services. All the necessary information regarding system requirements, availability, interface, etc. has been discussed.

4.1 GoldenEye

GoldenEye [17] is one of the popular HTTP Denial Of Service testing Tools. GoldenEye is the most recommended tool that can investigate the malware environment. GoldenEye uses KeepAlive (connection) paired with the Cache-Control options to preserve the socket connection busting through caching (whenever possible) until it consumes all accessible sockets on the HTTP/S server. This disruption can be done by flooding the targeted server with a huge number of requests which in turn overloads the system and interrupts the actual requests.

4.2 R-U-Dead-Yet

R-U-Dead-Yet [18] is an HTTP post-DOS attack tool that executes slow rate attack. This attack is a low and slow attack that can attack some connections to the target server for a short span of time and leave the session as long as possible. The POST method is used to performs a DOS attack with a long-form field submission. The interactive console menu comes with this tool. It detects forms for a given URL and lets users select which forms and fields should be used

for a POST-based DOS attack. The length and number of sessions can target to the resources and make it unavailable for legitimate users.

Table 1. Taxonomy of attacking tools

Attacking tools On Application Layer						
Tool name	Type of tool	Expertise required	Language	Cost of service	Interface	Year
Golden-Eye [17]	DoS (HTTP)	No	Python	Free	Both	2014
RUDY [18]	DoS (HTTP)	No	Python	Free	GUI	NA
Tor's Hammer [19]	Dos (TCP, HTTP)	Yes	Python	Free	Both	2014
DDOSIM [20]	DoS (TCS, HTTP)	No	C++	Free	GUI	2009
HULK [10]	DoS (TCS)	No	Python	Free	Both	2012
Slowloris [21]	DoS (HTTP)	Yes	Python	Free	CLI	2015
LOIC [22]	DoS (TCS, UDP, HTTP)	Yes	C++, C sharp	Free	GUI	2008

4.3 Tor's Hammer

Tor's Hammer [19] is another slow rate HTTP post-Dos testing tool introduced by phiral.net. Tor's Hammer is a leisurely post tool and it is written in Python. In this attack, an HTML post field is transmitted within a session using slow rate transmission. Tor client uses a native socks proxy to carried out in the Tor network. An extra advantage of this tool is that: During the attack, It can be run through a TOR network to be unsigned. It is a productive tool that can kill Apache or IIS servers in a few seconds. This attack is difficult to track due to the slow traffic transmission of data.

4.4 DDOSIM

DDOSIM (DDoS Simulator) [20] is another popular DOS attacking tool. As the name suggests, DDOSIM is used to execute DDOS attacks, simulates many zombie hosts with random IP address established a full TCP connection for the target server. When the connection is complete it starts the conversation with accepting packets from an application such as the HTTP server. DDOSIM is written in C++ and runs on Linux systems. It can perform HTTP DDoS attack with a valid request or invalid request, SMTP DDoS, and TCP connection flood using a random port.

4.5 HULK (HTTP Unbearable Load King)

HULK [10] is a kind of web server that can execute a DoS attack. Hulk is created to generate the volume of uncommon traffic at a web server to bypass

the caching and try to hit the server's resource pool directly. HULK generates a unique pattern for each request and for each generated request to jumbled traffic at a web server. To avoid attack detection via known patterns hulk uses many other techniques. Hulk can bypass caching engines and also uses referrer forgery. The developer of the tool tested it on an IIS 7 web server with 4 GB RAM. This tool brought the server down within one minute.

4.6 Slowloris

Slowloris [21] is a type of DoS attack tool that can allow the attacker to overwhelm a target server by sending several HTTP requests. Slowloris is primarily an HTTP Denial of Service attack that affects threaded servers. It makes an effort to keep numerous connections at a time with the target web server and tries to keep open as long as possible. The target server opens a thread for each request and try to close or timeout the thread if it takes too long. But the attacker sends partial requests in order to keep the connection alive, so that the target server is not able to close the connection and unable to respond new request.

4.7 LOIC

LOIC [22] is a powerful dos (denial of service) attacking tool that can send a large sequence of UDP, TCP or HTTP requests by the URL of the target server. This tool provides a GUI interface that contains several options such as the IP address of the target device, timeout, port, TCP/UDP message, and threads. One attacker can't stop the services of the target server but if the number of attackers work simultaneously for the same target then it would be more serious. In order to simultaneous attack, users can connect with the IRC (internet relay chat) and this makes the user machine part of a botnet.

5 Security Tools on Application Layer

IoT devices are more vulnerable due to a lack of standardizing security framework and protocols. So there are some tools that can overcome the security issues generated by attackers. In this section, we have discussed several security tools that are capable to provide security at the application layer (as shown in Table 2).

5.1 SecFog

SecFog [23] is a positive methodology that can be used to assess the level of security for multi-layered applications. It helps the providers to find out which application is the most secured application. SecFog plays two important roles for the users: Application operator (Deployment of designing and managing applications) and Infrastructure operator (Managing target cloud-edge nodes). SecFog is written in the ProbLog2 language for security level assessment.

Table 2. Taxonomy of security tools

Security tools on application layer						
Tool name	Type of service	Expertise required	Language	Cost of service	Interface	Year
SecFog [23]	Multii service	Yes	ProLog	Free	GUI	2013
ZAP [24]	Web application	No	Java	Free	GUI	2012
IDA Pro [25]	Web application	No	Python	Both	Both	2005
Netsparker [26]	Web Application	Yes	SAML	\$ 4,999/yr (std.)	GUI	2009
Acunetix [27]	Web application	Yes	HTML5, Javascript	14 day trial	GUI	2005
Indusface [28]	Web application	No	PHP, C++	30 day Free trial	GUI	2012
IronWASP [29]	Web application	YES	Python /Ruby	Free	GUI	2013

5.2 Zed Attack Proxy Project (ZAP)

The OWASP Zed Attack Proxy (ZAP) [24] is the most popular open-source security tool for penetration testing of web applications. It is used to find security vulnerabilities in web applications automatically across all operating systems. ZAP creates a proxy server and works as a scanner or security vulnerability finder for the traffic that passes through the proxy. ZAP incorporates proxy intercepting features such as spider, fuzzing, web socket testing, active scan, or alerts, etc.

5.3 IDA PRO

IDA Pro (Interactive Disassembler) [25] is a tool with various features like cross-platform, multi-processor disassembler, and debugger. This tool can translate machine-executable code to assembly code with the purpose of reverse engineering. If we choose a new file to disassemble, it can recognize the file format, processor type, and executable architecture. IDA Pro is also used to inspect server binaries to determine the exploitation of any web server.

5.4 Netsparker

Netsparker [26] is an automated scanner tool that will recognize vulnerabilities such as SQL Injection and Cross-site Scripting in web applications and in web APIs. It lessens the vulnerability finding time ratio and is available as a Windows software with online service. Netsparker provides several features like accurate reports with proof scanning, scan we asset, advanced web scanning and crawling, and automated vulnerability triage and management.

5.5 Acunetix

Acunetix [27] is a fully self-regulating web vulnerability scanner and testing tool. Acunetix is used to detect and report on over 4500 web application weaknesses

for all variants of SQL Injection, cross-scripting and XSS are included. It can also scan open-source software and custom-built applications with Vulnerability Management and easily review vulnerable data and generate multiple technical and compliance reports.

5.6 Indusface

Indusface [28] provides both manual Penetration testing bundled with its own automated web application vulnerability scanner and includes a Website reputation check of links, malware and defacement checks of the website in every scan. Every customer who gets a Manual PT done automatically will also get the automated scanner and can use that automated scanner on-demand for the whole year. It provides many features such as unlimited app scanning, malware monitoring, blacklisting detection, etc.

5.7 IronWASP

IronWASP (Iron Web application advanced security testing platform) [29] is an open-source scanner creator for web application using python scripting. It is fully automated scanning with enhanced cloud variant of Sboxr DOM and continuous scans the site and alerts to the server for new issues discovered. It is a GUI based tool that can generate reports in HTML and RTF formats. It can detect various vulnerabilities such as XSS (Cross-Site Scripting), Cross-Site Flashing, CSRF (Cross-Site Request Forgery), Click-jacking, Open URL Redirection, etc.

6 Case Study

In this section, we are dealing with an open-source attacking tool HULK [10] that can perform DoS attack [30] by generating the volume of traffic at a web server and bypass the caching. We also display the results in terms of memory utilization after attacking scenario.

6.1 Experimental Test-Bed Details

To implement the attacking scenario (as shown in Fig. 2 and Fig. 3), we have used a system having a main memory of 8 GB and a hard disk storage of 500 GB. The CPU of the system belongs to the family of 5th generation Intel Core i5. We have installed VMware ESXi 6.7 type 1 hypervisor that can be used as a host. The host has two virtual machines, one is the Ubuntu Server and the other is Ubuntu VM.

In order to simulate the network behavior of IoT devices, we have installed the Node-Red tool [11] in the target Ubuntu server. Node-Red is an open-source GUI tool and acts as a middle-ware agent which can connect the physical IoT devices with the cloud applications. In the Node-Red tool, we have developed a weather station simulator that imitates the IoT sensors such as temperature

and humidity sensors and installed MQTT broker in it. The simulator is then connected to Mosquitto MQTT broker to connect with AWS IoT core, while running the test-bed environment. For this, we configured the broker with some valid certificates and keys generated from the cloud IoT service.

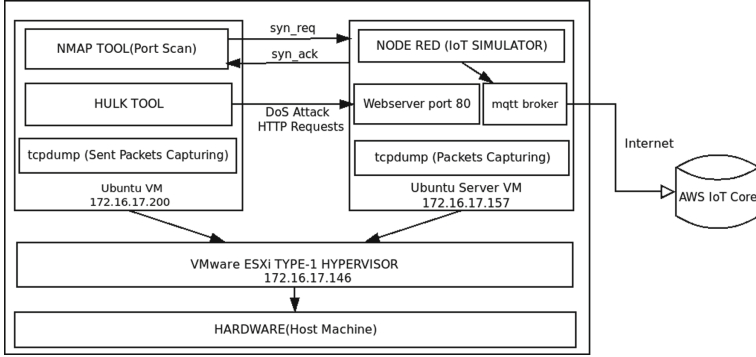


Fig. 2. Experimental test-bed

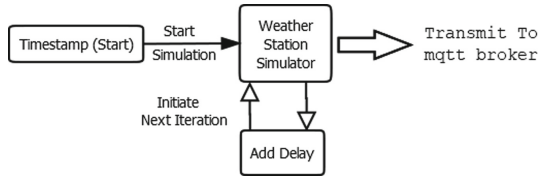


Fig. 3. Flowchart of weather station simulation

6.2 Attack Scenario

On the Ubuntu VM, Network Mapper (Nmap) [31] has been installed for port scanning. Nmap is an open-source network scanning tool, which offers a number of features including vulnerability and port scanning. Further, We have cloned the HULK tool [10] from GitHub and unzipped it. On the Ubuntu server VM, we have installed TCP-dump for capturing packets of the network traffic and the Node-Red [11] is started by typing *Node-Red* in the terminal and also the Node-Red is opened in the local browser where the simulator is connected to AWS public IoT hub through the MQTT broker.

6.3 Execution Steps

Step 1: N-Map Port Scanning

For running the DDoS attack, it is necessary to have port 80 open on the target machine (as shown in Fig. 4). For this, N-map tool has been used on the Ubuntu VM using the IP of the Ubuntu server.

```
ubuntu@virtual_vm0:~$ nmap 172.16.17.157
Starting Nmap 7.60 ( https://nmap.org ) at 2020-01-20 19:04 IST
Nmap scan report for 172.16.17.157
Host is up (0.0011s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    closed ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 17.17 seconds
```

Fig. 4. nmap port scanning

Step 2: Enable firewall

If port 80 is closed then it is required to enable the firewall using the uncomplicated firewall (ufw) command in the Ubuntu server that is *sudo ufw enable* and allow the required port *sudo ufw allow 80* or *sudo ufw allow http*. This will allow the incoming HTTP traffic on port 80.

Step 3: Launching Attack

After navigating to the cloned folder of HULK tool, the attack has been launched by executing its python file as shown in Fig. 5. In the command *sudo python hulk.py http://172.16.17.157:80*, '172.16.17.157' is the private IP address of the Ubuntu server and ':80' specifies the port number.

```
ubuntu@virtual_vm0:/home/sachin/Desktop/hulk-master$ sudo
python hulk.py http://172.16.17.157:80
[sudo] password for ubuntu:
-- HULK Attack Started --
22806 Requests Sent
22927 Requests Sent
23200 Requests Sent
23397 Requests Sent
23549 Requests Sent
23698 Requests Sent
```

Fig. 5. Capture the network traffic

Step 4: Network Traffic Analysis

While running the Hulk tool [10] from the Ubuntu VM having private IP address '172.16.17.200', we have executed tcpdump command on the victim machine i.e. the Ubuntu server in order to capture and analyze the network traffic (as shown in Fig. 6). The command *sudo tcpdump -i ens160 port 80* includes the interface (i.e. 'ens160') and the port number (i.e. '80'). The incoming packets from the

```

userver@userver-vn:~$ sudo tcpdump -i ens160 port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), capture size 262144 bytes
16:50:10.261728 IP 172.16.17.200.41546 > a23-32-176-32.deploy.static.akamaitech
nologies.com.http: Flags [S], seq 2589036777, win 29200, options [mss 1460,nop,
nop,sackOK,nop,wscale 8], length 0
16:50:10.261957 IP a23-32-176-32.deploy.static.akamaitechnologies.com.http > 17
2.16.17.200.41546: Flags [S.], seq 733811084, ack 2589036778, win 29200, option
s [mss 1460,nop,nop,sackOK,nop,wscale 7], length 0
16:50:10.262037 IP 172.16.17.200.41546 > a23-32-176-32.deploy.static.akamaitech
nologies.com.http: Flags [F.L], ack 1, win 115, length 0

```

Fig. 6. Analysis the network traffic

attacker machine's IP can be easily captured. The HTTP requests flooded the victim's web server and can be observed from the packet capturing step. This makes it slow to respond to the intended clients, due to which a certain delay is observed in updating of the sensors' data on the cloud platform.

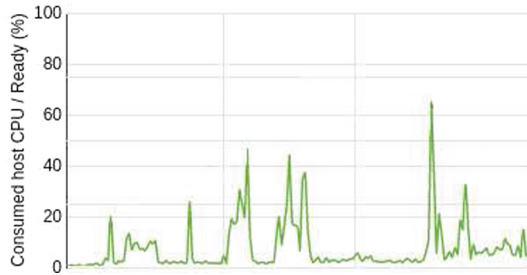


Fig. 7. Memory utilization of victim IoT-BoT before attack



Fig. 8. Memory utilization of victim IoT-BoT after attack

Step 5: Validation

After the execution of DoS attack, there will be a sudden increase in CPU usage (as shown in Fig. 7 and Fig. 8). If memory utilization increases of the victim machine IoT-BoT then it may affect to the other systems. There are several security tools and techniques to prevent such kind of attacks.

7 Conclusion

IoT devices are cover almost every possible field that reduces human intervention. Despite having many advantages, security issues in IoT devices still remain a major concern. In this paper, we have proposed the taxonomy of various application layer attacking and security tools. Further, we have described some of those tools and provide a comparative analysis among them. As the understanding of their working of attacking mechanisms is important to develop a secure model for IoT devices. The case study involves the DoS attack which has been performed on a virtually simulated IoT device in order to analyze the system's performance after the attack. In the future, the analysis work can be expanded for tools working on the rest of the other layers of an IoT architecture.

References

1. Vyas, D.A., Bhatt, D., Jha, D.: IoT: trends, challenges and future scope. In: IJCS 2015, vol. 7, pp. 186–197 (2015)
2. ZDNet, Massive Leaks. <https://www.zdnet.com/article/hacker-leaks-passwords-for-more-than-500000-servers-routers-and-iot-devices/>. Accessed 20 Jan 2020
3. Sicari, S., Rizzardi, A., Grieco, L.A., Coen-Porisini, A.: Security, privacy and trust in Internet of Things: the road ahead. *Comput. Netw.* **76**, 146–164 (2015)
4. InfoSec Skills, Attacking Tools. <https://resources.infosecinstitute.com/dos-attacks-free-dos-attacking-tools/gref>. Accessed 5 Jan 2020
5. Ammar, M., Russello, G., Crispo, B.: Internet of Things: a survey on the security of IoT frameworks. *J. Inf. Secur. Appl.* **38**, 8–27 (2018)
6. G2 Group, Security Tools. <https://www.g2.com/categories/iot-security>. Accessed 15 Dec 2019
7. Grammatikis, P.I.R., Sarigiannidis, P.G., Moscholios, I.D.: Securing the Internet of Things: challenges. *Threats Solut.* **5**, 41–70 (2018)
8. Tao, F., Cheng, Y., Da Xu, L., Zhang, L., Li, B.H.: CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. *IEEE Trans. Ind. Inform.* **10**, 1435–1442 (2014)
9. Burhan, M., Rehman, R., Khan, B., Kim, B.-S.: IoT elements, layered architectures and security issues: a comprehensive survey. *Sensors* **18**, 2796–2812 (2018)
10. Abdmeziem, M.R., Tandjaoui, D., Romdhani, I.: Architecting the internet of things: state of the art. In: *Robots and Sensor Clouds*, pp. 55–75 (2016)
11. IBM's Emerging Technology Services team, Node-Red. <https://nodered.org/>. Accessed 18 Dec 2019
12. Serrano, J., Muñoz, J.L., León, O., Mikkelsen, L., Schwefel, H., Bröring, A.: Privacy risk analysis in the IoT domain. In: *2018 Global Internet of Things Summit (GIoTS)*, pp. 1–6 (2018)
13. Ziegeldorf, J.H., Morchon, O.G., Wehrle, K.: Privacy in the Internet of Things: threats and challenges. In: *Security and Communication Networks*, vol. 7 (2014)
14. Deogirakar, J., Vidhate, A.: Security attacks in IoT: a survey. In: *I-SMAC*, pp. 32–37 (2017)
15. Meng, W., Wang, Y., Wong, D.S., Wen, S., Xiang, Y.: TouchWB: touch behavioral user authentication based on web browsing on smart phones. *J. Netw. Comput. Appl.* **117**, 1–9 (2018)

16. Li, P., Li, J., Huang, Z., Gao, C.-Z., Chen, W.-B., Chen, K.: Privacy-preserving outsourced classification in cloud computing. *Clust. Comput.* **21**, 277–286 (2018)
17. Seidl, J.: Goldeneye. <https://github.com/jseidl/GoldenEye>. Accessed 10 Dec 2019
18. Orbital Satellite. R-U-Dead-Yet. <https://sourceforge.net/projects/r-u-dead-yet>. Accessed 18 Nov 2019
19. Solar Stone, Tor’s Hammer. <https://sourceforge.net/projects/torshammer>. Accessed 20 Nov 2019
20. Furtuna, A.: DDOSIM. <https://sourceforge.net/projects/ddosim>. Accessed 25 Nov 2019
21. Yaltirakli, G.: Slowloris. <https://github.com/gkbrk/slowloris/blob/master>. Accessed 27 Nov 2019
22. Abatishchev: LOIC. <https://sourceforge.net/projects/loic/>. Accessed 28 Nov 2019
23. Stefano: Secfog. <https://github.com/di-unipi-socc/SecFog>. Accessed 28 Nov 2019
24. OWASP: Zed Attack Proxy Project (ZAP). <https://www.owasp.org/index.php>. Accessed 30 Nov 2019
25. Hex-Rays: IDA Pro. <https://www.hex-rays.com/products/ida/>. Accessed 30 Nov 2019
26. Chen, S.: Netsparker. <https://www.netsparker.com>. Accessed 10 Dec 2019
27. Acunetix. <https://www.acunetix.com/penetration-testing>. Accessed 15 Dec 2019
28. OWASP: Indusface. <https://www.indusface.com/products>. Accessed 20 Dec 2019
29. Kuppan, L.: IronWASP. <https://sboxr.com>. Accessed 22 Dec 2019
30. Tanenbaum, A.S., Van Steen, M.: *Distributed Systems: Principles and Paradigms*. Prentice-Hall, Upper Saddle River (2007)
31. Lyon, G.: nmap. <https://nmap.org/>. Accessed 20 Dec 2019



Accident Prevention of Automobile Using Real-Time Tracking System

S. B. Rudraswamy, M. G. Pruthvi, Sameera Fatima, Sneha Jangamashetti^(✉),
and M. Sathya

JSS Science and Technology University (Formerly SJCE) Mysore, Mysore, Karnataka, India

Abstract. The advancement in technology has provided a lot of ease and solved many safety concerns in various fields. It had also catered various applications in automotive sector and has the capability to solve safety issues in order to prevent the occurrence of accidents to a large extent. Accident Prevention of Automobile using Real-time Tracking (APART) system is one such system which uses the real time tracking of all the vehicles and provides a map view containing real-time plots of all registered vehicle to each of the drivers, So that the driver will have prior knowledge of vehicles in its vicinity to avoid the accidents which would predominantly occur at the junctions, fog areas and blind curves, moreover it also increases the fuel efficiency by reducing the braking and accelerating instances. This paper presents a complete overview of APART system comprising of a raspberry pi and a GPS module to send the location to the real-time database and a map development process by using Google API key in order to display the resultant map.

Keywords: Blind junctions · Fog · Real-time · Multiple tracking · GPS module · Database · REST API · Flask

1 Introduction

Transportation has become a basic necessity in almost every person's life and so are the numbers of commuters increasing day by day. Due to this, the density of traffic on roads is increasing and also the number of accidents. According to the survey most of the accidents occur at the junctions, blind curves and also due to the unfavorable environmental condition such as heavy rains, fog, and smog. These accidents lead to loss of lives in severe cases and also create traffic jams on roads for hours to days, consequently resulting in loss of one's valuable time. So in order to prevent these accidents we have devised a solution to educate the driver with the live movements of his surrounding vehicles, so that he can have prior knowledge of vehicles nearby to take a right decision at the right time.

According to the statistical data, it has been seen that there is sharp increase in the number of accidents occurring at the sites where the driver's field of view is constrained due to different road architectures, unfavorable geographical and environmental conditions. The accidents occurring at the junctions like the T-junctions, Y junctions, four arm

junctions, staggered junctions, round-about were 1,75,853 in 2017 of which 1,24,024 accidents took place at uncontrolled junctions which had no traffic signals or police control, this accounts to 73.3 per cent of fatality at road junctions, this went up to 2 lakhs in the year 2018 [1]. The accidents due to the thick fog which was recorded to be 5886 in the year 2014 and it increased to 11,090 in the year 2018 [2]. Our paper is devised to avoid the occurrence of these accidents.

1.1 Problem Statement

To design and implement an APART device to avoid accidents at cross-junction, blind curve and fog region due to unpredicted approach of vehicles.

1.2 Related Work

According to various paper researches many measures have been taken to avoid accidents of vehicles. Wireless sensors [4, 5] such as RF transmitter, RF receiver [7], Radar [9], ultrasonic sensors, Li-fi technology which will be deployed near the cross junctions to detect the movement of vehicles, transmitter will continuously transmit and receivers will respond to the transmitted signal. But a major drawback in using wireless sensors is that they may get damaged in adverse weather conditions, and even miss the count of vehicles when more than two vehicles passes by simultaneously. But since our APART device discretely sends the location updates from each vehicle, this shortcoming can be overcome. GPS [3] was used in few research papers which collects the current latitude and longitude position of the moving vehicle and sends it through wireless communication link using ZigBee [5] or GSM [6] module but it is only restricted to a local area network. Our system is globally connected and has the capability of tracking multiple locations of all the connected vehicles and displays user's vehicle with blue indicator and the neighboring vehicles with a green indicator.

Few papers adopted image processing methods using cameras and FEM model simulation [10] to detect vehicular movements at the blind curves, but these captured images cannot be relied on during night times and bad weather conditions like fog, dust, smoke, etc. APART system overcomes this shortcoming by using GPS to fetch locations irrespective of the weather conditions. Artificial intelligence algorithm [8] was used to differentiate between pedestrians and vehicles based on their speeds.

As research is going on self-driving cars which consist of large number of sensors and GPS connected with internet, but such vehicles cannot be afforded by everyone. So our device efficiently overcomes this shortcoming as well and can be installed in all types of vehicles without making any significant alteration.

1.3 Objective

- Main objective of this paper is to provide assistance to the driver to have a safe journey, it gives the topological view of the region to the driver, as well as marks the position of the vehicles in its vicinity on the map, so that the driver gets to know about all the vehicles arriving from places which are out of his field of vision.

- Increase in the fuel efficiency: On sudden breaking or acceleration of the vehicle there is a high fuel consumption. But, by using this device the need of sudden breaking is almost eliminated as the driver can take decision beforehand.

2 Proposed System

APART device will be installed in every vehicle. The live location updates from this device will be collected and stored in the real-time database from all the vehicles. Once the server is invoked, this data will be processed and marked on the map by the Data processing unit, with the help of Google map API Key. All the locations are marked on the same map as shown in the Fig. 1. This map will be sent back to the user with the help of DPU and Buffers. At the user end there will be a display unit to display the map to the user. Every user will get a map focused around his location. In order to distinguish the user from the rest of the users on the maps, Blue identifier is given to the user and the neighboring vehicles are represented by Green indicator.

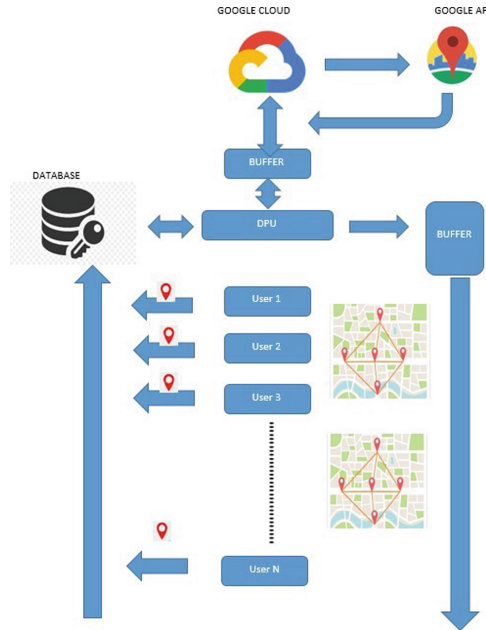


Fig. 1. Overview of APART System

3 Implementation

APART device is designed to be installed in existing vehicles as well as the upcoming vehicles. The implementation of the system basically involves two major processes, namely, registration of the user and real-time location updating from the device.

3.1 Registration

To ensure security and user location confidentiality a registration process is designed which takes care of authorization and authentication of each user. Initially the user needs to enter his name, email address, vehicle number and vehicle type in the registration page as shown in the Fig. 2, after which a unique user ID will be sent to the user which has to be entered in the proceeding login page for verification as shown in the Fig. 3.

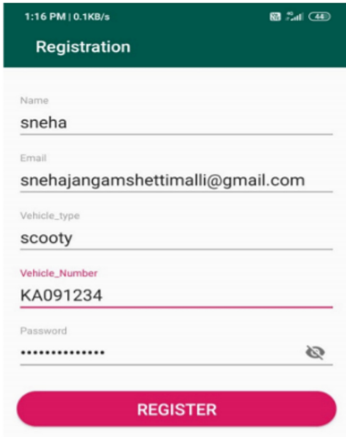


Fig. 2. Registration of an apart device

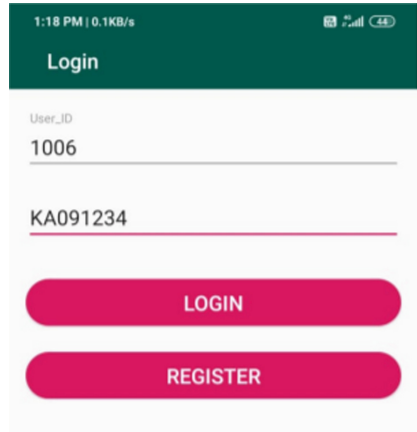


Fig. 3. Login page

After successful login, the device starts updating the live location to the created database under the respective unique id which was previously assigned. The flow diagram of the mentioned process is shown in Fig. 4.

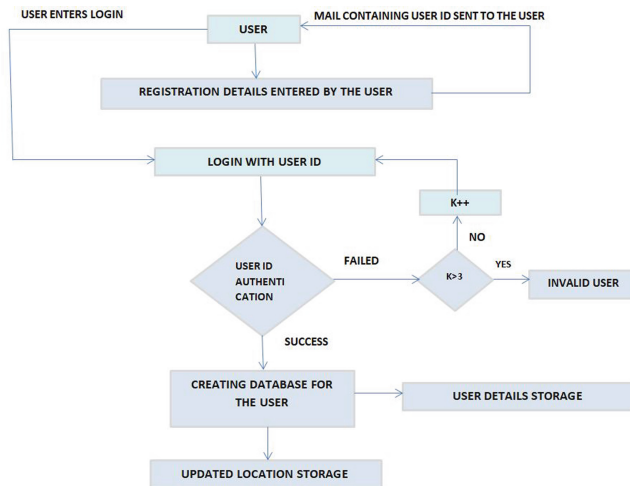


Fig. 4. Steps of registration

At the back end the details entered by the user are stored in the database under a folder created with its name as the unique user ID as shown in the Fig. 5 to ensure user confidentiality in the database. Each folder consists of user and location details.

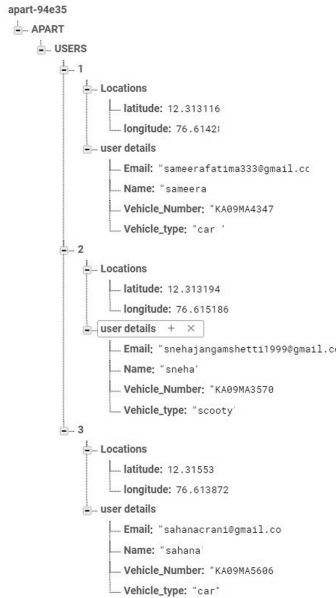


Fig. 5. Real- time database

The amount of memory required for each user is about 200 bytes i.e. we can store 5 users in just 1 KB of memory. For 1 GB of memory we can store up to 5,242,880 users. These figures prove memory storage and maintenance is not a challenging task. Moreover, Firebase provides high flexibility, hence any alterations in the memory can be easily by the root user according to the users.

3.2 Location Updation

APART device consists of a raspberry Pi interfaced with a GPS module as shown in the Fig. 6, The device can work on minimum supply voltage of 5 V with a provision of a constant internet connectivity.

Figure 7 depicts the flow diagram of working of APART device. Firstly, on starting the vehicle, the device is powered ON. The GPS module in the device gets activated and starts sending its location details to the raspberry pi controller which stores it in the database in a compatible format. AWS Server fetches all the location updates from the real time database and converts it into JSON format to ease the transfer of data between files and processes. As the system functions in real time AJAX is used to automatically fetch and update the locations. The locations will be then converted into markers on the Google maps with the help of Google map API. Only a single map is used to display all

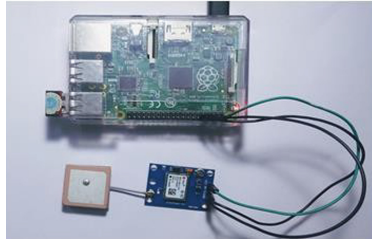


Fig. 6. APART device

the locations present in the database, and then a html page embedding the map is web hosted using a public IP to the registered users which can be accessed by logging in with their user ID.

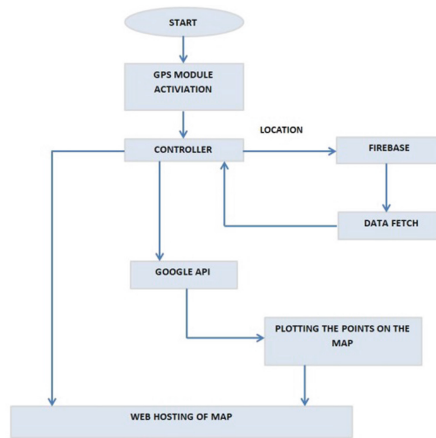


Fig. 7. Flow diagram of working of an apart system

4 Results and Analysis

The device was installed in few vehicles and tested in real time. Figure 8 shows the device installed in a Scooter and an Auto-Rickshaw. When the vehicles starts, these modules gets powered up and start sending its location updates to the server and simultaneously view the real time tracking of vehicles around.

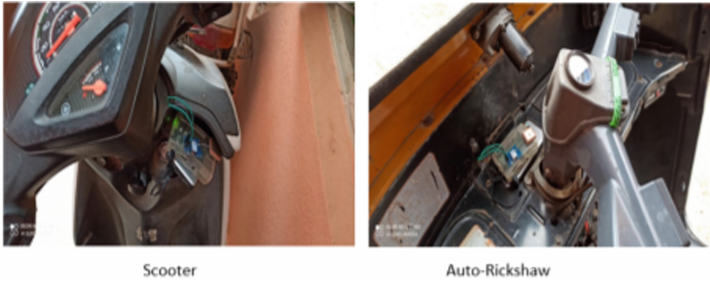


Fig. 8. Modules embedded in the vehicles

The left side of Fig. 9(a), depicts the map view visible to the scooter driver, the center part shows the road view of the scooter driver. And the right part depicts the map view seen in the device installed in auto rickshaw.

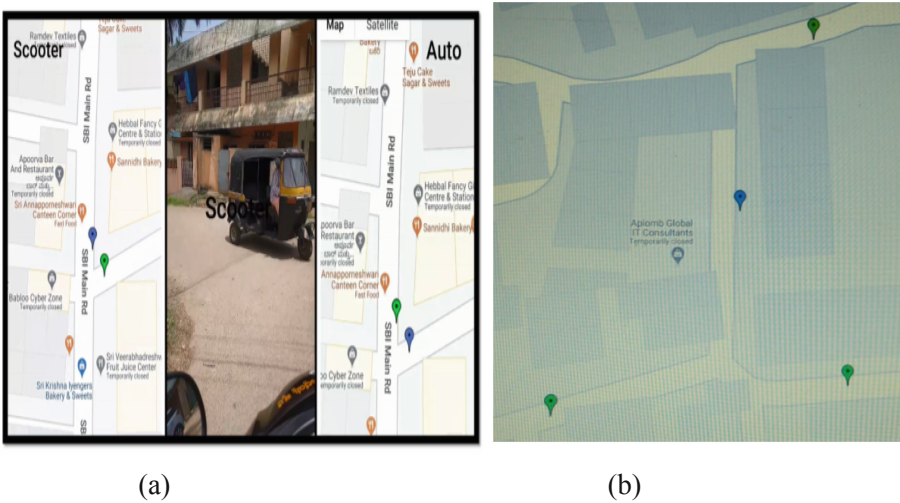


Fig. 9. (a) Real time Scenarios with three view details and location in real time. (b) Map view with multiple location blue mark represents user vehicle and green mark represents neighbor's vehicles

As we can see from the above figure, even though map is provided from the same server the maps are different for each user. In order to distinguish the user from the his neighboring vehicles, Blue mark is given to the user and Green marks for the neighboring vehicles as in Fig. 9(b). Hence the left part and right part of the Fig. 9(a) markers are interchanged.

The important requirement of our system is Internet Connection, as we are concentrating in the regions like Ghats, where the internet connectivity is very low. In addition as we are processing the information in real time, delay matters due to the Internet

speed. Hence, the delay was analyzed for different Internet speeds and estimated the delay factors. In below Table 1 time delay is recorded for different internet speed.

Table 1 Estimation of time delay at different internet speed

Serial No	Internet speed in bps	Time delay in milli seconds
1	1K to 10K	2000
2	20K to 50K	1000
3	100K	300
4	1M	200
5	50M	200
6	100M	200
7	1G	200

The Graph showed in Fig. 10 give the real time delays for different Internet speeds. Here maximum delay recorded was 2000 ms at a speed of 1–10 Kbps. The common speed at many places was in the range of 50Kbps to 500Kbps, for which the delay was negligible of 250 ms-500 ms. Moreover, time required for location updating in firebase was 100 ms. And the time taken to return the map to the user was also 100 ms, which accounts to an additional 200 ms round about time.

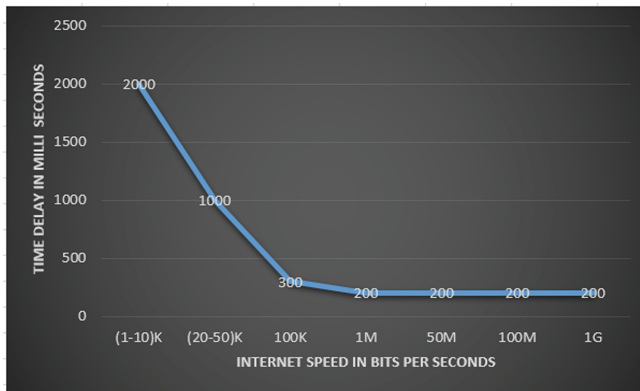


Fig. 10. Internet speed vs. Time delay

5 Novelty of Proposed Work

APART system uses a simple communication pathway which includes the user, Apart server and database, and a GPS module installed in each device fetches the location

coordinates independently, this design eliminates the need for inter-vehicular communication which ensures the privacy of user data and makes it less prone to hacking instances as the users are not connected to each other directly, APART system works on a global scale and doesn't need local base stations functioning at various junctions as deployed in few previous works, APART device is a compact device installed within the vehicle and does not take any physical parameters as its input, so its efficiency is not affected by unfavorable weather conditions or change in the ambient light, this was seen as a major drawback in works which used sensors and cameras at the junctions and blind spots to detect the movement of vehicles whose efficiency decreased during unfavorable conditions. APART device is designed to be very versatile in nature as it not only prevents accidents at junctions but also at the ghat roads, foggy regions and all the places where the drivers field of view is limited and the probability of accidents is high.

In contrast to Google Maps, using which the user can track only his path and get navigation updates, our device will track and display the locations of all the vehicles in its vicinity in real time, which makes it convenient for a driver to be informed about the vehicles arriving from places out of his field of vision and hence prevent accidents.

6 Application

1. This safety device is used to avoid head-on collisions, multi vehicle pile-up which could occur due to the lack of proper view during unfavorable weather conditions like the heavy rains and dense fog
2. Collisions at the junctions are also prevented, as the driver gets to know about the vehicles approaching at the junction.
3. Moreover, overtaking at the highways can be made more safe and effective using this device
4. At night times the device assists the driver at low level of illumination to proceed safe. It can be used in the Ghats roads at steep turns.

7 Conclusion

As the upcoming 5G technology would provide uninterrupted data services at any possible geographical area, our proposed system exploits this capability to assist the drivers to have a safe journey and moreover increase the fuel efficiency. Our device successfully updated all the real time locations of the registered vehicles and provided a map view to the driver using which he could know the arrival of the vehicles from places out of his field of vision, it was tested at the cross junctions which was found to be effective to avoid accidents as it gave results with a very minimal delay.

8 Future Scope

As the GPS signal strength might get reduced near buildings, bridges and very adverse weather conditions, GPS repeaters may be used and implemented in these places to boost the signal strength. Moreover, predictive algorithms like the linear quadratic estimation

(Kalman filtering) algorithm which uses the past speed, time and other parameters to predict the next location of the vehicle can be used to predict the movement of the vehicles for small distances when the vehicle loses connectivity in a tunnel or hilly areas.

References

1. Road Accident occurred at different Junction in India in the year 2017. https://morth.nic.in/sites/default/files/Road_Accidents_in_India_2017.pdf
2. Accidents occurred in fog region. <https://timesofindia.indiatimes.com/india/over-10000-lives-lost-in-fog-related-road-crashes/articleshow/67391588.cms>
3. Anurag, D., Ghosh, S., Bandyopadhyay, S.: GPS based vehicular collision warning system using IEEE 802.15. 4 MAC/PHY standard. In: International Conference on ITS Telecommunications. IEEE, Kolkata (2008)
4. Ramesh, S., Ranjan, R., Mukherjee, R., Chaudhur, S.: Vehicle collision avoidance system using wireless sensor networks. *Int. J. Soft Comput. Eng.* **2**(5), 2231–2307 (2012)
5. Wheeler, A.: Commercial applications of wireless sensor networks using ZigBee. *IEEE Commun. Mag.* **45**(4), 70–77 (2007)
6. Devikiruba, B.: Vehicle speed control system using GSM/GPRS. *Int. J. Comput. Sci. Inf. Technol.* **4**(6), 983–987 (2013)
7. Vengadesh, A., Sekarm, K.: Automatic speed control of vehicle in restricted areas using RF and GSM. *Int. Res. J. Eng. Technol.* **2**(9), 875–877 (2015)
8. Amrith, P., Umamaheswari, E., Anitha, R.U., Mani, D., Ajay, D.M.: Smart detection of vehicle accidents using object identification sensors with artificial intelligent systems. *Int. J. Recent Technol. Eng. (IJRTE)*, **7**, ISSN: 2277-3878 (2009)
9. Liu, G., Wang, L., Zou, S.: A radar-based blind spot detection and warning system for driver assistance. In: Information Technology Electronic and Automation Control Conference (IAEAC), pp. 2204–2208 (2017)
10. Simon, B., Franke, F., Riegl, P., Gaull, A.: Motion planning for collision mitigation via FEM-based crash severity maps. In: Intelligent Vehicles Symposium (IV), France, pp. 2187–2194 (2019)



Attention LSTM for Time Series Forecasting of Financial Time Series Data

Yedhu Shali^(✉), Banalaxmi Brahma, Rajesh Wadhvani,
and Manasi Gyanchandani

Department of Computer Science and Engineering,
Maulana Azad National Institute of Technology, Bhopal, India

Abstract. Time series Forecasting has attracted attention over the last decade with the boost in processing power, the amount of data available and the development of more advanced algorithms. It is now widely used in a range of different fields including Medical Diagnostics, Weather Forecasting, Financial time series etc. In this paper, we propose a model of attention mechanism that allows for attended input to be fed to the model instead of the actual input. The motivation for the model is to show a new way to view the input so that the model can make more accurate predictions. The proposed LSTM model with the attention mechanism is then evaluated on common evaluation metrics and the results are compared with state of art models like CNN-LSTM and Stacked LSTM to show its benefits.

Keywords: Attention · Time series forecasting · RNN · LSTM · CNN

1 Introduction

Time series consists of a sequence of observations collected sequentially through time. Forecasting of time series involves learning history of time series data for a variable and using it to predict its future values. Various models have been applied to time series forecasting including Vanilla LSTM model, CNN-LSTM and Deep LSTM [1]. But the research in the field is still on a rise due to the substantial impact that a slight improvement can have for a company. Financial time series data is particularly challenging because of the dynamic nature of financial market. The times series data is usually noisy and is non stationary and lack a generic trend in the data [9].

Various methods are used for time series forecasting including decomposition, smoothing, regression and machine learning methods. These techniques can broadly be categorized into statistical and soft computing approaches. Method of decomposition involves splitting the time series into its trend, seasonality, cycle and noise component. The seasonal component do not change, so the seasonally adjusted times series can be easily forecasted using regression or similar techniques. While this not only improves understanding of the data, it also helps

in improving prediction accuracy of the model[2]. Regression based forecasting techniques are similar to supervised prediction models where y_t is target, p and q are coefficients and t is the independent variable for time. The aim is to forecast future value of y at time t based on estimated values of p and q obtained through training.

$$y_t = p * t + q \quad (1)$$

Autoregression(AR), Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) are some of the different statistical methods [3]. Most commonly used one is the ARIMA, which is being widely used in industries for forecasting production, price prediction etc. It is a linear regression model and requires data to be stationary. One of the common approaches used for it is differencing. ARIMA incorporates both autoregressive and moving average features, where autoregressive term values are regressed on lagged values of itself and moving average term is the linear combination of past error terms. Due to the assumptions made to perform predictions with statistical methods, the models are not able to generalize well for unseen data [14]. It performs better than deep learning models for simple problems. But when it comes to learning complex non-linear and highly volatile data like stock price, these models fails to make accurate predictions.

Different machine learning models have been employed for time series prediction of financial times series data like Support Vector Machine (SVM) and Artificial Neural Network (ANN). Kara in [17] showed that SVM outperforms the ARIMA models [17] in stock prediction and similarly ANN have shown better prediction compared to both SVMs [16] and ARIMA [15] model. When the data considered is sequential data, it requires the model to have the ability to learn the dependencies in data. Recurrent Neural Network (RNN) is a network of gated cells, designed to capture the temporal information in data [17]. Since then, gated cells have been used in various sequential prediction applications [18,19].

Models being used in practice include Long short term memory (LSTM), Recurrent neural network (RNN), Convolution neural network (CNN) etc. [4]. CNN models have been applied in Natural language problems in order to gain meaningful understanding of the sentence [20,21]. Similarly, LSTM models have shown superior results in sequence prediction problems like speech recognition[18], sentiment analysis [22] and time series prediction[12]. There are also different hybrid models being used in various papers, that show good prediction accuracy, but they fail to provide good accuracy across different sets of data. Even though LSTM models solve some of the issue in sequence prediction to some extent, it does not do very well for long sequences. Attention mechanism is perceived as a solution to the problem. Originally developed to improve the neural machine translation[6]. Since then there has been several attempts to implement similar mechanism into various fields like Medical diagnosis[10], weather forecast[11], finance[7–9,12] for achieving similar improvements. Here, we attempt to use the mechanism for time series forecasting of financial data and propose an attention LSTM model for time series prediction. The model is

evaluated with three different sets of data and its performance is compared with other state of art time series forecasting models.

The rest of the contents are organized in sections. Section 2 presents an overview of LSTM cell architecture. Section 3 explains the attention mechanism which used in the proposed model. Section 4 describes the real data application for the experiments conducted on three real world financial datasets. Section 5 presents the results and discusses the inferences drawn with Sect. 6 concluding the paper.

2 Long Short Term Memory Neural Network

Time series generally have dependencies between the data points, which are difficult to be learnt by traditional neural network models. So, normal cells cannot be used to learn a sequence. Recurrent neural networks are special cells which have the ability to retain states. This allows the cell to learn dependencies in sequential data as well as recognize pattern and predict trends. The RNN model has been used since for sequential data problems like time series forecasting, machine translation and other similar applications [5]. But even RNN fails, when it comes to learning really long sequences, that is, they are only able to predict short term dependencies in sequential data because of the vanishing gradient problem. It is not a problem fundamental to neural networks, instead it is a problem with gradient based methods and certain activation functions.

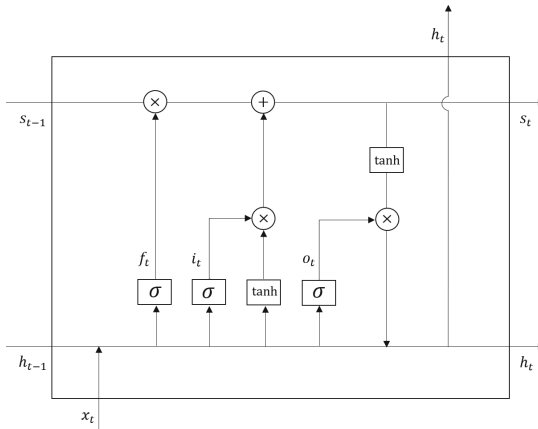


Fig. 1. Block diagram of an LSTM cell

LSTM was introduced by Hochreiter and Schmidhuber in late 90 s [10] to defeat this hurdle. LSTM can learn dependencies in sequential data over very long periods. This is achieved through its internal gate structure which allows it to selectively forget irrelevant information, update relevant information and

output certain information. This can be observed from the block diagram of an LSTM cell presented in Fig. 1. The forget gate is a sigmoid layer which takes in output from previous time-step and the current input. The output of this layer is then multiplied with the internal state to determine if the previous internal state is completely forgotten or not (2). The input gate is also a sigmoid layer that receives previous output and new input and outputs a value in range 0 to 1. The output is then added to candidate layer to update internal state (3). The output gate controls how much of the internal state is passed to the output and works similar to other gates (5).

$$f_t = \sigma(W_f[h_{t-1}; \bar{x}_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i[h_{t-1}; \bar{x}_t] + b_i) \quad (3)$$

$$s_t = f_t * s_t + i_t * \tanh(W_i[h_{t-1}; \bar{x}_t] + b_s) \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}; \bar{x}_t] + b_o) \quad (5)$$

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

where $[h_{t-1}; \bar{x}_t]$ is a concatenation of the previous hidden state and current input, f_t is the forget gate, i_t is the input gate, o_t is the output gate, s_t is the cell state and h_t is the hidden state. W_f , W_i , W_o are weight matrices and b_f , b_i , b_o are the bias learned during training. σ is the logistic sigmoid function, $*$ is elementwise multiplication.

3 Attention Mechanism for Forecasting

The idea behind attention mechanism was to imitate selective attention in the visual cortex of the human brain. It was introduced in natural language processing for improving neural machine translation [6]. Machine translation till the time employed encoder-decoder model. The model consists of two parts, first is the encoder which takes as input the input vector and encodes it into a fixed length vector called context vector. Next the context vector is passed as input to the decoder section which decodes it to give the output. However, when it came to translating long sentences, the model did not perform well, partly because the entire information was present in a single node, which made it difficult for the decoding model to interpret it.

Attention mechanism was implemented to overcome the limitation in the encoder-decoder model. In encoder-decoder model the complete sequence information in a timestep need to be stored into the last encoder node. The output from the last node is the context vector that carries complete information that decoder can use to translate. This resulted in not so good model as the context vector could not possibly store all the information specific to relevant data in

the sequence. So by incorporating attention, the model learns to attend to the relevant part of the input sentence in a particular timestep. This is achieved by assigning score, during training (7). As a result the decoder is having more information which allows it to better interpret the encoded sentence and ultimately results in a good translation model. Since then, attention mechanism has been studied and employed in various fields of prediction.

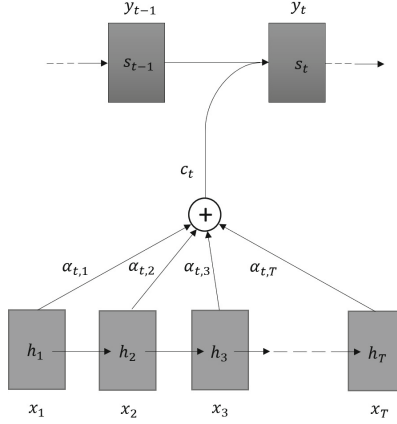


Fig. 2. Attention mechanism.

$$\text{score}(h_{t-1}, s_{t-1}) = v_a^T \tanh(W_a[h_{t-1}; s_{t-1}] + W_x x + b_a) \quad (7)$$

$$\alpha = \frac{\exp(\text{score}(h_{t-1}, s_{t-1}))}{\sum_{i=1}^t \exp(\text{score}(h_{i-1}, s_{i-1}))} \quad (8)$$

$$\bar{x} = (\alpha_1 x_1, \alpha_2 x_2, \dots, \alpha_t x_t) \quad (9)$$

where x is the given input, h_{t-1} is the hidden state and s_{t-1} the cell state. And W_a , W_x and b_a are the attention weights and bias. We then provide the attended input to the LSTM layer. This is shown in Fig. 2 in which the attended input is fed to the decoder LSTM layer.

4 Real Data Application

4.1 Dataset and Evaluation Metrics

The dataset used in the experiment are all financial time series data. The raw data is multifeild, having the following fields: Open, High, Low and Close. The Open indicates the rate at which stock opened that day, High indicating the highest rate that it reached during the day, Low signifies the lowest rate that it went during the day and close tells the closing rate for the day. Dataset that are

used in the experiment are from EURUSD Daily Exchange price data, Yahoo Stock price data and the Google stock price data. EURUSD dataset is a free historical data in daily, weekly and monthly time intervals for exchange rate of EURO to US\$. For the purpose of model training and evaluation we have used the daily interval data from the past 10 years. The training and testing the data is split, in the ratio of 2:3 and 1:3 for training and testing respectively. Yahoo stock price data consists of yahoo stock price data for the past 24 years approximately 6000 examples. Out of which training data comprises of historical data for 15 years. The rest is used for testing. Google data has a total of 8 yrs of stock data which is approximately 2000 examples. We used the 5yrs of data for training and the rest of the data to test the model. The train and test data are normalized separately to avoid capturing test information in training while normalizing. Additionally for some dataset, we have scaled the data to get better visualization and uniformity.

To compare the model in a standardized way, we have chosen to use the common evaluation metrics. The metrics used are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE).

$$\text{Mean Absolute Error (MAE)} = \sum_{t=0}^n \frac{|y_t - \bar{y}_t|}{n} \quad (10)$$

$$\text{Mean Absolute Percentage Error (MAPE)} = \frac{1}{n} \sum_{t=0}^n \frac{|y_t - \bar{y}_t|}{n} \quad (11)$$

$$\text{Mean Squared Error (MAE)} = \frac{1}{n} \sum_{t=0}^n (y_t - \bar{y}_t)^2 \quad (12)$$

4.2 Model Setup

The time series prediction model proposed in the paper uses LSTM incorporating the attention mechanism for improved accuracy for sequential data. We have developed the model for a univariate time series prediction, the architecture overview of which is shown in Fig. 3. It consists of two stage. The first stage is the attention stage that receives the actual input. So, this stage is where relevant information are attended using the attention mechanism. It takes the input, tries to learn trends and other salient features from input. The information learnt is used to update the attention weight matrices during training. The attended inputs are then fed to the second stage of neural network. It processes the attended inputs more effectively, thus, providing the model with much better idea about what part of the input to learn from and what are irrelevant information.

The model receives as input X_T , which is the raw input sequence. The first layer nodes add the current input to previous state output and cell state to give score for particular instance in time step. It is then transformed by hyperbolic function before exponentiation. The output is passed as input to softmax layer

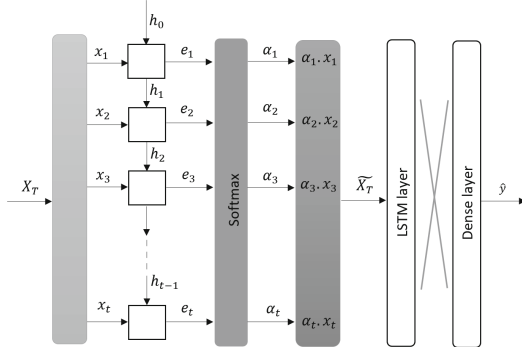


Fig. 3. Architecture overview

which normalises the score for instance. The generated coefficients define the importance of a point at a particular time step. The resulting value is then multiplied by the initial input to provide the attended input \tilde{X}_T to the LSTM. The LSTM neural architecture receives the attended input for training and learns the input better. The last layer of the network summarizes the learning and the output is generated by the dense layer \hat{y} . Regularization is applied to the model to prevent the model from overfitting. We have used early stopping using the loss function to implement the early stopping mechanism while training the model.

4.3 Methods for Comparison

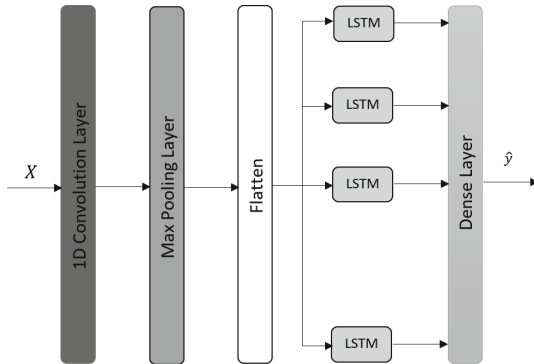


Fig. 4. CNN-LSTM architecture overview

The models used here for comparative performance evaluation are CNN-LSTM and Stacked LSTM. Stacked LSTM model contains LSTM layers stacked one on top of other. We have used a 3 layer stacked LSTM here for the model with a dense layer to provide the final output. CNN-LSTM, as shown in Fig. 4, is a

hybrid architecture involving convolution model and the LSTM model, where CNN and LSTM acts as encoder and decoder pairs. The convolution layer requires the data to be in three-dimensional format. For the experiment, a 1D convolution layer is implemented followed by a max pooling layer. The output is then flattened and fed to the LSTM layer. The LSTM layer is followed by a dense layer to provide the output.

5 Results and Discussion

Each model was trained with 10 years of the daily data samples and tested against data sample for 5 years. The model is optimized with fixed window size of 5 and forecasting size of 1 and training window size of 30 time steps. The error metrics were collected for each model runs. The experiment was repeated and average of the results were tabulated corresponding to the metrics.

Table 1. Evaluation metrics for comparative analysis of models

Model	EURUSD Dataset		
	MAE	MAPE	MSE
Stacked LSTM	0.0669150	7.1220994	0.0061878
CNN-LSTM	0.0459189	4.6624772	0.0026651
Attention LSTM	0.0202070	1.4430677	0.0006837
	Google Dataset		
	MAE	MAPE	MSE
Stacked LSTM	0.0530725	6.7868390	0.0057705
CNN-LSTM	0.0939735	21.7802674	0.0144475
Attention LSTM	0.0527558	3.4473487	0.0046541
	Yahoo Dataset		
	MAE	MAPE	MSE
Stacked LSTM	2.3425099	0.0217643	0.0007310
CNN-LSTM	0.0613030	6.4808816	0.0045402
Attention LSTM	0.0180222	1.1258750	0.0005943

Figure 5 represents the testing data plot for the three models corresponding to the three different datasets. Figure 5(a) is the test data plot for the EURUSD dataset. From the plot it can be seen that the two models LSTM and CNN-LSTM are not able to learn the trend in the data. However, Attention LSTM is able to predict the trend in the test data much better, with the same set of data is being used for all the models. One reason could be that the other models require more data to learn the trend in data, which is not seen in Attention model as it is able to learn the underlying data quite well with limited samples.

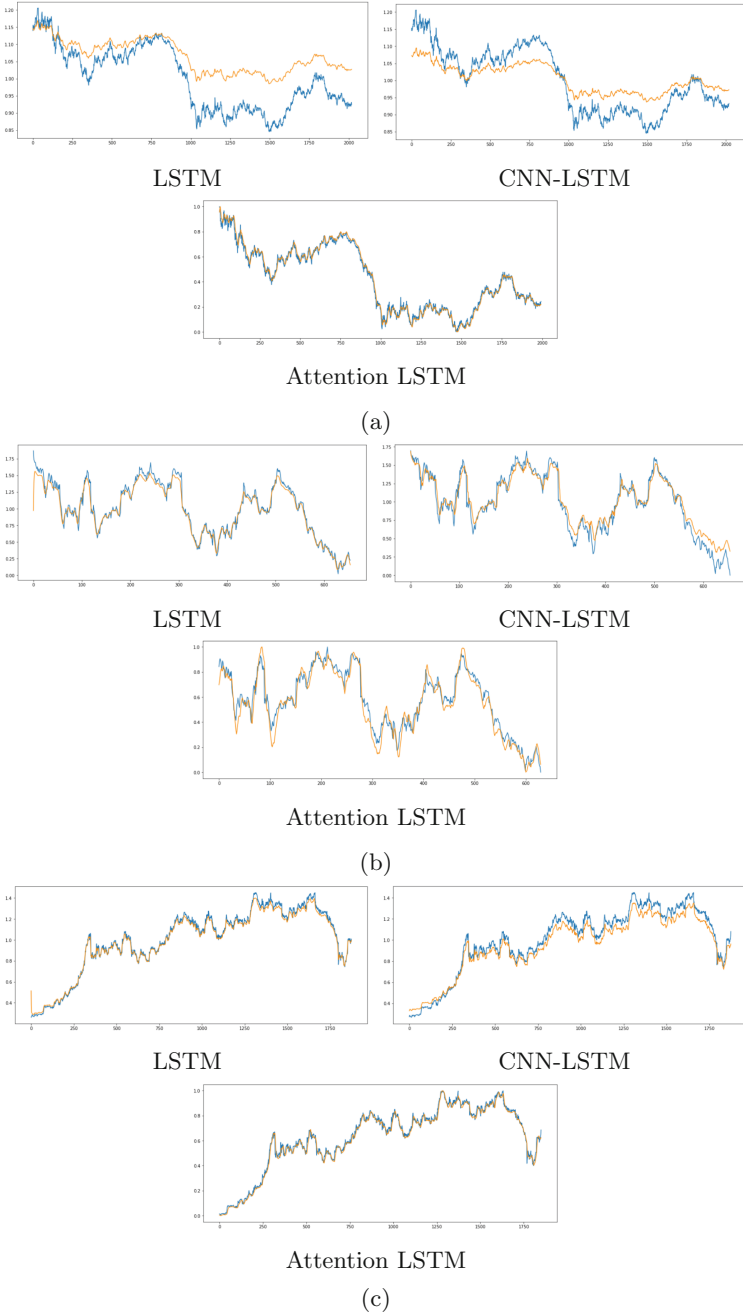


Fig. 5. Testing plot showing Actual (Blue) and Predicted (Orange) for: (a) EURUSD Dataset, (b) Google Dataset and (c) Yahoo Dataset

Figure 5(b) shows the graph plot for Google stock data. The testing plot for the three models show that models are able to learn the data well. Similarly, the Fig. 5(c) shows a better prediction accuracy for the Attention model compared to the Stacked LSTM and the CNN-LSTM model.

The performance metrics collected for the model in the experiments are summarized in the Table 1. The first column displays the different models used in the study, the next three column displays the error metrics MAE, MAPE and MSE values for the financial datasets on which the experiments were performed. For all the three datasets we can see that the Attention Model outperforms the rest clearly. The improvement seems to be very significant for EURUS Daily dataset, due to the nature of the particular dataset. As such, it can be inferred from the results that the model is more generic and provides competitive results across datasets. Other models struggle to capture long term dependencies and generality whereas attention proves itself to be potentially more suitable for grasping recurring patterns and being generic.

6 Conclusion

In this paper, we have proposed an LSTM model that is incorporated with an input attention mechanism. This mechanism allows for the model to better understand the data and make better prediction. The motivation behind using LSTM for implementing attention was due to the fact that, in the current scenario they are at the heart of all of all sequence prediction problems. To show the benefits we have used a vanilla LSTM model for comparison. The incorporation of attention enables the LSTM to learn the input better for improved prediction accuracy. The performance of the model is also compared with other state of the art hybrid models like CNN-LSTM. The models are trained and tested on three real world financial datasets. The evaluation metrics show the improvement that is achieved by incorporating the attention mechanism when compared with base models. It can be thus be stated that the incorporation of attention improves accuracy of the existing model. Future scope of the research is vast as there are still fields of research that could employ benefits of attention in them. Also there are statistical models, where incorporation of attention can help improve the accuracy. There is also scope for employing ensemble technique with attention models, which is yet another field to be looked at.

References

1. Pal, A., Prakash, P.K.S.: Practical Time Series Analysis. Packt Publishing, Birmingham (2017)
2. Deshpande, B., Kotu, V.: Data Science. 2nd Ed. (2019)
3. Brockwell, P.J., Davis, R.A.: Introduction to Time Series and Forecasting. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-319-29854-2>
4. Geron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Sebastopol (2017)

5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2017)
6. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations (2015)
7. Tran, D.T., Iosifidis, A., Kannianen, J., Gabbouj, M.: Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(5), 1407–1418 (2018)
8. Qin, Y., et al.: A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint [arXiv:1704.02971](https://arxiv.org/abs/1704.02971) (2017)
9. Wang, B., Huang, H., Wang, X.: A novel mining approach to financial time series forecasting. *Neurocomputing* **83**, 136–145 (2012)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
11. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. In: Proceedings of Advance Neural Information Processing System, pp. 2204–2212 (2014)
12. Xu, K., et al.: Show, attend and tell: Neural image caption generation with visual attention. In: Proceedings International Conference Machine Learning, pp. 2048–2057 (2015)
13. Chen, K., Wang, J., Chen, L.-C., Gao, H., Xu, W., Nevatia, R.: ABC-CNN: an attention based convolutional neural network for visual question answering (2015). <https://arxiv.org/abs/1511.05960>
14. Poterba, J.M., Summers, L.H.: Mean reversion in stock prices: evidence and implications. *J. Finan. Econ.* **22**(1), 27–59 (1988)
15. Kohzadi, N., Boyd, M.S., Kermanshahi, B., Kaastra, I.: A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing* **10**(2), 169–181 (1996)
16. Kara, Y., Boyacioglu, M.A., Baykan, Ö.K.: Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. *Expert Syst. Appl.* **38**(5), 5311–5319 (2011)
17. Cao, L.J., Tay, F.E.H.: Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Trans. Neural Netw.* **14**(6), 1506–1518 (2003)
18. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649 (2013)
19. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* **29**(6), 82–97 (2012)
20. Gao, J., Pantel, P., Gamon, M., He, X., Deng, L.: Modeling interestingness with deep neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2–13 (2014)
21. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 101–110. ACM (2014)
22. Palangi, H., et al.: Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. In: IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 24, no. 4, pp. 694–707 (2016)



Approximating Communication Cost for NFV-Enabled Multicasting

Yashwant Singh Patel¹, Shivangi Kirti²(✉), and Rajiv Misra¹

¹ Department of Computer Science and Engineering, Indian Institute of Technology, Patna, Bihta, India

{yashwant.pcs17,rajivm}@iitp.ac.in

² Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, India

sk2498@srmist.edu.in

Abstract. Network function virtualization (NFV) and service function chains (SFCs) effectively improve the flexibility of network service provisioning and increase the extent to which scaling can be done. However, finding an efficient deployment of virtual network functions (VNFs) for steering service function chain (SFC) requests is an NP-hard problem. The objective of our study is to obtain an optimal communication cost in VNF deployments and to allow for effective traffic steering in NFV enabled multicasting, when the number of SFC requests is large. Specifically, we have first formulated the problem and proved that it is NP-hard. We then present a 6α approximation algorithm for the centralized approach, where α is the approximation factor, and a $O(\log n)$ approximation algorithm for the distributed approach, where n represents the number of nodes. Through extensive simulations on synthetic and real-world networks, we have evaluated the performance of our proposed approach on multi-cast traffic to be better by 18.06% as compared to current state-of-the-art algorithms.

Keywords: Network function virtualization · Service function chain · NP-hard · Approximation

1 Introduction

Recently, network function virtualization (NFV) has drawn significant attention from vendors and telecom network operators by promising to enable flexible and cost-effective network management and service provisioning for future networks [1]. In a conventional network, each network service comprises of a number of network functions e.g. firewall, load balancers, intrusion detection systems (IDSs), WAN optimizers etc. Such network functions are integrated vertically in dedicated hardware middleboxes to ensure performance and security during data transfer. However, both the designing and deployment of proprietary hardware contribute significantly to the CAPital EXpenditure (CAPEX) and OPERATION

EXpenses (OPEX). NFV decouples network functions from underlying hardware and replaces the latter with flexible and innovative software middleboxes called virtual network functions (VNFs) [4]. Network traffic is required to pass through and get processed by a set of VNFs in a predefined order before reaching its destination as shown in Fig. 1. This phenomenon is known as Service Function Chaining (SFC). In realistic scenarios, the network topology and request

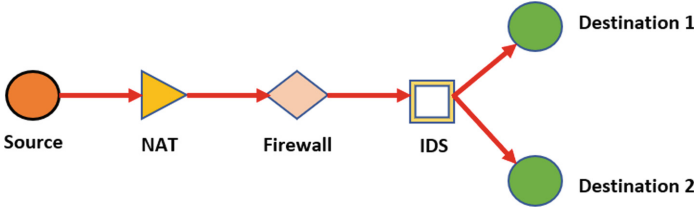


Fig. 1. Service function chain.

size can be massive and complex. However, finding an efficient deployment of Virtual Network Functions (VNFs) while steering Service Function Chain (SFC) requests is still an open problem. We focus on a scenario where a large number of multicast requests with network function requirements and data flows are required to be processed by a subset of VNFs.

Related Work: Recently, there have been several studies on placement and resource allocation issues for NFVs. Existing works mainly focus on routing where flows are directed through the required VNFs. Few other studies focus on the joint optimization of both VNF placement and routing. For example, Wood et al. [5] investigated the problem of NFV placement by formulating it as a mixed ILP (Integer Linear Programming) model and focusing on the minimization of resource utilization. Ghaznavi et al. [6] studied the problem of optimal VNF placement with the aim of minimizing the cost of host allocation and bandwidth resources. Eramo et al. [7] presented a multicast strategy for VNF placement and routing while ensuring the order of the flows to be passed (through VNFs). Soualah et al. [8] have proved VNF placement and chaining to be an NP-hard problem and have given a solution based on the extension of Monte Carlo Tree based model. Zhang et al. [9] have given an approach for multicast NFV placement and have also proposed an approximation algorithm with reduced complexity.

Our Contributions: To the best of our knowledge, all the existing heuristics solve the VNF routing problem using greedy paradigm-based heuristics. Our work focuses on finding an optimal communication cost using approximation algorithms. The key contributions of this paper are as follows:

- (i) Formulation of the optimal communication cost problem for minimizing VNF deployment and traffic steering cost and proving on the basis of reduction from Steiner Trees that the problem is NP-hard.
- (ii) Proposing a 6α approximation algorithm for the centralized model where α is an approximation factor.

- (iii) Proposing an $O(\log n)$ approximation algorithm for the distributed model where n is the number of nodes.
- (iv) Demonstrating the performance of the proposed algorithms over existing approaches through extensive analysis and experiments done on both synthetic and real-world networks.

2 System Model and Problem Formulation

We have considered a network in the form of an undirected graph having N heterogeneous servers as its nodes. We denote the set of servers where VNFs can be deployed by S_{vnf} and the set where such deployment is not possible by S_{free} (leading to $|S_{vnf}| + |S_{free}| = N$). Every server belonging to S_{vnf} has a minimum of one VNF being hosted by it. Additionally, all servers have varying amounts of the same resources allocated to them. The set of resources is denoted by R , and the amount of resource x present in server y is denoted by R_x^y . The network processes multiple flows at any given moment by steering them to the server(s) containing VNFs depending on individual requirement. A flow is denoted by F_x , and possesses a demand of flow D_x as well as a set of VNFs required for processing, $func_x$. It is imperative for each flow to get processed by all of the VNFs present in $func_x$ (for flow F_x). In order to calculate the feasibility of a particular path followed by a flow, an edge-based cost function is taken into consideration, denoted by $C_{i,j}$ (when the flow traverses through the edge $E_{i,j}$). Our objective is to reduce the total cost of processing all flows while meeting their computational requirements, taking in account the resources available to servers in the network and the total expenditure being incurred due to the traversal of the input flow from one server to another in order to satisfy its processing demands.

$$\min \left(\sum_{n \in |R|} \sum_{l \in |E|} \sum_{m \in |E|} C_{lm} g_{lm}^n + \sum_{i=1}^{n'} T_c(N_i) \right) \quad (1)$$

Subject to:

$$\sum_{l \in |V|} g_{lm}^n - \sum_{m \in |V|} g_{ml}^n = src(m), \forall n \in |I| \quad (2)$$

$$\sum_{l \in |V|} h_{lk}^n \geq 1, \forall n \in |I|, k \in F_n \quad (3)$$

$$\sum_{n \in |I|} d_f^n g_{lm}^n \leq B_{lm}, \forall l, m \in |E| \quad (4)$$

$$\sum_{n \in |I|} \sum_{k \in |F_i|} d_f^n r_{tk} h_{jk}^n \leq R_{jt}, \forall j \in |S|, t \in |R| \quad (5)$$

$$\sum_{l \in |V|} g_{lm}^n \geq h_{mp}^n, \forall n \in |I|, \forall m \in |V|, \forall p \in |F_k| \quad (6)$$

$$u_l^m - u_n^m + N g_{ln}^m \leq N - 1, \forall l, n \in |S|, \forall m \in |I| \quad (7)$$

$$h_{jl}^m \in 0, 1, \forall j \in |S|, \forall |m| \in |I|, \forall |l| \in |F_m| \quad (8)$$

$$g_{lm}^n \in 0, 1, \forall l, m \in |E|, \forall n \in |I| \quad (9)$$

Here, constraint (2) is a multi-constituent flow constraint, that can be applied to all nodes. Constraint (3) ensures that every function belonging to the processing requirement set of the flow must be processed at least once. Constraint (4) and (5) define the boundary conditions for the link capacity and server resource respectively. Constraint (6) represents the integral constraint of a request being processed by the functions hosted on a server X if and only if the request's path includes X as a constituent node. Constraint (7) prevents the creation of paths that form loops.

Theorem 1. *OPTimal Communication Cost (OPTCC) is an NP-Hard problem.*

Proof: Let the instance of the Steiner tree problem be referred to as P and that of our problem as P' . We initially reproduce P (as it is) to P' , and design the tree in such a way that it consists of only a source node, S , and a root node, R , which is the only node hosting VNFs. We designate all target nodes (for P) as the destination nodes (collectively referred to as D) for P' . The source and root nodes are connected via an edge, e , which has a random value $y > 0$ in order to obtain our problem instance P' . Thus, we prove that $OPT_{P'} = OPT_P + y$. As is obvious, e exists exclusively in P' and the solution of P' ought to have a subgraph in P (where P itself has a root node R , the same as that of P' and covers its target nodes). Thus, $OPT_P \leq OPT_{P'} - y$ holds. Also, $OPT_P \geq OPT_{P'} - y$ holds, a condition which if not valid would lead to any instance of P , along with e , to become a solution of P' with a cost smaller than $OPT_{P'}$. Hence, it is possible to obtain $OPT_{P'}$ (or OPT_P), if we are provided with OPT_P (or $OPT_{P'}$) via the deletion or inclusion of e' , leading to the proof of the theorem.

3 Centralized Model

We have considered uniform request cases. Every incoming request, $r_x(x \in [a])$, satisfies $r_x = r$, where $r \in \mathbb{Z}_+$. A k -tree that seeks to minimize the cost contribution from both the edge length and the terminal nodes is created in the initial stages, where $k = ra$. The sub-problem is modelled on an undirected graph $G = (V, E)$, a set of terminal nodes $T \subset V$, a root node $r \in V$, a function for defining the capacity of nodes belonging to T , $Cap: T \rightarrow \mathbb{Z}_+$, a function for defining the edge length, $E_l: E \rightarrow \mathbb{Z}_+$ and a function for defining the terminal cost value, $T_c: T \rightarrow \mathbb{Z}_+$. An acceptable solution tuple is (J, X) , where J is a k -tree and $X \subset T$ such that its capacity is greater than k . At a cursory glance, the problem description seems to resemble the classic node-weighted k -tree problem, though it is not quite so. The difference lies in the fact that the latter assigns costs to all the nodes and the aim is to form a tree T that seeks to minimize the expression $E_l(T) + T_c(V(T))$. For this proposed algorithm with uniform requests, we have assumed that $Cap(t) < ra$ holds $\forall t \in T$. If the condition does not hold for some $t \in T$, and $Cap(t) > r$, we append a set of new nodes $n_1, \dots, n_{Cap(t)/r}$ to t and connect them using edges that have an assigned length of 0. The capacities of the new nodes are defined as $Cap(n_i) = r \forall i < Cap(t)/r$ and $Cap(n_{Cap(t)/r}) = Cap(t) \bmod r$. t is removed from

T , and the set of new nodes $\{n_1, \dots, n_{Cap(t)/r}\}$ is added, following which the total number of nodes in the graph changes to $O(an)$. This in turn makes $Cap(t) < r$ for all the members of the set of terminal nodes. The proposed centralized algorithm is presented in Algorithm 1.

Algorithm 1: Centralized Algorithm

```

1 Input  $R, a, F_{R_i}, r, E, V, T_{in}, NT_{in}$ 
2 Output Optimal path  $\forall R_i \in R$ 
3 while ( $R \neq \phi$ )
4 begin
5   Set  $T \leftarrow T_{in}$ 
6   Set  $NT \leftarrow NT_{in}$ 
7   Update  $T$  and  $NT$  such that  $\forall T_i \in T, F_{T_i} \cap F_{R_i} \geq 1$ 
8   Set  $C_{T_i} \forall T_i \in T$ 
9   Take  $k = ra$ 
10  Define a terminal cost function,  $T_c$  for all the terminals
11  Solve the subproblem to obtain the solution tuple,  $(J, X)$ 
12  From  $J$ , find 'a' trees such that every edge in  $J$  is used by at most two trees.
    Use an allocation function  $\pi : T \times a \rightarrow Z_+$ 
13  For every solution tuple  $(J, X)$ , find a shortest path  $P$  from the root node to
    the member nodes of  $J$ 
14  Compute  $P \cap J$  and denote it as  $J'$ 
15  Output the value of  $J'$  for all requests, along with the allocation function  $\pi$ 

```

Lemma 1. $E_l(\hat{J}) + T_c(\hat{X}) \leq 3OPT$

Proof: Let $\{J'_1, \dots, J'_a\}$ be a set of trees and the function $\pi' : T \times [a] \rightarrow Z_+$ be used for assigning capacities to the terminal nodes of 'a' trees, which constitute an optimal solution for our problem. This leads to the value of OPT being equal to $\sigma_{i=1}^a E_l(J'_i)$. We take J'_x as an arbitrary spanning tree from amongst the union of the constituents of the set of trees. Additionally, we take X'_i to denote $\{t \in T(J'_x) : \pi'(t, i) > 0\}$ for each $i \in a$ and X' to denote $Cap_{i \in a} X'_i$. J' is a k tree and X' is a subset of $T(J')$ such that $Cap(U') \geq \sum_{i \in [a]} \sum_{t \in X'} \pi'(t, i) = k$. From the above, $E_l(\hat{J}) + T_c(\hat{X}) \leq E_l(J') + T_c(X')$. The definition of J' indicates that $E_l(J') \leq \sum_{i=1}^a E_l(J'_i) = OPT$. Hence, we can simplify the expression to be proved to $T_c(X') \leq 2OPT$. Let V' be a set defined as $t \in X' : \sum_{i \in [a]} \pi'(t, i) = Cap(t)$. We take the assumption that the function π' minimizes $\sum_{i \in [a]} |X'_i/V'|$ for all the allocations that combine to form a feasible solution with $\{J'_1, \dots, J'_a\}$. We know that the expression $|X'_i/V'| < 1$ holds for all $i \in [a]$ because if $p, q \in X'_i/V'$, the value of $\sum_{i \in [a]} |X'_i/V'|$ can be decreased by increasing $\pi'(p, i)$ and increasing $\pi'(q, i)$ until $\sum_{i \in [a]} \pi'(p, i) = Cap(t)$ or $\pi'(q, i) = 0$. We take $B = i \in [a] : X'_i/V' \neq \phi$ and t_i denote the node in X'_i/V' for all $i \in I$. $E_l(j, t) \leq E_l(J'_i)$ holds for each $t \in X'_i$ due to X'_i connecting both j and t . Additionally, $\sum_{t \in V'} \pi'(t, i) \leq r$ for all $i \in [a]$, and $Cap(t_i \leq r)$ for each $i \in I$. Hence,

$$\begin{aligned}
v'(X') &= \sum_{i \in [a]} \sum_{t \in V'} \frac{E_l(j, t) \pi'(t, i)}{r} \\
v &\leq \sum_{i \in [a]} E_l(J'_i) \sum_{t \in V'} \frac{\pi'(t, i)}{r} + \sum_{i \in I} E_l(J'_i) \frac{Cap(t_i)}{r} \\
&\leq 2 \sum_{i \in [a]} E_l(J'_i) = 2OPT
\end{aligned}$$

Lemma 2. There exists a set $\{J_1, \dots, J_a\}$ and a function for allocation $\pi : T \times [a] \rightarrow Z_+$, which satisfy the conditions specified in the stage of the centralized algorithm.

Proof: All the terminal nodes are initialized via the allocation function with a value of 0, following which the tree S is traversed using depth first search from the root node. When a particular terminal node is accessed for the first time by the algorithm, the smallest integer $i \in [a]$ is chosen such that $\sum_{t' \in T} \pi(t', i) < r$, and its cost value is modified to $\min(r - \sum_{t' \in T} \pi(t', i), Cap(t) - \sum_{i' \in [a]: i' < i} \pi(t, i'))$. The process is repeated until $\sum_{i \in [a]} \pi(t, i)$ becomes $Cap(t)$.

Lemma 3. $\sum_{i \in [a]} E_l(K_i) \leq T_c(X)$

Proof: If the condition $\pi(t, i) > 0$ is satisfied by a terminal, it implies $t \in V(J_i)$, from which $E_l(K_i) \leq E_l(j, t)$ is obtained. We know that $\sum_{t \in X} \pi(t, i) = r$ is valid for all $i \in [a]$. Hence, $\sum_{i \in [a]} E_l(K_i) \leq \sum_{i \in [a]} \sum_{t \in X} E_l(j, t) \pi(t, i) / r \leq \sum_{t \in X} E_l(j, t) Cap(t) / r = T_c(X)$

Theorem 2. *The centralized algorithm, on using an α approximation algorithm for the terminal cost k -tree problem, achieves 6α approximation.*

Proof: The length of the solution formulated by the algorithm is $\sum_{i \in [a]} E_l(J_i^*) = \sum_{i \in [a]} (E_l(J_i) + E_l(Q_i)) \leq 2E_l(J) + T_c(X)$, with the inequality being derived from Lemma 3 and the assumption that each edge is used by a maximum of two trees from the set $\{J_1, \dots, J_a\}$ being supported by Lemma 2. We know that the tuple (J, X) forms an α approximation solution for the node weighted k -tree problem, which implies, $E_l(J) + T_c(X) \leq \alpha(E_l(\hat{J}) + T_c(\hat{X})) \leq 3\alpha OPT$, in which the last inequality is obtained from Lemma 1. Thus, the length of the solution given by the centralized algorithm does not exceed $6\alpha OPT$.

4 Distributed Model

In this section we have developed an algorithm for a distributed model that processes uniform requests. The problem being NP complete, the algorithm is approximation based and has an approximation factor of $O(\log n)$. We have taken the set of all requests (which are equivalent) as $\{r_1, \dots, r_a\} = R$, and the graph as G (having weighted edges). We consider a specific case where the graph G is taken as a tree, wherein our algorithm formulates a solution having an approximation factor 2. When G is not taken to be a tree, another algorithm finds the path having the lowest cost value with an approximation factor of $O(\log n)$. This algorithm allocates a probability distribution across the constituent trees of the

graph. For each tree, all the nodes having a positive probability value (collectively denoted as N) satisfy the constraint $V \subset N$ and $|N| = O(n)$. Additionally, the function denoting edge length for the trees (E'_i) satisfies $E'_i(p, q) \leq E_l(p, q)$ for $(p, q) \in V$ where E_l is the function for allocating edge lengths in the graph. $E[E'_i(p, q)]$ is valid for all $p, q \in V$ and the complexity of the edge length between any two vertices is known to be $O(\log n)$. The proposed algorithm is presented in Algorithm 2.

Algorithm 2: Algorithm for Distributed Model

```

1 Input:  $G, R, a, F_{R_i}, r, E, V, T, NT_{in}$ 
2 Output: Set of sub-trees having no nodes in common while( $R! = \phi$ )
3 begin
4   for all  $r_i \in R$ :
5     begin
6       Designate terminal  $t_i$  for  $r_i$ 
7       for each  $r_i$  assign a root node  $R'$  and assume the graph to be a tree
8       for each terminal node  $t$  with degree  $> 2$ , add a new terminal node  $t'$ 
          to the original node and remove  $t$  from  $T$ 
9       Find a feasible solution of the sub-problem defined above, by joining trees of
          the solution set that have a node in common and repeating the process
          until convergence
10      Output the final set of trees formed as the solution

```

Theorem 3. *If the distributed model can have an α -approximation algorithm in a specific instance where the graph is a tree, an arbitrary instance of the model must have an $O(\alpha \log n)$ -approximation algorithm.*

Proof: We consider a tree $T'(V', E')$ with edge length function E'_i , obtained from a graph $G=(V, E)$, having an edge length function E_l . The tree is constructed via the aforementioned algorithm that allocates probability distribution across trees. This problem instance is solved using the α approximation algorithm, taking into consideration the metric, E'_i as the edge cost function and the original terminal set and capacity function. For a particular tree instance J'_i created for a request r_i , a corresponding tree J_i is created in G . Following this, a path that traverses through all the nodes in the tree J'_i via the implementation of Depth First Search and has a length of $2E'_i(J'_i)$ is created. All the nodes in $V(J'_i) \setminus V$ are shortcut. Each constituent edge of the path is replaced by the shortest alternative between the two vertices (of a particular edge) in G , and the edges that are redundant are deleted, to form the actual tree J_i . It is observed that the terminals of J'_i are a subset of that of J_i and that the expression $E[E_l(J_i)/E'_i(J'_i)]$ is valid. The solution consists of a set of trees $\{J_1, \dots, J_a\}$ which implement the same function for the allocation of costs as the one used for the solution component instance T' . We need to prove that the approximation factor is $O(\alpha \log n)$. We take the

set $\{Z_1, \dots, Z_a\}$ to represent the trees belonging to an optimal solution for the original instance and Z'_i to represent the sub-graph of T' that is obtained after modifying the edges in Z_i and replacing them with the shortest path between the two constituent nodes of each. It is observed that $E'_l(Z'_i) \leq E_l(Z_i)$, which in turn implies that $E[\sum_{i=1}^a E_l(J_i)] = O(\log n) \cdot E[\sum_{i=1}^a E'_l(J'_i)] = O(\alpha \log n) \cdot \sum_{i=1}^a E'_l(Z'_i) = O(\alpha \log n) \cdot \sum_{i=1}^a E_l(Z_i)$. The description of $\{J_1, \dots, J_a\}$ leads to the first equality, the fact that $\{J'_1, \dots, J'_a\}$ collectively form a solution having an approximation factor of α (where T' and E'_l are the instance components) is the basis for the second equality, and the third one is derived from the definition of $\{Z'_1, \dots, Z'_a\}$. This in turn proves that the solution can be found via an algorithm having an approximation factor of $O(\alpha \log n)$.

5 Performance Evaluation

To evaluate the performance, we first constructed a synthetic network with the help of ER random graph model [11]. Additionally, we have used a real-world network instance of Palmetto Networks [10] which has 45 nodes in the US. Using synthetic and real-world target networks, we have evaluated the performance of the proposed algorithms based on total destinations, number of nodes (50–300), VNF setup cost (based on normal distribution), link communication cost (based on Euclidean distance), and total length of SFC (5–20). We have compared our proposed algorithms with three baseline approaches: Steiner Tree, Set Cover and the Random Selection approach. The Steiner Tree algorithm spans a given set of vertices (also called Steiner points) with the minimal total distance calculated through the edges. The Set Cover algorithm selects the minimum number of nodes to cover as many VNFs as possible. The random method arbitrarily selects the nodes with sufficient capacities for different SFC requests.

5.1 Simulation Results

1. **Experiment with Synthetic Networks:** In Fig. 2, we have shown the variation of traffic delivery cost and running time with respect to growth in network size for centralized method under different destination ratios. It is observed that the average traffic delivery cost of the centralized method is 16.32% lower than the random method as shown in Fig. 2(a). We have represented the running time of algorithms in Fig. 2(b). As the network size and number of destinations increase, the total running time also increases in a nonlinear fashion. In Fig. 3, we have performed the same set of experiments for distributed method and observed that the average traffic delivery cost is 18.06% lower than the random method.

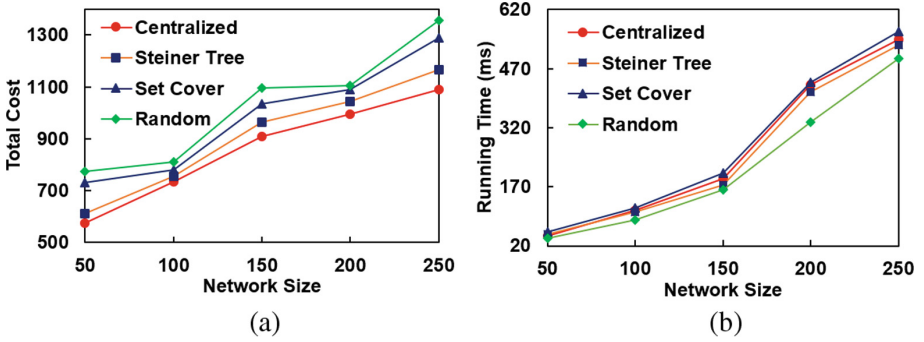


Fig. 2. Comparison of centralized model for (a) Traffic delivery cost and (b) Running time for $|D|/|V| = 0.1$ in synthetic network.

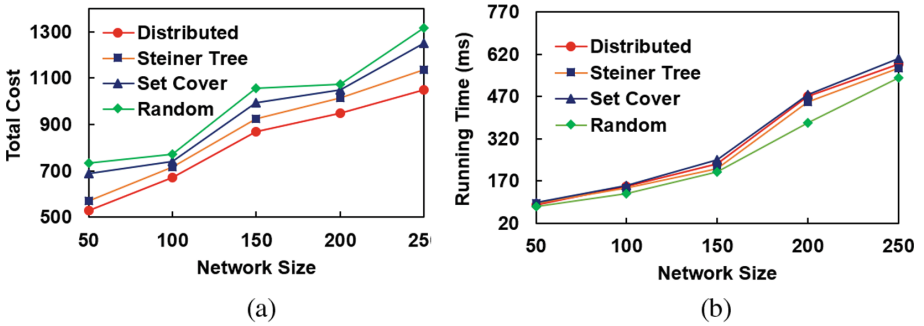


Fig. 3. Comparison of distributed model for (a) Traffic delivery cost and (b) Running time for $|D|/|V| = 0.1$ in synthetic network.

2. **Experiment with Real-world Network:** In this experiment, we evaluate the approximation algorithms in the real-world network topology of Palmetto [10]. To observe the impact of multicasting, we have fixed the SFC length to 10 and average deployment cost having $\mu = 2$. In comparison with the random method, the centralized method can minimize the average traffic delivery cost by 10.37% as shown in Fig. 4(a). On the other hand, the distributed method can reduce the average traffic delivery cost by 10.64% as shown in Fig. 5(a). The running time of all four algorithms increase with respect to the number of destinations as shown in Fig. 4(b) and Fig. 5(b).

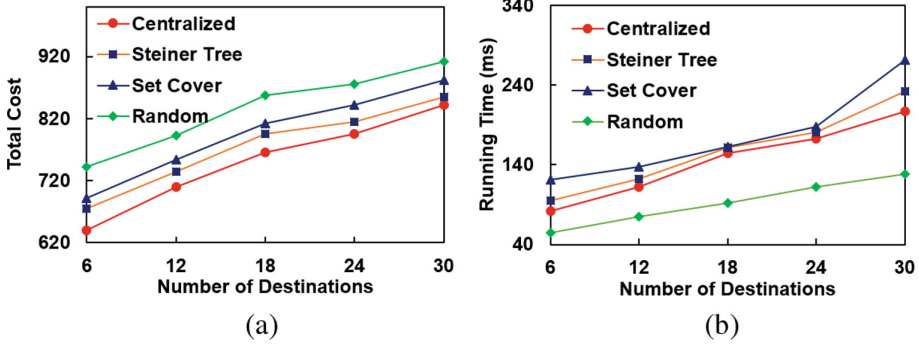


Fig. 4. Comparison of centralized model for (a) Traffic delivery cost and (b) Running time for PalmettoNet with different number of destinations.

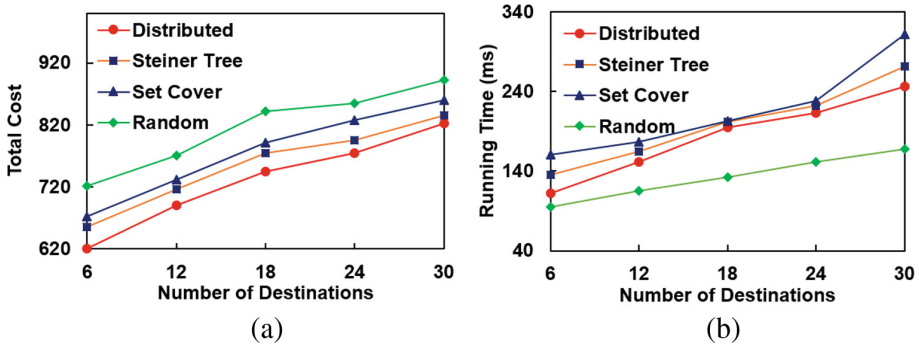


Fig. 5. Comparison of distributed model for (a) Traffic delivery cost and (b) Running time for PalmettoNet with different number of destinations.

6 Conclusion

In this paper, an extensive study of the communication cost problem for NFV enabled multicasting was made. Firstly, the problem was formulated and it was proved that it is NP-hard. Then, a 6α approximation algorithm was devised for the centralized approach, along with a $O(\log n)$ approximation algorithm for the distributed approach. Through theoretical analysis and extensive simulations, it was demonstrated that by applying our proposed solution, the cost saving of multicast traffic delivery is improved up to 18.06% compared to the current state-of-the-art algorithms.

Acknowledgement. The work is partially supported by the Department of Science & Technology (DST), Govt. of India under the ICPS Programme with Project No.: DST/ICPS/CPS-Individual/2018/403(G), “Low-cost Energy-Efficient Cloud for Cyber-Physical Disaster Management Systems.” The authors also acknowledge the Visvesvaraya Ph.D. Scheme for Electronics and IT under Ministry of Electronics and Information Technology (MeitY), Government of India.

References

1. Han, B., Gopalakrishnan, V., Ji, L., Lee, S.: Network function virtualization: challenges and opportunities for innovations. *IEEE Commun. Mag.* **53**(2), 90–97 (2015)
2. Martins, J., et al.: ClickOS and the art of network function virtualization. In: *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI 2014)*, pp. 459–473. USENIX Association, Berkeley (2014)
3. ETSI: Network function virtualisation. SDN and openflow world congress (2012). http://portal.etsi.org/NFV/NFV_White_Paper.pdf
4. Halpern, J.M., Carlos, P.: Service Function Chaining (SFC) Architecture. RFC 7665, 1–32 (2015)
5. Mohammadkhan, A., et al.: Virtual function placement and traffic steering in flexible and dynamic software defined networks. In: *The 21st IEEE International Workshop on Local and Metropolitan Area Networks, Beijing*, pp. 1–6 (2015)
6. Ghaznavi, M., et al.: Distributed service function chaining. *IEEE J. Sel. Areas Commun.* **35**(11), 2479–2489 (2017)
7. Eramo, V., Tosti, A., Miucci, E.: Server resource dimensioning and routing of service function chain in NFV network architectures. *JECE 2016* (2016)
8. Soualah, O., Mechtri, M., Ghribi, C., Zeglache, D.: Energy efficient algorithm for VNF placement and chaining. In: *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid*, pp. 579–588 (2017)
9. Zhang, S.Q., Tizghadam, A., Park, B., Bannazadeh, H., Leon-Garcia, A.: Joint NFV placement and routing for multicast service on SDN. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul*, pp. 333–341 (2016)
10. The internet topology zoo. <http://topology-zoo.org/maps/>. Accessed 23 June 2019
11. Erdos, P., Rényi, A.: On random graphs. *Publicationes Mathematicae (Debrecen)* **6**, 290–297 (1959)



Delay Analysis for P2P Systems Using LPWAN

Shivendu Mishra^(✉) and Rajiv Misra

Department of Computer Science and Engineering, Indian Institute of Technology,
Patna, Patna, India
{shivendu_2021cs08,rajivm}@iitp.ac.in

Abstract. LPWAN technologies such as-LoRa, Sigfox, and NB-IoT has been the key enabler in the advancement of Internet-of-things and Industry 4.0. LPWAN technologies known for long distances communication in low power devices at low operation cost for battery powered things. In this paper we have studied the performance of LPWAN in peer-peer (P2P) models for IoT-applications. We present a comparative analysis with respect to minimum data distribution delay for P2P system among LoRa, Sigfox, and Nb-IoT technologies, which shows that NB-IoT is best in LPWAN.

Keywords: Distribution delay · IoT · Latency · LPWAN · Long range connectivity · P2P

1 Introduction

Internet of thing (IoT) is based on billions or trillions of intelligent, vast processing power, and low power consuming sensors, actuators, and energy efficient appropriate connectivity medium. Connectivity is important for effective communication among sensors, actuators, and end users of IoT based /driven system. These systems use wired, wireless or combination of both medium for the connectivity. Wired connectivity mostly uses Ethernet technology but for wireless connectivity one can use Bluetooth, NFC, Wi-Fi, RFID, ZigBee, Wireless HART, IEEE 802.15.4, 6LowPan, Cellular based technology (2G-5G), LPWAN (Low-Power Wide Area Network) [1].

Wireless connectivity technology like Bluetooth, WI-Fi, RFID, NFC, and ZigBee mostly are used in short range communication, while Cellular communication (2G-5G), and LPWAN are extensively used for large range communication. As Cellular based technology provides larger coverage with extreme power consumption so IoT based /driven system mostly rely on LPWAN technologies which include LoRa, Sigfox, NB-IoT, Weightless, Telensa, Ingenu, and Dash7. LPWAN technologies fulfill specific design requirements such as long range connectivity (rural zones: 1040 km and urban zones: 15 km via low inference Sub 1GHz band and modulation schemes (Narrowband and Wideband)), low data rate, low energy consumption (via suitable topology (mostly star topology),

shorten device complexity, random access MAC protocol like ALOHA (some are using TDMA), and effective duty cycle), scalability (via exploiting the assortment in channel, time, space, hardware, adaptive channels selection and data rate), and cost effectiveness (via shorten the device complexity, license free band (ISM band), and sharing other network band (some of them using licensed band)) [2–4].

1.1 Road Map

The road map of the paper is as follows: Sect. 2, describes related works, Sect. 3, describes various LPWAN technologies, Sect. 4, describes minimum data distribution delay formulation, Sect. 5, describes comparative analysis of minimum data distribution delay in P2P system using LPWAN technologies (LoRa, Sigfox, and NB-IoT), Finally, the conclusions are given in Sect. 6.

2 Related Work

There are many works in literature on the analysis of LPWAN technologies. R.B. et al. [13] described analytical model to examine the performance of LoRaWAN in terms of latency, collision rate and throughput. Yousuf, A.M. et al. [24] described throughput, coverage and scalability of live LoRa networks using simulation. Rizzi et al. [14] assess physical and data link layers capability of LoRa link directly on the transceiver hardware. Pötsch et al. [15] presents experimental analysis of the end-to-end latency and jitter of a real-world LoRaWAN transmission. The authors in [5, 10, 11] have investigated the number of nodes support on a typical LoRaWAN, and some of the authors look on the scalability analysis of LoRa gateway [12, 16, 17]. Further, The authors in [18, 19] uses LPWAN in P2P IoT based applications. However, as per the knowledge none of the studies discussed about minimum data distribution delay analysis of LPWAN for P2P system in the literature. In this paper, We describes the formulation of minimum data distribution delay in P2P IoT system and compares LPWAN technologies based on the formulation.

3 LPWAN Technologies

LPWAN is progressively more ahead attractiveness in industrial and among research communities because of its design features. LPWAN technologies used in IoT applications like LoRa and Sigfox worked in unlicensed (free) frequency band. However, NB-IoT, Weightless, Telensa, Ingenu, and Dash7 worked in licensed (paid) frequency band. In this section, We look at the key LPWAN technology with their diverse role, technical aspect and explicit distinctiveness.

3.1 LoRa

LoRa means Long Range, is at present one of the most common LPWAN technology uses unlicensed frequency band (India: ISM band 865–867 MHz, Europe: 868 MHz, North America: 915 MHz, and Asia: 433 MHz). LoRa was developed by the founding member of the LoRa Alliance Cycleo of Grenoble, France, and later acquired by Semtech corporation. LoRa has six spreading factors (SF7 to SF12). The higher spreading factor permits longer communication range at the cost of lower data rate, and vice versa. In addition, LoRa technology have up to 20 km of coverage range, effective connectivity of up to millions of nodes, more than 15 years of battery life, and at max 50 kbps data rate. Further, basically, LoRa is a physical layer technology with maximum payload length for each message is 243 bytes. LoRa uses LoRaWAN protocol to defines the communication protocol and system architecture for the network. Additionally, LoRa uses the chirp spread spectrum (CSS) modulation to provides bidirectional communication, minimize signals noise levels, and enabling high interference resilience [6, 7].

LoRaWAN networks have usually three types of device classes, namely Class A, Class B, and Class C. Class A is a type of lowest power consumption device as it spending most of the time in sleep mode and is only waked up on a scheduled time or when it needs to transmit data. Class B type extended from Class A type, it opens additional receive windows at programmed time to determine the time of receiving data. Finally, Class C type is always listening to receive the data at any time except when transmitting data hence has the minimum latency among other types.

3.2 Sigfox

SigFox is developed by France Company Sigfox in 2009. It is working on non-licensed spectrum with the aims to create wireless networks for IoT devices with long range (rural areas: 20–50 km and urban areas: 3–10 km) connectivity, low power consumption, and low costs. SigFox has been quickly commercialized and provides network devices with ultra-narrow-band technology. The users of Sigfox are need to purchase end devices and subscription of Sigfox network providers to connect to regional Sigfox networks. Further, Sigfox also operates on unlicensed sub 1 GHz bands and utilizes a cloud server to receive, process the sensed data, and after processing sends it to a back-end server. The data rate of Sigfox is 100 bps with maximum payload length of 12 bytes for each uplink message with 140 messages per day over the uplink, and four messages per day over the downlink. In addition, the maximum payload length for each downlink message is eight bytes. Moreover, In Sigfox downlink communication only precedes uplink communication. In addition, uplink communication adapt binary phase shift keying (BPSK) and gaussian frequency shift keying (GFSK) adapted in downlink communication) [8].

3.3 NB-IoT

NB-IoT is developed as a fusion of NB-CIoT and NB-LTE by 3rd Generation Partnership Project (3GPP). NB-IoT utilizes the licensed frequency bands as LTE with three different deployment set-up, namely stand-alone (separate 200 kHz of spectrum), guard band (NB-IoT and LTE are co-located), and in-band (NB-IoT deployed within an LTE wide-band). Further, NB-IoT uses single-carrier frequency division multiple access (FDMA), orthogonal FDMA (OFDMA), and quadrature phase shift keying modulation (QPSK). Moreover, NB-IoT aim to provide a low cost, low power consumption (10 years of battery lifetime), highly scalable (up to 50,000 connections per sector), and wide range (1-10Km) of communication with low data rate (uplink: 20 kbps, and downlink: 200 kbps). Additionally, NB-IoT has support maximum 1600 bytes per message payload size, transmission bandwidth of 200 kHz, suitable in non-latency-sensitive (up to 10s) and low-bit rate applications [9].

3.4 Weightless

Weightless consist of a set of three LPWAN technology namely Weightless-W, Weightless-N, Weightless-P. Weightless developed by Weightless-SIG (Special Interest Group), a non-profit organization in 2008. The entire three Weightless technology is also operates in unlicensed sub 1 GHz spectrum with each of them has own unique features. Weightless-W uses TV whitespace spectrum, has a low overhead communication, high data rates (1 kbps to 10 Mbps), low battery life (up to 3 years), communication range (up to 5 km), and high deployment cost. In addition, the communication range of Weightless-W depends on the factors like obstacles, weather, etc.

Weightless-N is similar to Sigfox technology it adopts ultra-narrowband modulation scheme. Weightless-N has low cost (communication and deployment), low data rate (up to 100 bps) (Nwave, 2019), high battery life (up to 10 years), and the communication range of up to 5 km. Weightless-P uses ultra-narrowband modulation schemes (GMSK and offset-QPSK) and operates on the 12.5 kHz channels. Weightless-P is mainly developed for the industrial sector, has data rates ranging from 200 bps to 100 kbps, communication range of up to 2 km, battery life maximum 3 years, and provides a reliable bidirectional communication [20].

3.5 Telensa

Telensa is a proprietary LPWAN network developed in 2005 by electronic design consultancy Plextek. Typically, Telensa is also operates in unlicensed sub 1 GHz spectrum, can support bidirectional communication of maximum 5000 nodes, coverage range in urban areas of 2–3 km and in rural areas 5–8 km [22].

3.6 Ingenu

Ingenu is also operates in unlicensed sub 1 GHz band, developed in 2008 with the name On-Ramp Wireless and later in 2015 renamed as Ingenu. Typically, Ingenu is based on random phase multiple access (RPMA) technology. Ingenu have a long coverage range (up to 6 km), low-power consumption (10+ years battery life), and guarantee for secure wireless connectivity [21].

3.7 Dash7

Dash7 LPWAN technology developed by Dash7 Alliance, an open source protocol basically designed for wireless sensor network applications. Dash7 also operates in unlicensed sub 1 GHz band (433 MHz, 868 MHz and 915 MHz, can provide a long battery life, long communication range (up to 2 km), low latency, and low data rate (up to 167 kbit/s). Further, Dash7 support AES 128-bit shared key encryption (Dash7, 2019), consists of endpoints, sub-controllers, and gateways [23].

In summary, The following Table 1 shows technical aspect and explicit distinctiveness of the LPWAN technology's based on main features like range (communication), frequency band, data rate, battery life, and cost [2, 3].

Table 1. Comparative technical aspect and explicit distinctiveness of the LPWAN.

LPWAN	Range	Frequency band	Data Rate	Battery Life	Cost
LoRa/LoRaWAN	5–20 km	868/780/915 MHz	50 kbps	15 yrs	High
Sigfox	10–50 km	868/915 MHz	100 bps	15 yrs	High
NB-IoT	1–10 km	LTE and GSM	200 kbps	10 yrs	Very High
Weightless-N	2–4 km	Sub 1 GHZ	100 kbps	10 yrs	High
Telensa	2–8 km	Sub 1 GHZ	50 kbps	5 yrs	Medium
Ingenu	3–6 km	2.4 GHZ	8 kbps	10 yrs	High
Dash7	1–2 km	Sub 1 GHZ	167 kbps	10 yrs	Medium

4 Minimum Data Distribution Delay Formulation

Data distribution delay is the time taken to get a copy of data/information by all the end devices. Data/information distribution is mostly performed in IoT based /driven system to extracts and transmit sensor data for making smarter decision. Data distribution is mainly performed by wireless connectivity in either client server model or peer-2-peer model. In this section, We have described the formulation of minimum data distribution delay taken by LPWAN technologies in P2P system using LPWAN system model shown in Fig. 1.

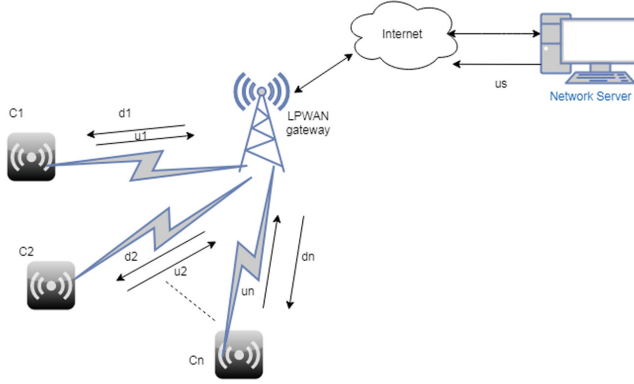


Fig. 1. LPWAN system model for delay analysis

4.1 LPWAN System Model

let us assume there are n end nodes $\{c_1, c_2, \dots, c_n\}$ connected to the network servers through LPWAN gateways with the downlink rate of end nodes i is D_{L_i} , and uplink rate of end node i is U_{L_i} . Moreover, Also assume that $D_{L_{min}}$ denotes downlink rate of end nodes with minimum value among others. Further, the data size is to be distributed is m , and server uplink is U_s . Now with respect to explained LPWAN system model in Fig. 1, the formulation of minimum data distribution delay of m size data for the case of P2P system is given in the subsequent subsection.

4.2 Minimum Data Distribution in P2P System

In this type of system the end nodes assist the server in distributing the data. i.e. after receiving the data, end nodes uses own upload link data rate to further distribute the data, and this process is repeated by other end nodes too so that after some times all end nodes in the system have the same information as the server. The following facts arises for P2P case in minimum data distribution delay:

- In the start, only server have the data of size m so must be upload by server once into the network. This uploading by server takes the minimum time $\{m/U_s\}$.
- The end nodes with minimum downlink rate $\{D_{L_{min}}\}$ takes at least $\{m/D_{L_{min}}\}$ time to download this m size file.
- Now as in P2P system server and all end nodes participates in the data distribution. Hence, the total data size $\{m.n\}$ is distributed collectively with say total upload link $U_t = U_s + U_{L_1} + U_{L_2} + \dots + U_{L_n}$ so the distribution delay is at least $n.m/U_t$

Thus using above observation distribution delay for transforming m size data into P2P case say T_{p2p} is given by the following equation:

$$T_{p2p} \geq \max\{m/D_{L_{min}}, \{m/U_s\}, \{m.n/U_t\}\} \quad (1)$$

Further, for simplicity let us consider the uplink and downlink of each end nodes is same and say it is u so $U_t = \{U_s + n.u\}$ and $D_{L_{min}} = u$. Now when number of end nodes n is very large then $m.n/U_t = m.n/\{U_s + n.u\} = m/u$, hence the distribution time is written by the following:

$$T_{p2p} \geq \max\{m/u, \{m/U_s\}, \{m/u\}\} \quad (2)$$

Again as U_s is more and more greater than u . Hence, We can write minimum distribution delay for P2P case as follows:

$$T_{p2p_{min}} = m/u \quad (3)$$

5 Comparative Analysis of Data Distribution Delay

In this section, We have compared minimum data distribution delay for P2P system only for loRa, sigfox, and NB-IoT communication technologies as these technologies are mostly used in IoT system in compared from others existing technologies. We have considered packet size in bytes (200B–1000B), minimum data distribution delay in milliseconds, and data rate of LoRa 50 kbps, Sigfox 100 bps, and NB-IoT 200 kbps respectively.

The following Fig. 2, and Fig. 3 obtained for data distributed in P2P case by applying the above data on the formula given in Eq. 3. The obtained Fig. 2, and Fig. 3 shows that in P2P case minimum data distribution delay is obtain lower in NB-IoT technology.

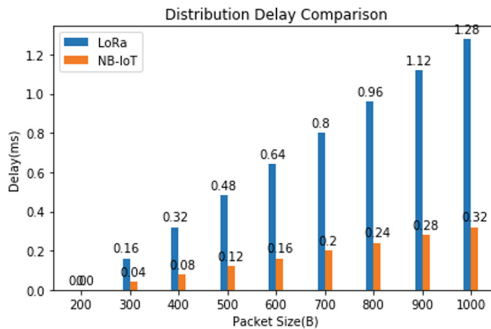


Fig. 2. LoRa vs NB-IoT minimum data distribution delay

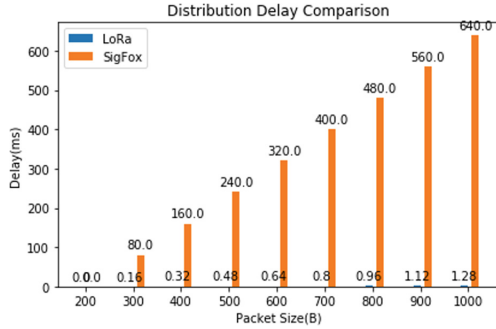


Fig. 3. LoRa vs Sigfox minimum data distribution delay

6 Conclusions

An important performance aspect for LPWAN technologies in peer-to-peer system is their minimum data distribution delay, which is the time taken to get a copy of data/information by all the end devices (peers). In this paper for P2P IoT systems, The formulation of minimum data distribution delay for LPWAN technologies have described. Moreover, comparative analysis of minimum data distribution delay among Lora, Sigfox, and NB-IoT based on the proposed formulation have also described. The comparative analysis shows that NB-IoT technology perform best in P2P case with respect to minimum data distribution delay.

Acknowledgment. This paper is financially supported by TEQIP-III of Rajkiya Engineering College Ambedkar Nagar, Uttar Pradesh, India which is a working institute of author Shivendu Mishra.

References

1. Cheruvu, S., et al.: Demystifying Internet of Things Security (2020)
2. Mekki, K., et al.: A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Exp.* **5**(1), 1–7 (2019)
3. Gu, F., et al.: Survey of the low power wide area network technologies. *J. Netw. Comput. Appl.* **149**, 102459 (2020)
4. Centenaro, M., et al.: Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. *IEEE Wirel. Commun.* **23**, 60–67 (2016)
5. Raza, U., et al.: Low power wide area networks: an overview. *IEEE Commun. Surv. Tutor.* **19**, 855–873 (2017)
6. Saari, M., et al.: LoRa - a survey of recent research trends. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings (2018)
7. Reynders, B., et al.: Range and coexistence analysis of long range unlicensed communication. In: 2016 23rd International Conference on Telecommunications, ICT 2016 (2016)

8. Margelis, G., et al.: Low throughput networks for the IoT: lessons learned from industrial implementations. In: IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings (2015)
9. Wang, Y.P.E., et al.: A primer on 3GPP narrowband internet of things. IEEE Commun. Mag. **55**, 117–123 (2017)
10. Mikhaylov, K., et al.: Analysis of capacity and scalability of the LoRa low power wide area network technology. In: European Wireless Conference 2016, EW 2016 (2016)
11. Bor, M., et al.: Do LoRa low-power wide-area networks scale? In: MSWiM 2016 Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (2016)
12. Georgiou, O., Raza, U.: Low power wide area network analysis: can LoRa scale? IEEE Wirel. Commun. Lett. **6**, 162–165 (2017)
13. Sorensen, R.B., et al.: Analysis of latency and MAC-layer performance for class a LoRaWAN. IEEE Wirel. Commun. Lett. **6**(5), 566–569 (2017)
14. Rizzi, M., et al.: Evaluation of the IoT LoRaWAN solution for distributed measurement applications. IEEE Trans. Instrum. Meas. **66**, 3340–3349 (2017)
15. Potsch, A., Hammer, F.: Towards end-to-end latency of LoRaWAN: experimental analysis and IIoT applicability. In: IEEE International Workshop on Factory Communication Systems Proceedings, WFCS, 2019-May (2019)
16. Voigt, T., et al.: Mitigating inter-network interference in LoRa networks (2016)
17. Aftab, N., et al.: Scalability analysis of multiple LoRa gateways using stochastic geometry. Internet Things **9**, 100132 (2020)
18. Kim, D.Y., et al.: P2P computing for trusted networking of personalized IoT services. Peer-to-Peer Netw. Appl. **13**, 601–609 (2019). <https://doi.org/10.1007/s12083-019-00737-z>
19. Chang, Y.C., et al.: A machine learning based smart irrigation system with LoRa P2P networks. In: 2019 20th Asia-Pacific Network Operations and Management Symposium: Management in a Cyber-Physical World, APNOMS 2019 (2019)
20. Weightless (2019). <http://www.weightless.org/>. Accessed 10 Feb 2020
21. Ingenu (2019). <https://www.ingenu.com/>. Accessed 10 Feb 2020
22. Telensa (2019). <http://www.telensa.com/technology>. Accessed 11 Feb 2020
23. Dash7 (2019). <http://www.dash7design.com>. Accessed 11 Feb 2020
24. Yousuf, A.M., et al.: Throughput, coverage and scalability of LoRa LPWAN for internet of things. In: 2018 IEEE/ACM 26th International Symposium Quality of Service, IWQoS 2018, pp. 1–10 (2019)



Data Mining Techniques in IoT Knowledge Discovery: A Survey

Beza Mamo Rabdo  and Asrat Mulatu Beyene  

BDA and HPC CoE, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia
asrat.mulatu@aastu.edu.et

Abstract. IoT is a buzzword nowadays and of course, it should be. The widespread of electronic and electromechanical devices with connecting ability to the Internet makes IoT be dominant from the user, manufacturer and services/goods provider perspective. Via IoT, the status of almost anything can be tracked, configured and maintained by different computing techniques using user devices or remotely from server ends. Determination of status can be easily known with data mining techniques that follow a distinct ladder until the representation of knowledge. In this survey work, we examined articles published from 2010 to date in the area of IoT. We followed a systematic literature review approach and scrutinize the different data mining steps followed by various scholars, and further classify the data mining techniques used in IoT as a conventional and non-conventional approach. Data cleaning, regression, model visualization, and summarization techniques were considered as challenging tasks due to the nature of IoT settings. This in turn demanded a new direction of research so as to come up with enhanced service provision in the area of IoT. Overlooked data mining techniques and comparison of the different approaches were criticized and reported. Moreover, the interdependency of IoT technologies with data mining approaches is discussed. Ultimately, an attempt has been made to indicate the research trend of IoT.

Keywords: IoT · Data mining techniques in IoT · Data cleaning · Regression · Data visualization and summarization · Context awareness

1 Introduction

The term Internet of Things (IoT) was coined by Ashton in 1999 [1]. Since then it has gained many contemplations by both academia and industry. The work of Sagar Sukode et al. in [2] suggested that the technology roadmap of IoT and its evolutions starts from communication between Hosts in the 1960s up to the existence of the actual Internet of Things which includes Hosts, People, ATM, Home, Mobile, Car, and the Web, among others. As per the Cisco Systems prediction [3], by 2020 over 50 billion devices are supposed to join the Internet every year. The diverse machines, devices and interactions among them are going to produce five hundred Zeta bytes of data every year [4].

An architecture suggested by Uviase et al. in [5] shown in Fig. 1 provides a concise illustration that summarizes and segments IoT in to three distinct layers as sensing, middleware and application. Regardless of the physical sensing arrangement the middleware involved in device and service discovery, context and quality of service management. In this work, we reviewed existing research work on the middleware layer. Additionally, the state of the art in IoT research has been reviewed which is, mainly, shifting towards the capability of connected objects to see, hear, and smell the physical world by thereof, and share the observations whenever required.

The newly designed paradigm of IoT, named as Cognitive Internet of Things (CIoT), prevail IoT with a ‘brain’ for high level intelligence as learnt from [6] and [7]. The CIoT makes use of a four layered approach that passes through from data sensing up to the intelligent service layer.

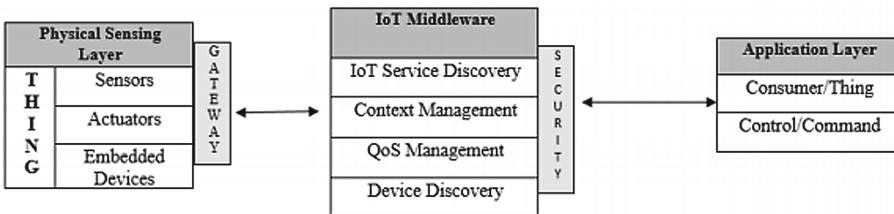


Fig. 1. IoT architecture [5].

As depicted in Fig. 2 the IoT middleware demands the consideration of data mining techniques in relation to Device Discovery and IoT Service Discovery sub layers. Accordingly, the need for data mining techniques is inevitable. And hence, it is one of the main focuses of this survey work.

Data mining techniques involve a stage of extracting the contained information and knowledge which is potentially useful and not known in advance. Information extraction is an effort of mining facts/meaning/context from huge, incomplete, noisy, fuzzy and random data. Information/Knowledge extraction from such unclean data involves a sequence of stages. The most frequently used steps in data mining are given in Fig. 2. It describes the successive scientific activities to be followed from data cleaning to knowledge representation.

It is apparent that IoT sensors continually generate large amount of data [8, 9] and [10]. Storing, managing and querying of such streaming IoT sensor data is a challenging task [8]. Extracting valuable or sensible information from a huge set of data using patterns can be attained via data mining techniques [10]. Moreover, originating helpful information from hidden raw data demand the use of various data mining algorithms and techniques as learnt from [9].

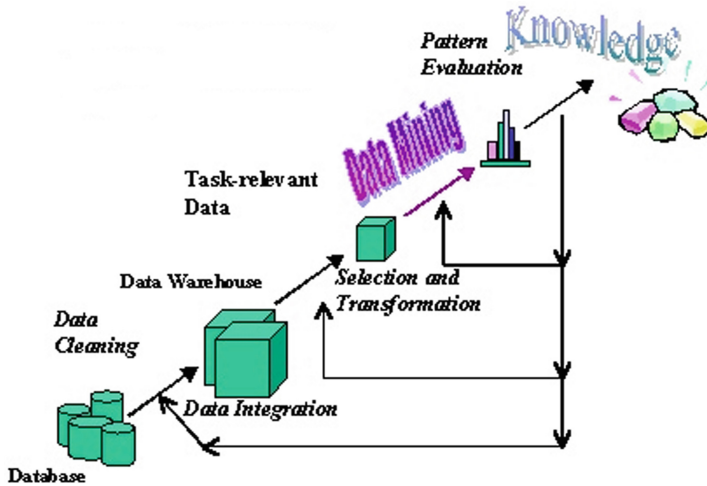


Fig. 2. Data mining as knowledge discovery process.

2 Research Method and Analysis

In this work, Systematic Literature Review (SLR) technique has been used as a fundamental research method that has five distinct steps as learnt from [11]. Accordingly, we subsequently set the research objective and question, search strategy, study selection, quality assessment criteria, and data extraction and synthesis details. The following subsections provide discusses each of adopted SLR activities.

2.1 Research Objective and Question

The major goal of this work is to undergo a literature review, since 2010, so as to recognize the practice of data mining steps in IoT and the usage of conventional and non-conventional data mining approaches in IoT.

To achieve this objective the following lists of research questions are identified:

- RQ1:** Identify which steps of data mining are mostly used in current IoT?
- RQ2:** Which steps of data mining are overlooked? Why?
- RQ3:** Identify which techniques of conventional data mining are exploited in IoT?
- RQ4:** Which data mining techniques are overlooked? Why?
- RQ5:** Identify which of non-conventional data mining techniques are mostly used in IoT?
- RQ6:** Which recent data mining techniques are overlooked? Why?

2.2 Search Strategy and Results

To address the questions articulated in Sect. 2.1, the following steps are undergone:

- i. Search information from two reputable data sources in the area of computer science, software engineering, information technology and other related subjects. The data sources used are IEEE Xplore and ScienceDirect databases.
- ii. Define search strings like “Data Mining in IoT”, “Data Mining Techniques in IoT”, “Information/Knowledge Extraction in IoT”, and “Knowledge Summarization in IoT”, among others.
- iii. The following additional criteria were set as part of our basic strategy:
 - a. Search result should contain only English Language
 - b. Discard the following articles: existing survey, and/or review papers
 - c. Results are sorted in their year of publication
- iv. Perceptive analysis of search results based on the expectations set forth. That means, papers that missed most of the criteria set for the evaluation purposes were discarded.

Based on our search strategy elaborated above we have got a total of 50 plus papers that are relevant for our analysis. Table 1 given below, indicates the summary of retrieved articles from the aforementioned sources.

Table 1. Summary of retrieved articles, URLs, search strings and other details

Database	URL	Search string	Date	Filter applied	Result	Remark
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp	Data mining in IoT	April 22/2019	Journals only	102	37 are applicable
ScienceDirect	https://www.sciencedirect.com/search/advanced	Data mining in IoT	April 22/2019	Journal and conference proceedings	51	13 are applicable

2.3 Study Selection

In conducting this study we rely on the following facts so as to set our study selection:

- Primarily focus on research works that are related with data mining in IoT domain. Even if there are other research databases we limit ourselves to IEEE Xplore and ScienceDirect due to acceptability, impact factor and reputability.

- We set the year 2010 with the ambition of getting research work in relation to data mining in IoT. We considered the year 2010 as base line and keep on looking the growing potential of mining techniques in IoT.

2.4 Quality Assessment Criteria

As part of quality assessment criteria, this study segregates the review process in to three broader evaluation matrixes:

- i) From the point of data mining steps:
 - Data cleaning, data integration, data transformation, pattern evaluation, and knowledge representation.
- ii) From the usage of conventional data mining techniques:
 - Association rule learning, regression, classification and clustering.
- iii) From the inclusion of non-conventional data mining approaches:
 - Context aware data selection, exploratory data analysis (EDA), model visualization and summarization.

2.5 Data Extraction and Synthesis

Evaluation of existing work by considering fundamental parameters of data mining techniques suggested by the work of [12] and [13] are used to extract and synthesize valuable information. The evaluation technique assesses each research work from [6, 7, 14–63] as indicated in Table 2 using basic parameters listed below:

- (++) Comprehensive: if the work exhaustively covers the issue and mention the parameter as part of its core work, methodology and summary;
- (+) Partial: if the work states the issue but not fully cover as part of its basic research;
- (–) None: if the work under review didn't consider the issue totally and/or put the parameter as part of their future work.

Table 2. Evaluation of research article on the basis of data mining steps and techniques

	Evaluation Parameter												
	Data Mining Steps					Conventional Data Mining Technique				Non- Conventional Data Mining Technique			
	Data Cleaning	Data Integration	Data Transformation	Pattern Evaluation	Knowledge Representation	Association Rule Learning	Regression	Classification	Clustering	Context Aware Data Selection	Exploratory Data Analysis	Model Visualization	Summarization
[14]	-	++	-	-	-	-	++	+	++	+	++	-	-
[15]	-	++	-	++	++	+	+	-	+	++	++	++	-
[16]	-	-	++	-	++	++	-	-	+	+	++	-	-
[17]	-	++	++	++	-	-	-	++	++	++	+	-	-
[18]	-	++	++	++	-	-	+	++	-	+	-	-	+
[19]	-	-	-	-	+	-	-	++	++	-	++	-	-
[20]	-	+	-	-	-	-	-	-	++	++	+	-	-
[21]	-	++	++	++	++	-	-	++	++	++	++	++	+
[22]	-	-	++	-	-	-	+	+	-	-	+	-	++
[23]	-	+	++	++	-	++	-	+	+	++	++	-	-
[24]	-	-	-	++	-	-	-	-	++	-	+	-	++
[25]	-	-	-	-	-	-	-	-	-	-	-	-	-
[26]	-	++	+	-	++	++	-	++	-	-	-	-	-
[27]	-	-	++	-	++	-	-	+	++	+	-	-	-
[28]	-	-	-	++	-	-	-	++	++	-	++	-	-
[29]	-	-	-	-	++	-	-	-	++	-	-	-	-
[30]	+	-	-	++	-	++	-	-	-	-	-	-	-
[31]	-	-	+	-	-	-	-	++	-	-	+	-	-
[32]	-	-	-	-	-	++	-	++	+	-	-	-	-
[33]	-	-	-	++	-	+	-	+	-	-	++	-	-
[34]	-	+	++	-	+	-	+	++	-	-	++	-	-
[35]	-	-	-	-	-	-	-	-	++	-	+	-	-
[36]	-	-	-	-	-	-	+	-	-	-	++	++	-
[37]	-	-	-	++	-	+	-	+	-	-	-	-	-
[38]	-	-	-	++	++	++	-	+	++	++	+	-	-
[39]	-	-	+	-	-	++	++	+	-	-	++	-	-
[40]	-	-	-	-	-	+	-	+	-	-	-	-	-
[41]	-	-	-	++	-	-	+	++	-	-	-	-	-
[42]	-	+	-	++	-	-	-	+	+	+	++	-	-
[43]	-	-	-	-	-	-	-	++	++	-	+	+	-
[44]	+	++	++	-	-	-	-	++	-	-	++	-	-
[45]	-	-	+	++	-	++	-	+	+	-	-	-	-
[46]	-	++	-	+	-	++	+	++	++	-	++	-	-
[47]	-	-	++	+	-	-	++	++	-	+	+	-	-
[48]	-	+	+	-	-	-	+	+	-	++	++	++	-
[49]	-	-	+	-	-	++	-	++	-	-	-	-	-
[50]	-	++	+	-	++	++	-	++	+	-	++	-	++
[51]	-	++	-	-	-	-	+	+	+	-	-	-	-
[52]	-	-	++	-	-	-	-	-	+	-	-	-	-
[53]	-	++	++	+	+	-	-	+	-	-	-	-	-
[54]	-	++	++	++	-	-	-	-	-	++	-	-	-
[55]	-	++	-	-	+	++	-	-	+	-	-	-	-
[56]	-	-	-	-	-	-	-	++	-	-	-	-	-
[57]	-	-	-	+	++	-	-	+	-	-	++	-	-
[58]	-	++	+	+	++	+	-	+	+	-	+	-	-
[59]	-	-	+	++	-	-	-	++	++	-	+	-	-
[60]	-	-	-	++	-	-	-	-	++	-	-	-	-
[61]	-	+	+	++	++	++	-	-	++	-	-	-	+
[62]	-	-	-	++	+	++	-	++	++	-	-	++	-
[63]	-	-	+	-	++	-	-	-	-	-	++	-	-
[64]	-	-	++	-	-	-	++	-	-	-	-	-	-
[65]	-	-	+	++	++	++	-	+	+	-	-	-	+
[66]	++	+	-	++	-	++	-	++	++	-	-	-	-

3 Analysis

In this sub-section we further examine the existing work with respect to the well-known mining techniques used in big data which is more relevant by its nature to IoT domain as depicted in Table 3. Additionally, we scrutinize all papers based on the algorithm used as part of their core research work as shown in Table 4.

Table 3. Data mining tools used

Reference list	Platform/tools used
[27, 31, 35, 46, 50, 51, 59, 61, 63]	Hadoop
[20]	Radpid Miner, Weka, MOA, SAMO, & Orange
[31, 51, 59]	MapReduce
[51]	Mahout
[16]	Pajek64 software tools**

** A tool used for analysis and visualization of large networks

4 Discussions

In this section, we discussed the results shown in Table 2 with assigned weighted values for each evaluation parameters as specified below:

- 1: for comprehensive (++)
- 0.5: for partial (+) and,
- 0: for none (–) so as to quantify the analysis results.

Accordingly, Sects. 4.1 up to 4.3 provide the survey summary with pictorial illustrations.

4.1 Data Mining Steps Used in IoT

Table 5 summarizes and rates the data mining steps used in IoT. It is apparent that data cleaning is ignored by most research works. The calculated subjective value (i.e., total weight) is further elaborated pictorially using Fig. 3.

Based on Fig. 3, it is noticeable that data cleaning, also called data cleansing or scrubbing activity, has given lighter attention as compared to the rest of data mining steps. Data mining steps without data cleaning sounds as doing successive steps of data mining on junk data. Inconsistent data from many different sources is the main challenge while handling big data [64]. To address this challenge, one possible solution is to situate stringent regulations from the commencement of data formation (data entry, submission, acquisition, and so on) as elaborated in [65]. The nature of IoT and big data environment which accommodate varied format of data set such as text, video, audio, video, and others by itself hinders the attempt of data cleaning as a widespread achievement.

Table 4. Data mining algorithms used

Reference list	Algorithm used
[17]	Obfuscation and Cognitive Concealment Algorithms
[6]	Device Clustering Based on Multimodal Data Correlation Algorithm
[19]	Output Perturbation (OPP) and Objective Perturbation (OJP) Algorithms
[21]	L-SIP, ClassAct, and BN and G-SIP Algorithms
[22]	Batch Variational Bayes Algorithm
[23]	Time Discrepancy Identification Algorithm
[24]	NPB Protocol Algorithm
[25]	Multidimensional Fusion Algorithm
[26]	Tri- Factorization Algorithm
[28]	Last Destination, Weighted Median, Largest Cluster, and Best Time Algorithms
[29]	FP Growth Based Algorithm
[30]	Gradient Boosting Algorithm
[32]	BOMO Algorithm
[33]	Algorithm to Detect the FDI Attacks in ICS
[36]	MADT Algorithm
[7]	Alternating Direction Method of Multipliers (ADMM)
[37]	Quad Tree Decomposition For Premium and Regular Partitioning Algorithm
[38]	ICSBP, BPNN, CSBPNN, ICSBPNN and Genetic Algorithm Optimization Back- Propagation Neural Network (GABPNN)
[41]	T-SNE Algorithm
[43]	CCCP Algorithm
[47]	Artificial Bee Colony (ABC) Algorithm
[53]	Deductive Algorithm (DA)
[57]	Bellare Canetti Krawczyk (BCK) with Pre-Shared Key and ECC Algorithms
[60]	Domain Generation Algorithm (DGA)
[59]	PARMA Algorithm
[63]	MAX-MIN Ant System Algorithm (MMAS)

Table 5. Assessment of data mining steps in IoT

Data mining steps	Comprehensive (1 pt each)	Partial (0.5 pt each)	Total weight
Data cleaning	[29, 42, 63]		3
Data integration	[14, 15, 17, 18, 20, 25, 42, 44, 48, 49, 51-53, 56, 59, 63]	[19, 22, 33, 40, 46]	18.5
Data transformation	[16-18, 20-22, 26, 42, 45, 50-52, 62]	[25, 30, 33, 37, 43, 46-48, 56, 57, 59, 61]	19.5
Pattern evaluation	[7, 15, 17, 18, 20, 22, 23, 27, 29, 32, 36, 39, 40, 43, 52, 57-60, 63, 64]	[44, 45, 51, 55, 56]	23.5
Knowledge representation	[7, 15, 16, 20, 25, 26, 28, 48, 55, 56, 59-61]	[6, 33, 51, 53]	16

4.2 Conventional Data Mining Techniques Used in IoT

Table 6 summarized the conventional data mining techniques used in IoT. It is apparent that regression technique is overlooked by most researchers. The calculated subjective value (i.e., total weight) is further elaborated pictorially by Fig. 4.

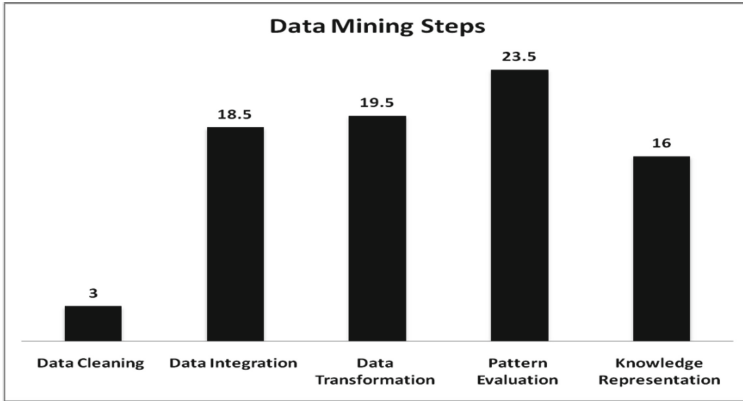


Fig. 3. Summary of data mining steps

As can be learnt from Fig. 4, regression techniques were used in limited instances when compared with other conventional data mining techniques. In general, conventional data mining approaches are not desirable to be applied in IoT. Latest trends and technologies like ubiquitous computing, semantic web, and IPv6 technologies are going to be integrated with IoT urging the need of novel data mining techniques [66].

One of the shortcomings of linear regression is that it requires deciding the type of basis function. It is often hard to decide the optimal basis functions [67]. Linear regression which frequently deals with economics, market analysis, and energy usage in relation to big data has a very limited implementation in IoT domain. As per our investigation none of the works addressed multiple regression techniques, among which [14, 33, 45], and [49] dealt with logistic regression techniques and the rest tried to explore linear regression.

Table 6. Survey of conventional data mining techniques

Conventional mining techniques	Rated as comprehensive	Rated as partial	Total weight
Association rule learning	[7, 16, 22, 25, 29, 31, 37, 43, 44, 47, 48, 53, 59, 60, 63]	[15, 32, 36, 38, 56]	18.5
Regression	[14, 45, 62]	[8, 33, 39, 46, 49]	5.5
Classification	[6, 17, 18, 20, 25, 27, 30, 31, 33, 39, 41, 42, 44, 45, 47, 48, 54, 57, 59, 63]	[7, 14, 21, 22, 26, 35–38, 40, 43, 46, 49, 51, 55, 56]	28
Clustering	[6, 7, 14, 17, 19, 20, 23, 26–28, 34, 41, 44, 57–60, 63]	[15, 22, 31, 40, 43, 48–50, 53, 56]	24

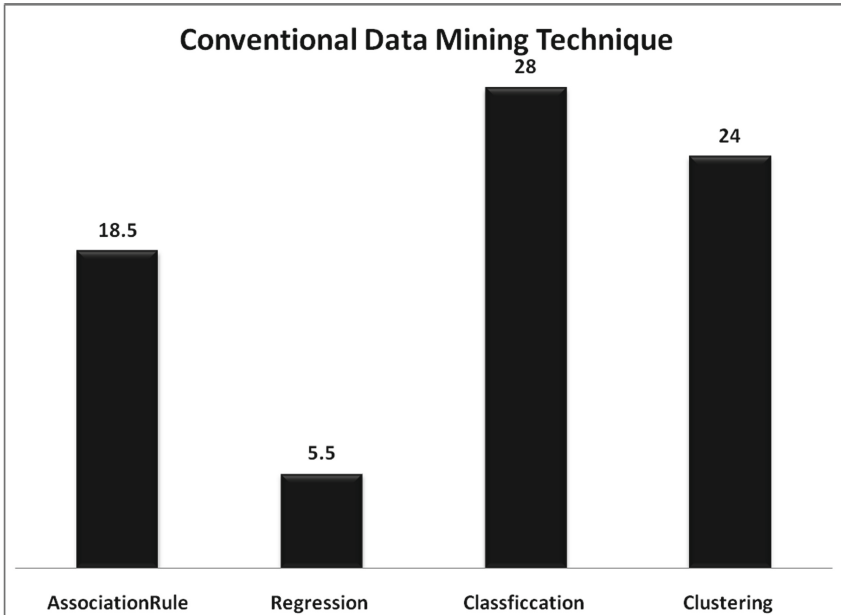


Fig. 4. Bar chart showing conventional data mining techniques

4.3 Non-conventional Data Mining Techniques Used in IoT

Table 7 condensed non-conventional data mining techniques used in IoT. It is evident that model visualization and summarization techniques are overlooked by most research works. The calculated subjective value is further elaborated using Fig. 5.

From the same figure, it is easy to see that model visualization and summarization techniques of data mining shows equal minimal rate. Both approaches have the highest level of knowledge representation which demands significant research work. Both are exceptionally difficult as IoT involves a huge data that made the summarization and model visualization approach very complicated.

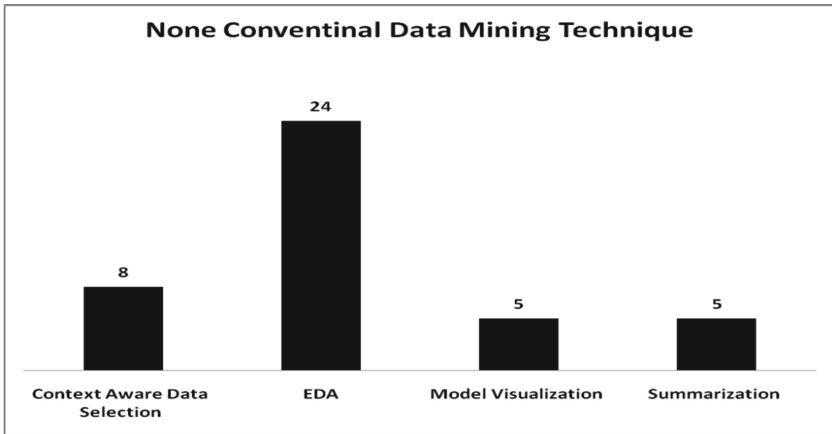
As per the work of [68] big data visualization is in a weak position by three factors: i) data challenge (volume, velocity, variety), processing challenge (like data collection and similarity resolution) and management challenge (like data privacy and data security). We strongly believe that the challenges exhibited in big data are also assumed to be the major challenges in IoT data visualization. Likewise, traditional document summarization methods are restricted to summarize appropriate information from the exploding IoT big data in the Cloud. The work of [69] represents the inherent structure of big documents using the semantic feature by the non-negative matrix factorization (NMF) as emerging techniques of data summarization.

5 Research Gaps, Open Issues, and Conclusions

In this section potential research gaps and open issues are summarized and conclusions are made that comprehend the overall study.

Table 7. Survey of contemporary data mining techniques in IoT

Mining techniques	Comprehensive	Partial	Total weight
Context Aware data selection	[7, 15, 17, 20, 22]	[14, 16, 18, 26, 40, 45]	8
Exploratory data analysis (EDA)	[6, 14–16, 20, 22, 27, 32, 33, 35, 37, 40, 42, 44, 46, 48, 52, 55, 61]	[7, 17, 19, 23, 30, 34, 41, 45, 55, 57]	24
Model Visualization	[15, 20, 35, 46]	[21, 41]	5
Summarization	[21, 23, 48]	[18, 20, 59, 63]	5

**Fig. 5.** Summary of non-conventional data mining techniques

5.1 Research Gaps and Open Issues

The following points are identified as the main research gaps and open issues based on our study:

- Considering the data mining steps in relation to IoT the concept of data cleaning is not yet fully addressed because of the nature of dissimilarity of data format, the uncontrolled and bulky nature of data, and lack of concise data set availability at large.
- Conventional data mining techniques are not desirable to extract valuable information in IoT. Due to the heterogeneous and unstructured data in IoT existing data mining techniques are not quite sufficient and requires new data mining techniques as learnt from [9]. As per our exploration, existing IoT research make use of the extensive techniques of conventional data mining techniques among which regression techniques are not used by most of the researchers as selection of basis function remained a challenging activity.

- Model visualization and summarization techniques are assumed to be the highest level of data representation which requires reliable and clean data representation. Having clean data is one of the challenging tasks as already mentioned above. This contributed a lot for failure of data visualization and summarization approach.
- Privacy and security issues remain open in the area of context awareness for a long period of time and still continued due to the spontaneous connectivity of pervasive systems to its surrounding appliances. Equally likely, as per our analysis the representation of low-level context which is similar to the formation of raw data in IoT is challenging task. Raw context data generated from diversified heterogeneous sensors require a thorough analysis to produce a meaningful data representation which is also a challenging task and requires due consideration in future IoT research efforts.
- Fog and Edge computing have got little attention by the majority of the work due to the limitation of processing and data holding capabilities. Scarcity of computational resources by edge computing can be supported by fog computing resources and it is easy to see that the new direction of computing lies on the formation of fog and edge computing features. Computing attempt on fog or edge computing brought enhanced service provision with better security and privacy features compared to cloud computing. Fog computing extends the cloud computing paradigm to the edge of the network, thus enabling a new breed of applications and services [70].

5.2 Conclusions

The trend of IoT research using data mining techniques is increasing from year to year based on the papers we analysed [6, 7, 14–63]. Figure 6 indicates research demand in the area of IoT at a glance. Moreover, based on our assessment, Table 8 summarizes the different categories of mining techniques used, the specific techniques in place, and the relative value of percentages within each category.

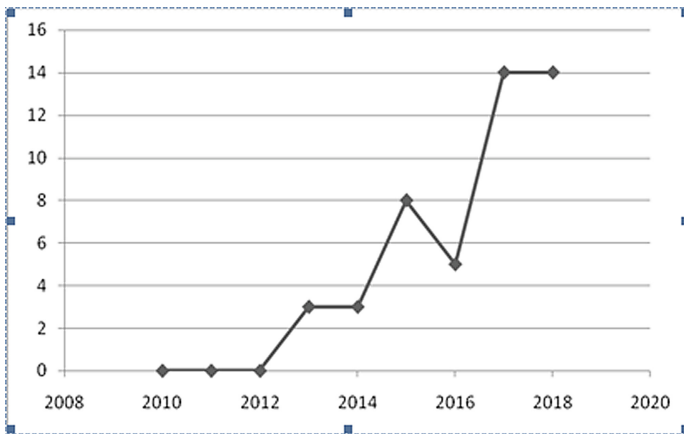


Fig. 6. Trend of IoT research in relation to data mining techniques

Table 8. Summary of mining techniques

Techniques	Data mining approaches	Value in %
Data mining	Data cleaning	3.8
	Data integration	20.89
	Data transformation	25.32
	Pattern evaluation	29.75
	Knowledge representation	20.25
Conventional data mining	Association rule learning	24.34
	Regression	7.24
	Classification	36.84
	Clustering	31.58
Non - conventional data mining	Context aware data selection	19.05
	EDA	57.14
	Model visualization	11.9
	Summarization	11.9

Even if all areas of mining techniques in relation to IoT are not well studied, we prioritize the following areas as per their value in percentage given in Table 8. Accordingly, data cleaning technique, regression, model visualization and summarization technique require due consideration.

Additionally, context awareness concerns in association with privacy, security and raw data representation requires in depth exploration as already indicated in section three and four of this work.

References

1. Ashton, K.: That ‘Internet of Things’ thing. *RFID J.* **22**(7), 97–114 (2009)
2. Sukode, S., Gite, S., Agrawal, H.: Context aware framework in IoT: a survey. *Int. J.* **4**(1), 1–9 (2015)
3. Evans, D.: The Internet of Things how the next evolution of the internet is changing everything (April 2011). White Paper by Cisco Internet Business Solutions Group (IBSG) (2012)
4. Shi, F., et al.: A survey of data semantization in Internet of Things. *Sensors* **18**(1), 313 (2018)
5. Uviase, O., Gerald, K.: IoT architectural framework: connection and integration framework for IoT systems. arXiv preprint [arXiv:1803.04780](https://arxiv.org/abs/1803.04780) (2018)
6. Lin, K., et al.: Device clustering algorithm based on multimodal data correlation in cognitive Internet of Things. *IEEE Internet Things J.* **5**(4), 2263–2271 (2017)
7. Wu, Q., et al.: Cognitive Internet of Things: a new paradigm beyond connection. *IEEE Internet Things J.* **1**(2), 129–143 (2014)
8. Patel, M., Minal, B.: Raw data processing framework for IoT. In: 2019 11th International Conference on Communication Systems & Networks (COMSNETS). IEEE (2019)
9. Savaliya, A., Aakash, B., Jitendra, B.: Application of Data Mining Techniques in IoT: A Short (2018)

10. Tapedia, K, Wagh, A.: Data mining for various Internets of Things applications. *Int. J. Res. Advent Technol.*, 127–132 (2016)
11. Okoli, C., Schabram, K.: *A Guide to Conducting a Systematic Literature Review of Information Systems Research* (2010)
12. Kumar, A., Tyagi, A.K., Tyagi, S.K.: Data mining: various issues and challenges for future - a short discussion on data mining issues for future work. *Int. J. Emerg. Technol. Adv. Eng.* **4**(1), 1–8 (2014)
13. Al Zamil, M.G., et al.: An annotation technique for in-home smart monitoring environments. *IEEE Access* **6**, 1471–1479 (2017)
14. He, W., Yan, G., Da, X.L. : Developing vehicular data cloud services in the IoT environment. *IEEE Trans. Ind. Inform.* **10**(2), 1587–1595 (2014)
15. Akbar, A., et al.: Real-time probabilistic data fusion for large-scale IoT applications. *IEEE Access* **6**, 10015–10027 (2018)
16. Elmisery, A.M., Sertovic, M., Gupta, B.B.: Cognitive privacy middleware for deep learning mashup in environmental IoT. *IEEE Access* **6**, 8029–8041 (2017)
17. Quick, D., Kim-Kwang, R.C.: IoT device forensics and data reduction. *IEEE Access* **6**, 47566–47574 (2018)
18. Verma, P., Sood, S.K.: Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet Things J.* **5**(3), 1789–1796 (2018)
19. Du, M., et al.: Big data privacy preserving in multi-access edge computing for heterogeneous Internet of Things. *IEEE Commun. Mag.* **56**(8), 62–67 (2018)
20. Ganz, F., et al.: A practical evaluation of information processing and abstraction techniques for the Internet of Things. *IEEE Internet Things J.* **2**(4), 340–354 (2015)
21. Gaura, E.I., et al.: Edge mining the Internet of Things. *IEEE Sensors J.* **13**(10), 3816–3825 (2013)
22. Puschmann, D., Barnaghi, P., Tafazolli, R.: Using LDA to uncover the underlying structures and relations in smart city data streams. *IEEE Syst. J.* **12**(2), 1755–1766 (2017)
23. Liu, Y., et al.: Exploring data validity in transportation systems for smart cities. *IEEE Commun. Mag.* **55**(5), 26–33 (2017)
24. Liu, W., Nakachi, K., Shoji, Y.: A neighbor-based probabilistic broadcast protocol for data dissemination in mobile IoT networks. *IEEE Access* **6**, 12260–12268 (2018)
25. Zhou, J., et al.: An efficient multidimensional fusion algorithm for IoT data based on partitioning. *Tsinghua Sci. Technol.* **18**(4), 369–378 (2013)
26. Hu, L., et al.: Semantic representation with heterogeneous information network using matrix factorization for clustering in the Internet of Things. *IEEE Access* **7**, 31233–31242 (2019)
27. Tianrui, Z., Mingqi, W., Bin, L.: An efficient parallel mining algorithm representative pattern set of large-scale item sets in IoT. *IEEE Access* **6**, 79162–79173 (2018)
28. Sui, P., Li, X., Bai, Y.: A study of enhancing privacy for intelligent transportation systems: *k*-correlation privacy model against moving preference attacks for location trajectory data. *IEEE Access* **5**, 24555–24567 (2017)
29. Tang, M., et al.: Mining collaboration patterns between apis for mashup creation in web of things. *IEEE Access* **7**, 14206–14215 (2019)
30. Huang, J., et al.: Efficient classification of distribution-based data for Internet of Things. *IEEE Access* **6**, 69279–69287 (2018)
31. Kaur, J., Kaur, K.: A fuzzy approach for an IoT-based automated employee performance appraisal. *Comput. Mater. Continua* **53**(1), 24–38 (2017)
32. Zhu, X., et al.: Mining effective patterns of chinese medicinal formulae using top-k weighted association rules for the internet of medical things. *IEEE Access* **6**, 57840–57855 (2018)
33. Zhang, Z., Wang, Y., Xie, L.: A novel data integrity attack detection algorithm based on improved grey relational analysis. *IEEE Access* **6**, 73423–73433 (2018)

34. Zhang, Q., Almulla, M., Boukerche, A.: An improved scheme for key management of RFID in vehicular Adhoc networks. *IEEE Lat. Am. Trans.* **11**(6), 1286–1294 (2013)
35. Choi, S., et al.: Chronological big data curation: a study on the enhanced information retrieval system. *IEEE Access* **5**, 11269–11277 (2016)
36. Gu, Y., Ren, F.: Energy-efficient indoor localization of smart hand-held devices using bluetooth. *IEEE Access* **3**, 1450–1461 (2015)
37. Wang, W., Wang, Q., Sohrawy, K.: Multimedia sensing as a service (MSAAS): exploring resource saving potentials of at cloud-edge IoT and fogs. *IEEE Internet Things J.* **4**(2), 487–495 (2016)
38. Li, S., et al.: An improved information security risk assessments method for cyber-physical-social computing and networking. *IEEE Access* **6**, 10311–10319 (2018)
39. Abeshu, A., Chilamkurti, N.: Deep learning: the frontier for distributed attack detection in fog-to-things computing. *IEEE Commun. Mag.* **56**(2), 169–175 (2018)
40. Fernandez Molanes, R., et al.: Deep learning and reconfigurable platforms in the Internet of Things: challenges and opportunities in algorithms and hardware. *IEEE Ind. Electron. Mag.* **12**(2), 36–49 (2018)
41. Lei, L., Qi, J., Zheng, K.: Patent analytics based on feature vector space model: a case of IoT. *IEEE Access* **7**, 45705–45715 (2019)
42. Zamil, A., Mohammed, G., et al.: An annotation technique for in-home smart monitoring environments. *IEEE Access* **6**, 1471–1479 (2017)
43. Zhang, D., et al.: NextMe: localization using cellular traces in Internet of Things. *IEEE Trans. Ind. Inform.* **11**(2), 302–312 (2015)
44. Gao, T., et al.: Interest-aware service association rule creation for service recommendation and linking mode recommendation in user-generated service. *IEEE Access* **6**, 57721–57737 (2018)
45. Zdravevski, E., et al.: Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering. *IEEE Access* **5**, 5262–5280 (2017)
46. Liu, S., et al.: Internet of Things monitoring system of modern eco-agriculture based on cloud computing. *IEEE Access* **7**, 37050–37058 (2019)
47. Tian, Q., Li, J., Liu, H.: A method for guaranteeing wireless communication based on a combination of deep and shallow learning. *IEEE Access* **7**, 38688–38695 (2019)
48. Farruggia, A., Magro, R., Vitabile, S.: A text based indexing system for mammographic image retrieval and classification. *Fut. Gener. Comput. Syst.* **37**, 243–251 (2014)
49. Ashokkumar, K., Sam, B., Arshadprabhu, R.: Cloud based intelligent transport system. *Proc. Comput. Sci.* **50**, 58–63 (2015)
50. Jiang, H., et al.: A secure and scalable storage system for aggregate data in IoT. *Fut. Gener. Comput. Syst.* **49**, 133–141 (2015)
51. Villalba, Á., et al.: servIoTicy and iServe: a scalable platform for mining the IoT. *Proc. Comput. Sci.* **52**, 1022–1027 (2015)
52. Koo, D., Kalyan, P., John Matthews, C.: Towards sustainable water supply: schematic development of big data collection using Internet of Things (IoT). *Proc. Eng.* **118**, 489–497 (2015)
53. Xiao, B., Kanter, T., Rahmani, R.: Constructing context-centric data objects to enhance logical associations for IoT entities. *Proc. Comput. Sci.* **52**, 1095–1100 (2015)
54. Allam, F., et al.: Analysis of eight data mining algorithms for smarter Internet of Things (IoT). *Proc. Comput. Sci.* **98**, 437–442 (2016)
55. Xia, M., et al.: Closed-loop design evolution of engineering system using condition monitoring through Internet of Things and cloud computing. *Comput. Netw.* **101**, 5–18 (2016)
56. Akhbar, F., et al.: Outlook on moving of computing services towards the data sources. *Int. J. Inf. Manage.* **36**(4), 645–652 (2016)

57. Gunupudi, R.K., et al.: Clapp: a self constructing feature clustering approach for anomaly detection. *Fut. Gener. Comput. Syst.* **74**, 417–429 (2017)
58. Suzuki, N., Matsuno, H.: Radio wave environment analysis at different locations based on frequent pattern mining. *Proc. Comput. Sci.* **112**, 1396–1403 (2017)
59. Rashid, M.M., Gondal, I., Kamruzzaman, J.: Dependable large scale behavioral patterns mining from sensor data using hadoop platform. *Inf. Sci.* **379**, 128–145 (2017)
60. Li, J., et al.: Mining repeating pattern in packet arrivals: metrics, models, and applications. *Inf. Sci.* **408**, 1–22 (2017)
61. Guo, K., Tang, Y., Zhang, P.: CSF: crowdsourcing semantic fusion for heterogeneous media big data in the Internet of Things. *Inf. Fusion* **37**, 77–85 (2017)
62. Rodríguez, S., Gualotuna, T., Grilo, C.: A system for the monitoring and predicting of data in precision agriculture in a rose greenhouse based on wireless sensor networks. *Proc. Comput. Sci.* **121**, 306–313 (2017)
63. Tsai, C.-W., Liu, S.-J., Wang, Y.-C.: A parallel metaheuristic data clustering framework for cloud. *J. Parallel Distrib. Comput.* **116**, 39–49 (2018)
64. Chen, J., et al.: Big data challenge: a data management perspective. *Front. Comput. Sci.* **7**(2), 157–164 (2013)
65. Anagnostopoulos, I., Zeadally, S., Exposito, E.: Handling big data: research challenges and future directions. *J. Supercomput.* **72**(4), 1494–1516 (2016). <https://doi.org/10.1007/s11227-016-1677-z>
66. Deshpande, P., Brijesh, I.: Research directions in the internet of everything (IoET). In: 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE (2017)
67. Mahdavinjad, M.S., et al.: Machine learning for Internet of Things data analysis: a survey. *Digit. Commun. Netw.* **4**(3), 161–175 (2018)
68. Olshannikova, E., Ometov, A., Koucheryavy, Y., Olsson, T.: Visualizing big data with augmented and virtual reality: challenges and research agenda. *J. Big Data* **2**(1), 1–27 (2015). <https://doi.org/10.1186/s40537-015-0031-2>
69. Ji, Y.-K., Kim, Y.-I., Park, S.: Big data summarization using semantic feature for IoT on cloud. *Contemp. Eng. Sci.* **7**(21–24), 1095–1103 (2014)
70. Bonomi, F., et al.: Fog computing and its role in the Internet of Things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. ACM (2012)



Literature Review on Answer Processing in Community Question Answering System

Saman Qureshi^(✉) and Sri. Khetwat Saritha

Department of CSE, MANIT, Bhopal 462003, India

Abstract. Community question answering (CQA) websites like Quora, Yahoo!Answers, Reddit enables users to ask questions as well as to answer questions. These sites are online communities that are popular now a days on the internet due to the increase of Question Answering (QA) websites and covers a wide variety of topics. Answer Processing task is classified as the ranking of answers, selection of answer through voting correlation, predicting the answer, selecting an appropriate answer from the candidate answers by classifying answer in good, bad, and potential category and then performing Yes/No task on selected answers or through best answer prediction or best answer selection. The shortcomings in the current approaches are the lexical gap between text pairs, dependency on external sources, and manual features which leads to a lack of generalization ability and to learn the associate patterns among answers. These shortcomings are resolved by already proposed work but they lack generalization ability and their performance is not satisfying. Feature extraction based methods mostly involve manual featurization which are not generalized form, therefore it can be avoided by deep learned feature. Whereas to focus on rich quality answers attention mechanism can be integrated with the neural network.

Keywords: Answer processing · Community question answering · Online community · Ranking · Voting · Answer selection

1 Introduction

CQA systems are a powerful mechanism that expects to give the most reasonable answers in the shortest possible time to the posted questions. Every day a colossal number of new questions posted and to answer these questions CQA systems can control the explicit knowledge or tacit knowledge so that it can be used effectively. Nevertheless, the user's request can be overloaded without appropriate collaboration support, due to which the CQA system would not able to achieve its main goal as askers would not be able to answer in the shortest possible time. And thus, to support the process of question answering, many approaches have been already proposed, pertaining to questions, answers, and users several data analyses and case studies have been conducted so far.

The several steps in the typical workflow of CQA portals are as follows. The asker firstly posts a new question in the CQA system, and then other users answer the question. In the CQA, the necessary data can be planned all the more accurately as the question

can be described in natural language and it does not have to be limited to some basic semantics. Therefore, the appropriate answer can be effectively received. After getting some answers to the question by posting remarks or voting the answers, the asker can choose the most appropriate answer and alternatively can be voted how good the answer is by other users.

The community question answering system has three stages: Question Processing, Document processing, and the last stage answer processing. Every stage involves a few steps. Parsing and classifying a question and reformulating query comes under question processing stage whereas document processing will find candidate documents and perform answer identification. And the last stage that is answer processing rank the best one answer or selects the best answer from a candidate answers after extraction. The proposed methods are based on patterns, statistical, and features. The workflow of a community question answering system is depicted in Fig. 1.

This paper is structured as follows. Section 2 states related work on answer processing phase in a CQA system. Then Sect. 3 discusses Answer Selection in detail. While Sect. 4 contains conclusion and future work.

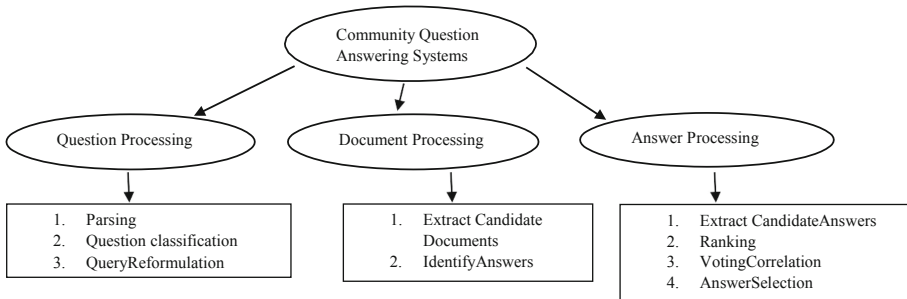


Fig. 1. Workflow of CQA system.

2 Related Work

2.1 Answer Processing

Answer Processing is the final stage of the question answering system where answer extraction is done. It is the most challenging task in CQA systems. When a user posts any question in a community site the answer to the question is given by other users. There can be more than one answer to a question all these answers in an all is called as candidate answers. The main task in answer processing is to select the right and related one answers to a question from these bunch of answers called as candidate answer. The work on answer processing tasks includes semantic similarities between a question and an answer, an answer which is more similar to a question is extracted or through voting correlation or ranking an answer or an answer selection. By these methods, answer processing is done in the CQA system.

2.1.1 Answer Processing Through Voting Correlation

The usability of Community Question Answering (CQA) facilitates the lives of users greatly as day by day its popularity is increasing, where ideas are exchanged and people seek help on the internet. Apart from asking and answering questions, users can provide feedback to these questions/answers through voting or commenting. Like in Stack Overflow forum programmer upload their programming questions and other programmers can give an answer for those questions and then that answer is validated by feedback from others. Such forums are used by millions of programmers when they encounter any kind of programming problems [1]. How to clear the doubts of users by detecting the correct answer? Can a good answer attract for a question? These questions are answered by [2] by voting correlation. [2] correlates the voting score of the answer with its question, and verifies such correlation in two data set that in turn boost the prediction performance. The voting score of a question or answers is characterized as the distinction of the absolute number of upvotes and the total number of downvotes. This voting score acts as an indicator of the intrinsic value of a question or answer.

Other related work is on the measurement of questions and answers by focusing on the quality of question/answer posts [3] in which human annotators label the quality of posts manually. [4] and [5] are proposed frameworks that determine the answer quality. The reiteration of a question is characterized under the estimation of question utility [6]. The methods by the authors: Jeon et al. [5], Suryanto et al.[4], Li et al. [7], Agichtein et al. [8] and Bian et al. [9] are some of the prediction methods for measurements. In the software forums a single question can have more than one answer and to find relevant answers Gottipati et al. [10] focuses on it.

A chunk of co-prediction algorithms is proposed by [2] where the high-impact question is acknowledged by the users in CQA sites through early detection of rich-quality questions/answers. Also to classify a useful answer that can avail positive feedback from users. This paper conjecture two things, one is that an interesting question can get more attention to receive high-score answers from potential answerers and whereas it might be very difficult for a low score question having weak expression in language, or lack of interesting topic to attract high-score answers. Mathematics Stack Exchange and Stack Overflow are the two real CQA sites that are studied for these conjectures. Armed with this verified correlation, the proposed method aims to identify the high-score potentially as soon as it is posted on the CQA sites. The contextual features which is focused are questioners'/answerers' reputation, the number of past questions/answers, length of the body, and title of a question or an answer. These features are extracted at every one hour whenever a question or answer is posted. Over the best contender, this joint forecast strategy accomplishes up to 15.2% net precision improvement and it allows to predict the result of voting for an answer before it appears on site. The effect of question/answer content on its dynamic and correlation is not covered by any proposed methods.

2.1.2 Answer Processing by Answer Ranking

Answer processing task can be considered an answer ranking task. Zhenlei Yan et al. [11] states the problem of the CQA system that many new questions are not able to be solved effectively by a suitable answerer. To resolve this routing task Zhenlei Yan et al.

rank the potential answerers to solve the question by their ability. A novel approach is proposed which simultaneously captures latent semantic relations among question, asker, and answerer by concatenating tensor model and topic model. A new learning procedure is proposed with tensor factorization which optimizes asker-topic-answerer model to execute the optimal answerer ranking task by maximizing multi-class AUC (Area under the ROC Curve). With two real world datasets from Tencent Wenwen (TW) and Yahoo! Answers (YA) this approach outperforms other related approaches.

The two features of new community systems are an ask-reply mechanism and social relations. Due to this researcher's concerns have shifted towards seeking potential answerers from finding existing answers. HAN Wenwen et al. [12] propose a hybrid method to address this problem. The framework considers the user's activity, social status, and authority by partitioning it into three parts question-user network, social graph, and ranking model using an optimized PageRank algorithm.

WikiAnswers, Yahoo! Answers, Baidu Zhida, are some Community web sites where users post a question and the answer to this question is answered manually by other users or it can also be answered automatically from existing community question answer knowledge base. These types of community sites have the CQA knowledge base which consists of question-answer pairs on a large scale. Question retrieval and answer ranking are the two main tasks in this domain. The former task estimates the semantic similarity between question-question pairs to detect similar questions whereas the later one task check the answer responses and rank them on the basis of semantic relatedness between question-answer pairs.

By identifying the major context of the question and some forms of question topic [13] performs the question retrieval task. The author in ref. [14] solve the word mismatch and word ambiguity problems in question by proposing a statistical machine-translation method where other languages are considered to get semantic information between question-question. Whereas for question-answer pairing [15] and [16] authors represent semantic relatedness between question and answer by constructing tree edit models. Considering answer selection task as answering ranking in ref. [17] the author calculate the semantic distance between the question and answer pairs using topic models to rank answers whereas Xiaobing Xue et al. [18] and Zhou et al. [19] uses translation and syntactic based approach. Many cases of semantic similarities are still not captured by these methods and this gap is covered by the authors in ref. [20] and [21] through Convolution Neural Network and Long short term memory deep learning models. The author in ref. [20] works on a question-question pairing task where it uses Ask Ubuntu data which is a part of the StackExchange community and improves accuracy by performing word embedding on different sizes with CNN. An LSTM model is used by the author in ref. [21] for question-answer pairing which sequentially reads words and gives relevance scores to rank answer. A part from these works [23] integrates the two tasks and both are considered as ranking tasks to improve the accuracy of CQA. Two ranking strategies: one is learning-to-rank with ref. to [22] where pairwise training is done and its output is used directly as a ranking score. And second, one train Support Vector Machine and Logistic regression supervised classification model and the probability of confidence score is used as a ranking score. SemEval CQA dataset is used and 45.12% of

MRR value achieved in answering tasks with the help question-question pairing. While propagating from question retrieval to answer ranking, this method reduces errors also.

2.1.3 Answer Selection

In the CQA system answer processing is a critical phase to extract the best answer in a less amount of time. The main problem in a community site is that when a question is posted a bunch of answers is given by users and in these answers, many are not so associated to the question asked and, in certain answers, even shift the topic to the context to a different subject as an example in Fig 1. This issue definition is nowadays considered as this resolves the criticality of answer processing (Fig. 2).

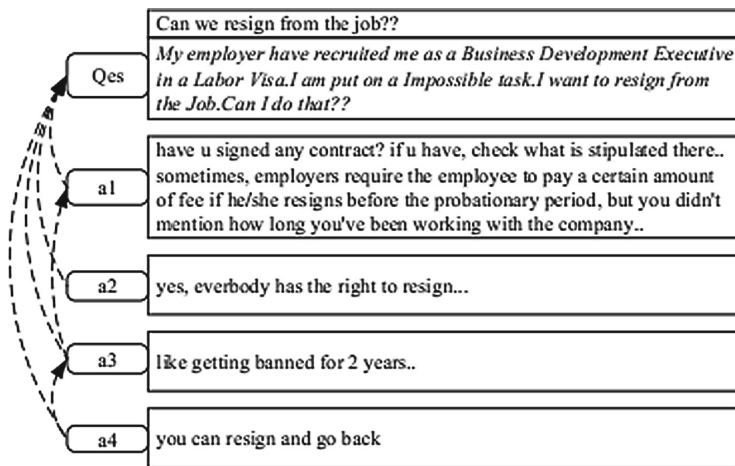


Fig. 2. An example of answers to a question [35].

3 Summary of Answer Selection Based Answer Processing Approaches

Yangsen Zhang et al. [25] removes dependencies of outer assets and manual features as they lack the generalization ability in most cases. These shortcomings can make up by deep learning architecture to catch the semantic data in texts with the utilization of word vector. The two models BLSTM and attention mechanism based on BLSTM is constructed to calculate semantic similarity. InsuranceQA dataset is used to evaluate the proposed approach. The answer with high semantic similarity is selected and accuracy QA-BLSTM achieve is 66.9% whereas QA-Attention Mechanism achieves 68.1%. Baseline models like QA-CNN-S [21], QA-CNN-GESD [21], QA-BLSTM-S [26] and QA-BLSTM-S-A [26] are compared with the two models which prove that BLSTM performs better than CNN as the former one capture a high measure of semantic data from a question and its candidate answers than the later one.

Yin et al. [27], did the comparative study of RNN and CNN. They did the comparison between LSTM and GRU in which they have found out that LSTM is good at modeling the sequence units in long text whereas CNN has an advantage in the short text by extracting invariant features.

Taihua Shao et al. [28] proposed the collaborative learning for answer selection which resolves the drawback of using a single deep neural network that fails to extract the rich sentence features. [28] build a parallel architecture by combining more than one neural network to collaboratively learn there presentations of question and answer. Firstly QA-CL model is built by deploying CNN with BiLSTM which will combine learn word vector matrix of question and answer parallely. Then, the QA-CL is extended to a hybrid collaborative QA-CLWR model which uses baseline weight removal (WR) to combine the generated sentence embedding with a joint distributed sentence representation. This experiment is conducted on the InsuranceQA dataset. The proposed models are compared with a non-neural network QA-WR [29] model, QA-CNN [30] model, and QA-LSTM/CNN [26] a hybrid model and shows a better performance against them. By achieving the accuracy of 61.22% the experiment performs better only with a medium number of questions as compared to a too small or too large number of questions. Table 1 compare the proposed methods on an InsuranceQA dataset.

Table 1. Result of different methods with InsuaranceQA dataset

Answer processing methods	Accuracy (%)
QA-CLWR [28]	61.22
QA-CL [28]	60.17
QA-LSTM/CNN [26]	59.33
QA-CNN [30]	57.33
QA-WR [29]	36.83
QA-CNN-S [21]	62.8
QA-CNN-S-GESD [21]	65.3
QA-BLSTM-S [26]	63.1
QA-BLSTM-S-A [26]	68.1
QA-BLSTM [25]	67.7
QA-Attention Mechanism [25]	68.5

3.1 Semantic Evaluation (SemEval)-2015 Task3

Semantic Evaluation is a progressing arrangement of assessments to evaluate semantic analysis system, where semantic analysis means analysis of meaning that is the nature of meaning in language is explored. Before SemEval Task 3, the proposed methods are on different independent datasets and to compare these methods results is a complex

task. Therefore, the common framework is provided by Task 3 of SemEval to compare different methods in multiple languages.

The task 3 in SemEval-2015 is related to answer selection in CQA. The feature of the task is a semantic similarity, natural language inference, and textual entailment. This task is initiated to automate the process of identifying the correct answer from the answer thread by classifying the answers as good, bad, and potential and producing all the valid answers by summarizing them as YES/NO.

To identify answer quality, JAIST [31] works on only Task A for English by extracting 16 features which belong to 5 groups (special component features, topic-modeling-based features, word-matching features, translation based features, and non-textual features). The system although achieves high results with 72.52% accuracy and holds rank one but due to heavy dependency on the bag-of-word the potential class is not handled properly.

A hierarchical classification method and a multi-classifier method are proposed by HITSZ-ICRC [32] team for English subtask A, English subtask B, and Arabic task. Two-level hierarchical classification and ensemble learning are proposed to classify answers for all three tasks English subtask A, English subtask B, and Arabic task. Fatwa dataset is used for Arabic task. Three submissions (primary, contrastive1, contrastive 2) were submitted for all three tasks. The Accuracies of English subtask A, English subtask B, and Arabic task is 68.87%, 64%, and 74.53% respectively, and holds the second rank.

QCRI [33], this team also works on the three tasks as HITSZ-ICRC works. In the Arabic task, this team holds the first rank and in the English subtasks the third rank. A supervised Machine learning approach is used considering numerous features i.e. text similarity, the context of a comment, sentiment analysis, word n-grams, and the presence of specific words. For Arabic task logistic regression is used and linear SVM is used for English subtask A. The team has also conducted a Post Experiment without and only a feature to understand the different features performance. The F1 score of Arabic task, English subtask A, and English subtask B is 78.55, 53.74, and 53.60 respectively.

ICRC-HIT [34] proposed a deep learning strategy and present a comment labeling system. To recognize a good comment, a recurrent convolution neural network is used.

The answer selection by Hongjie Fan et al. [35] is done using a multi-dimensional feature combination method. From every question and comment in the dataset, the information is extracted. The total 20 features were extricated dependent on the content description, text similarity, and attribute description. Using the SVM. Gradient Boosting Decision Tree (GBDT) and random forest, a model is built from the extracted features to classify dimensions obtained. Then an experiment is conducted which shows that the three methodologies are more effective than baseline models, and when contrasted with other proposed methods, relatively its ranking is on an all high. The selection of super-parameter of the model is randomly done which are not fine-grained and only 20 features were selected. But despite these limitations, the models ranking is high as compared to others. Different proposed methods for this task are stated in Table 2 and Table 3 for task A and task B respectively with their achieved accuracy.

3.2 Answer Selection by Predicting Best Answer

The objective of Question answering communities is to allow users to share knowledge by means of asking questions or by answering the questions asked by some other user.

Table 2. Result of methods for SemEval Task A

Answer processing methods	Accuracy (%)
JAIST [31]	72.52
HITSZ-ICRC [32]	68.67
QCRI [33]	70.50
ECNU [51]	70.55
ICRC-HIT [34]	67.68
GBDT [35]	68.12
Random Forest [35]	65.89
SVM [35]	43.35
Baseline	50.46

Table 3. Result of methods for SemEval Task B

Answer processing methods	Accuracy (%)
VectorSlu [52]	72.0
GBDT [35]	65.59
ECNU [51]	68.0
QCRI [33]	64.0
HITSZ-ICRC [32]	64.0
Random Forest [35]	65.52
SVM [35]	58.63
Baseline	60

Due to the large flow of information and lots of facilities communities' sites are being widely used nowadays. One of the issues in the answer processing task is to foresee the most fitting answer as not every asker has the capacity or information to choose the most fitting solution for his question.

Dalia Elalfy et al. [37] gives a model based on content feature to select the best answer by prediction method. The learning of the model is based on labeled data and it uses three type of features (1). Answer-answer feature, (2) question-answer features and (3) answer content features. Opposite to this model the [38] model is based on non-content feature where popularity score of the user who is responding to question in the stack overflow portal rather than Yahoo! Answer is measured. Merging these two proposed models with enhancement a hybrid model is build by [36] which consist of 3 different classifiers (Logistic Regression, Random Forest, and Naïve Bayes) to predict the most appropriate answer using some newly added features. The prediction results increase in the hybrid model as compared to the other two models and the accuracy is very promising.

To find autonomously the best answer in CQA services is an essential step. To validate a post voting up and voting down is done by users. The extraction of features is the main challenge while automating the selection of the best answer. Usually, the features are extracted from questions, answer, and metadata. Gkotsis et al. in [40] include comments for each answers as one of the features whereas the variance and average of comments are considered as the main feature by Tiametal.in [41]. [39] considered comments as a feature where text mining technique that is sentiment analysis is applied and answers spell checking is done. The social behavior of users and their activities are considered as informative features. Four big stack exchange websites (Math.SE, English.SE, Ask Ubuntu.SE, and skeptic.SE) from one of the biggest English CQA stack exchanges are considered to verify the work. The model uses 23 features which are selected from three categories Question and answer, comments, and user behavior. The performance of the model is tested on decision tree classifiers (like Adaboost) and some Alternate Decision Trees (ADT) classifier using Weka10. Evaluation of the model is done using F-measure

with a 10-fold cross-validation method. Results show improvement in performance as compared to other models by finding the best blend of different features.

3.3 Answer Selection by Selecting Best Answer

The expansion in utilization of CQA sites within incalculable questions and their relating answers increases the size of contents in this site. Traditionally best answer selection is done manually for the question asked, which is monotonous as to examine such semi-organized and colossal textual contents alongside the associate post score. To automate the selection of answers [42] proposed a model which instead of taking only question-answer related data it takes both answerers and question-answer data into account. This work analyses Stack Overflow Q&A posts, hence the Stack Overflow dataset is used. Based on activity signatures [43–45], domain knowledge [46], and topical similarity [47] the active answerers are identified to the asked questions. Also, topic modeling, topical interest, topical expertise [48], and voting scores are used. Then the relationship between Q&A pairs is found through topic relevance like [47]. At last to predict the best answer to the question asked at least five answers of Q&A posts are analyzed to focus on features involved as in [49] and [50] for pattern identification based on topic modeling and classifier. The results are evaluated with Precision-Recall Area Under Curve, Accuracy, Receiver Operating Characteristics Area under Curve, and Accuracy. The accuracy of the two classifiers (Bayes Net and Naïve Bayes) is calculated where Bayes Net outperform Naive Bayes by achieving an overall 69%. The calculation of expertise level and potential experts cannot be done with this model and pre-processing can affect the performance parameter for other CQA sites due to different meta data arrangements.

4 Conclusion and Future Work

Community Question answering websites consist of three phases: question phase, document or passage retrieval phase, and the last one answer processing phase. The answer processing is the challenging one task in Question Answering websites. The selection of the right from candidate answers for a question is the problem stated by CQA systems. The framework or method proposed for this problem is based on pattern matching, static-based, and feature-based. Giving upvote or downvote to an answer is allowed by many community sites and through voting correlation answer extraction is done. And the other ways are ranking the answer, or predicting the answer or answers election to process an answer. Challenges faced by CQA while answer processing is the lexical gap between question and question and a lexical gap between questions and answers and also a deviation from a question. These challenges are covered by proposed frameworks and methods but still their performance lack generalization ability and still, its accuracy can be improved more. Due to the use of external semantic resources and manual features, the generalization of the framework is not achievable and its performance is still can be improved. The probable solutions can be using deep learned feature instead of manual features, the lexical gap can be bridged by deep learning method as it can avoid feature engineering. And also to focus on high quality answers attention mechanism can be integrated with a neural network.

References

1. Osbourn, T.: Getting the most out of the web. *Softw. IEEE* **28**(1), 96 (2011)
2. Yao, Y., Tong, H., Xie, T., Akoglu, L., Xu, F., Lu, J.: Detecting high-quality posts in community question answering sites. *Inf. Sci.* **302**, 70–82 (2015). <https://doi.org/10.1016/j.ins.2014.12.038>, ISSN 0020-0255
3. Harper, F.M., Raban, D., Rafaeeli, S., Konstan, J.A.: Predictors of answer quality in online Q&A sites. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*, pp. 865–874. Association for Computing Machinery, New York (2008). <https://doi.org/10.1145/1357054.1357191>
4. Suryanto, M.A., Lim, E.P., Sun, A., Chiang, R.H.: Quality-aware collaborative question answering: methods and evaluation. In: *WSDM 2009: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pp. 142–151. Research Collection School of Information Systems (2009)
5. Jeon, J., Croft, W.B., Lee, J.H., Park, S.: A framework to predict the quality of answers with non-textual features. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pp. 228–235. Association for Computing Machinery, New York (2006). <https://doi.org/10.1145/1148170.1148212>
6. Song, Y.I., Lin, C.Y., Cao, Y., Rim, H.-C.: Question utility: a novel static ranking of question search. In: *Proceedings of the National Conference on Artificial Intelligence*, vol. 2, pp. 1231–1236 (2008)
7. Li, B., Jin, T., Lyu, M.R., King, I., Mak, B.: Analyzing and predicting question quality in community question answering services. In: *Proceedings of the 21st International Conference on World Wide Web (WWW 2012 Companion)*, pp. 775–782. Association for Computing Machinery, New York (2012). <https://doi.org/10.1145/2187980.2188200>.
8. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM 2008)*, pp. 183–194. Association for Computing Machinery, New York (2008). <https://doi.org/10.1145/1341531.1341557>
9. Bian, J., Liu, Y., Zhou, D., Agichtein, E., Zha, H.: Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In: *Proceedings of the 18th International Conference on World wide web (WWW 2009)*, pp. 51–60. Association for Computing Machinery, New York (2009). <https://doi.org/10.1145/1526709.1526717>
10. Gottipati, S., Lo, D., Jiang, J.: Finding relevant answers in software forums. In: *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, pp. 323–332. IEEE Computer Society, USA (2011). <https://doi.org/10.1109/ASE.2011.6100069>
11. Yan, Z., Zhou, J.: Optimal answerer ranking for new questions in community question answering. *Inf. Process. Manage.* **51**(1), 163–178 (2015). <https://doi.org/10.1016/j.ipm.2014.07.009>, ISSN 0306-4573
12. Wenwen, H., Xirong, Q., Siqi, S., Ye, T., Wendong, W.: Ranking potential reply-providers in community question answering system. *China Commun.* **10**(10), 125–136 (2013). <https://doi.org/10.1109/CC.2013.6650325>
13. Duan, H., Cao, Y., Lin, C.Y., Yu, Y.: Searching Questions by Identifying Question Topic and Question Focus, pp. 156–164 (2008)
14. Zhou, G., Liu, F., Liu, Y., He, S., Zhao, J.: Statistical machine translation improves question retrieval in community question answering via matrix factorization. In: *Proceedings of the Conference on ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1.852–861 (2013)

15. Heilman, M. and Smith, N.A.: Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In: Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT 2010), pp. 1011–1019. Association for Computational Linguistics, USA (2010).
16. Chang, M.W., Goldwasser, D., Roth, D., Srikumar, V.: Discriminative learning over constrained latent representations. In: Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT 2010), pp. 429–437. Association for Computational Linguistics, USA (2010)
17. Qin, Z., Thint, M., Huang, Z.: Ranking answers by hierarchical topic models. In: Chien, B.C., Hong, T.P., Chen, S.M., Ali, M. (eds.) Next-Generation Applied Intelligence. IEA/AIE 2009. Lecture Notes in Computer Science, vol. 5579. Springer, Heidelberg (2009)
18. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008), pp. 475–482. Association for Computing Machinery, New York (2008). <https://doi.org/10.1145/1390334.1390416>
19. Zhou, G., Cai, L., Zhao, J., Liu, K.: Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives, pp. 653–662 (2011)
20. Bogdanova, D., dos Santos, C., Barbosa, L., Zadrozny, B.: Detecting Semantically Equivalent Questions in Online User Forums, pp. 123–131 (2015). <https://doi.org/10.18653/v1/K15-1013>
21. Wang, D., Nyberg, E.: A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering, pp. 707–712 (2015). <https://doi.org/10.3115/v1/P15-2116>
22. Trotman, A.: Learning to rank. *Inf. Retr.* **8**, 359–381 (2005). <https://doi.org/10.1007/s10791-005-6991-7>
23. Lan, M., Wu, G., Xiao, C., Wu, Y., Wu, J.: Building mutually beneficial relationships between question retrieval and answer ranking to improve performance of community question answering. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 832–839. Vancouver (2016). <https://doi.org/10.1109/IJCNN.2016.7727286>
24. Zhou, X., Hu, B., Chen, Q., Wang, X.: Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing* **274**, 8–18 (2018). <https://doi.org/10.1016/j.neucom.2016.07.082>. ISSN 0925-2312
25. Zhang, Y., Peng, Y.: Research on Answer Selection Based on LSTM, pp. 357–361 (2018). <https://doi.org/10.1109/IALP.2018.8629166>
26. Tan, M., Santos, C.D., Xiang, B., Zhou, B.: LSTM-based Deep Learning Models for Non-factoid Answer Selection (2015)
27. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of CNN and RNN for natural language processing (2017). <https://arxiv.org/abs/1702.01923>
28. Shao, T., Kui, X., Zhang, P., Chen, H.: Collaborative learning for answer selection in question answering. *IEEE Access* **7**, 7337–7347 (2019). <https://doi.org/10.1109/ACCESS.2018.2890102>
29. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: Proceedings of International Conference on Learning Representations, pp. 1–16 (2016)
30. Feng, M., Xiang, B., Glass, M.R., Wang, L., Zhou, B.: Applying deep learning to answer selection: a study and an open task. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 813–820, December 2015
31. Tran, Q.H., Tran, D.V., Vu, T., Le Nguyen, M., Pham, S.B.: JAIST: Combining multiple features for Answer Selection in Community Question Answering, pp. 215–219 (2015). <https://doi.org/10.18653/v1/S15-2038>
32. Hou, Y., Tan, C., Wang, X., Zhang, Y., Xu, J., Chen, Q.: HITSZ-ICRC: Exploiting Classification Approach for Answer Selection in Community Question Answering, pp. 196–202 (2015). <https://doi.org/10.18653/v1/S15-2035>

33. Nicosia, M., Filice, S., Barrón-Cedeno, A., Saleh, I., Mubarak, H., Gao, W., Nakov, P., Martino, G.D.S., Moschitti, A., Darwish, K., Marquz, L.M.: QCRI: Answer Selection for Community Question Answering – Experiments for Arabic and English (2015). <https://doi.org/10.18653/v1/S15-2036>
34. Zhou, X., Hu, B., Lin, J., Xiang, Y., Wang, X.: ICRC-HIT: A Deep Learning based Comment Sequence Labeling System for Answer Selection Challenge, pp. 210–214 (2015). <https://doi.org/10.18653/v1/S15-2037>
35. Fan, H., Ma, Z., Li, H., Wang, D., Liu, J.: Enhanced answer selection in CQA using multi-dimensional features combination. *Tsinghua Sci. Technol.* **24**, 346–359 (2019). <https://doi.org/10.26599/TST.2018.9010050>
36. Elalfy, D., Gad, W., Ismail, R.: A hybrid model to predict best answers in question answering communities. *Egypt. Inform. J.* **19**(1), 21–31 (2018). <https://doi.org/10.1016/j.eij.2017.06.002>. ISSN 1110-8665
37. Tian, Q., Zhang, P., Li, B.: Towards predicting the best answers in community-based question-answering services. In: *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, pp. 725–728 (2013)
38. Lin, C., Richi, N.: Leveraging the network information for evaluating answer quality in a collaborative question answering portal. *Soc. Netw. Anal. Min.* **2**(3), 197–215 (2012)
39. Eskandari, F., Shayestehmanesh, H., Hashemi, S.: Predicting best answer using sentiment analysis in community question answering systems. In: *2015 Signal Processing and Intelligent Systems Conference (SPIS)*, Tehran, pp. 53–57 (2015). <https://doi.org/10.1109/SPIS.2015.7422311>
40. Gkotsis, G., Stepanyan, K., Pedrinaci, C., Domingue, J., Liakata, M.: It’s all in the content: state of the art best answer prediction based on discretisation of shallow linguistic features. In: *Proceedings of the 2014 ACM Conference on Web Science*, pp. 202–210 (2014)
41. Tian, Q., Zhang, P., Li, B.: Towards predicting the best answers in community-based question-answering services. In: *ICWSM* (2013)
42. Sahu, T.P., Nagwani, N.K., Verma, S.: Selecting best answer: an empirical analysis on community question answering sites. *IEEE Access* **4**, 4797–4808 (2016). <https://doi.org/10.1109/ACCESS.2016.2600622>
43. Anusha, J., Rekha, V.S., Sivakumar, P.B.: A machine learning approach to cluster the users of stack overflow forum. In: Suresh, P.L., Sekhar, D.S., Ketan, P.B. (eds.) *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014*, vol. 2, pp.411–418. Springer, India, New Delhi (2015)
44. Sinha, V.S., Mani, S., Gupta, M.: Exploring activeness of users in QA forums. In: *Proceedings of 10th Workshop Conference Mining Software Repositories*, pp. 77–80 (2013)
45. Bouguessa, M., Dumoulin, B., Wang, S.: Identifying authoritative actors in question-answering forums: the case of Yahoo! answer. In: *Proceedings of 14th ACM SIGKDD International Conference Knowledge Discovery Data Mining*, pp. 866–874 (2008)
46. Yang, B., Manandhar, S.: Tag-based expert recommendation in community question answering. In: *Proceedings IEEE/ACM International Conference Advance Social Network Analysis Mining (ASONAM)*, pp. 960–963, August 2014
47. Barua, A., Thomas, S.W., Hassan, A.E.: What are developers talking about? An analysis of topics and trends in stack over_ow. *Empirical Softw. Eng.* **19**(3), 619_654 (2014)
48. Yang, L., et al.: CQArank: jointly model topics and expertise in community question answering. In: *Proceedings 22nd ACM International Conference on Information & Knowledge Management*, pp. 99–108 (2013)
49. Tian, Q., Zhang, P., Li, B.: Towards predicting the best answers in community-based question answering service. In: *Proceedings of ICWSM*, pp. 725–728 (2013)

50. Shah, C., Pomerantz, J.: Evaluating and predicting answer quality in community QA. In: Proceedings of 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 411–418 (2010)
51. Zhao, J., Zhu, T.T., Lan, M.: ECNU: One Stone Two Birds: Ensemble of Heterogenous Measures for Semantic Relatedness and Textual Entailment, pp. 271–277 (2014). <https://doi.org/10.3115/v1/S14-2044>
52. Belinkov, Y., Mohtarami, M., Cyphers, S., Glass, J.: VectorSLU: a continuous word vector approach to answer selection in community question answering systems. In: Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, CO, USA, pp. 282–287 (2015)



Shallow over Deep Neural Networks: A Empirical Analysis for Human Emotion Classification Using Audio Data

Chandresh S. Kanani^(✉), Karanjit Singh Gill^(✉), Sourajit Behera^(✉),
Anurag Choubey^(✉), Rohit Kumar Gupta^(✉), and Rajiv Misra^(✉)

Department of Computer Science and Engineering,
Indian Institute of Technology Patna, Patna 801106, India
{sourajit.pcs17, anurag.pcs17, rajivm}@iitp.ac.in

Abstract. Human emotions can be identified in numerous ways, ranging from analyzing the tonal properties of speech to the facial expressions created before speech delivery and even the body gestures that can suggest various emotions without saying anything. Knowing the correct emotions of an individual can help us understand the situation and even react to it. This phenomena is even true for many feedback systems used for day-to-day communication with humans, specifically the ones used for smart home solutions. The field of automated emotion recognition involves use-cases in different fields of research from computer vision, physiology to even artificial intelligence. This work focuses on classifying emotions into eight categories which are neutral, happy, sad, angry, calm, fearful, disgust and surprised based on the way those sentences have been spoken, using the “Ryerson Audio-Visual Database of Emotional Speech and Song” (RAVDESS). We propose a novel approach for emotion classification of audio conversations based on speech signals. Acoustic properties based emotion classification is independent of any spoken language and it can be used for cross-language emotion classification. The aim of the contribution was to develop a system capable of automatically recognising emotions for real-time speech. We performed several simulations and were able to achieve the highest accuracy of 82.99% with our shallow CNN model.

Keywords: Deep learning · Shallow network · Computer vision · Speech · Emotion

1 Introduction

Human-Machine interaction (HMI) has enabled humans to interact with machines and computing devices, from touchscreen, automated switches to punch cards. It uses the technologies that exploit the nature of light, sound or touch etc. [10]. Human speech recognition systems have the potential to serve as

the most fruitful way of HMI systems [17]. Speech recognition systems for HMI are significantly used for identifying the most intrinsic traits of living understanding and is used as a very efficient security system. Different audio signals are used to identify the nature of speech to create a more trustable and helpful unique information system [23].

With the advent of smart home online services, speech recognition systems have become a path-breaker among smart solution enabling companies, who are constantly trying to evaluate the human behavior and living through various data processing feature and introducing this knowledge to identify the behaviors at the most granular level. Human emotion recognition through speech is one of such intriguing functionalities which deep learning technologies can enable for building a smart future. Identifying the emotional state of a speaker from speech signal is a very dynamic research area under the Automatic speech emotion recognition (SER) system. Tonal properties, facial expressions, body gestures etc. are few of the properties used to recognize emotions [8]. In the era of computer vision, Artificial intelligence and HMI, the identification of accurate human emotion, will play a significant role in enriching human-machine interaction. Speech is generally categorized as voiced and unvoiced based on the periodic nature of the signal attained after representation. These speeches are then preprocessed with noise and silent region removal to finally attain features with wide variety of physiological, mental and physical events understanding.

To study the availability and validity of a reliable emotional expressions, RAVDESS [11], a public dataset is prepared containing audiovisual recordings of speakers in North American English as speeches and songs [11]. In this work, we have just considered the audio recordings of the RAVDESS dataset to create a system capable of classifying the different human emotions using the shallow networks of deep learning infrastructure as a computer vision problem. We through our understanding of the already available deeper neural networks in computer vision suggest that under limited training samples and systematically created speech dataset, shallow networks perform better than deep networks in terms of accuracy besides being lightweight.

In the next section, we discuss the earlier works done in the field of HMI, more specifically in audio based human speech recognition. Section 3 discusses about the RAVDESS dataset and the preparation steps involved in it. Section 4 discusses the different benchmark approaches used in analyzing a speech and the methods used in each. Section 5 discusses the proposed methodology and problem formulation of the emotion recognition system. In Sect. 6, we discuss the results and other associated parameters. Finally, Sect. 7 gives a conclusion of the paper with a brief on the our contribution as well as future possibilities of improving upon the achieved results.

2 Related Work

Several works have used speech signals to process emotions with raw processing of the audio waves with different machine learning and deep learning and only

a few have used the wave front images of the speech signals to learn the various emotions as part of computer vision. MFCC (Mel-Frequency cepstral coefficient), entropy, zero crossing rates, pitch, energy etc. are the most common feature used for processing audio signals. Moore et al. [14] proposed a powerful speech recognition system called “PRESENCE”, which is a pioneer in speech science technology that includes psychological and human spoken language behavior. Sun et al. [20], Sheikhan et al. [18] improved the power of speech recognition systems, but on an older recognition system. Iqbal et al. [8] proposed a gradient boosted tree based real-time speech recognition system for English language on the verbal emotion database. But, their work is limited to only four class classification. Munot et al. [15] identifies how emotional content in speech impacts speech recognition for commercial systems by studying three different audio datasets. But, this works is just an observatory implication on the different emotions across the different datasets with the use of any artificial intelligence technology.

3 Dataset Description

The RAVDESS Dataset has five distinguished features [11], with most of the sets containing less than 200 clips in the form of a dynamic, multimodal structure. The original construction of this datasets contains three modality formats namely for audio-only, audio-visual and video-only. In the RADVESS dataset, each actor has to perform 8 emotions by saying two sentences. As a result, each actor would induce 2 samples for each emotion. The database contains 24 professional actors (12 females, 12 males), vocalizing two lexically-matched statements in a neutral North American accent. Each audio wave is around 4 s, the first and last second are most likely silenced. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Also, each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression. Intensity being used as the most salient feature in speech recognition is used in a multi-dimensional emotion space [11]. RAVDESS lexically matches the speeches and songs presented in the database. In our analysis of the RAVDESS, the speech files collectively contains 1440 files: 60 trials per actor \times 24 actors = 1440. The standard sentences are “Kids are talking by the door” and “Dogs are sitting by the door”.

4 Baseline Architectures

In this section, we discuss about the set of baseline architectures utilized for solving the multi-class classification problem. There are four baseline architectures we have formulated under three different sections in this work, namely the Multi-Layer perceptron (MLP), the Recurrent Neural Network (RNN), VGG-16 and Inception-V3. We compare all these architectures with our proposed approach CNN-X.

4.1 Numerical Approach

Multi-layer Perceptron (MLP). Artificial neural networks (ANN) [12] with their inherent abilities for learning, recognizing and tackling complicated tasks are one of the most sought of models utilized for solving various supervised/unsupervised complex problems [6,22]. Single layer perceptron (SLP), an abstract version of ANNs with two input and output layers [7], however has been proved to be unable for handling non-linearly separable designs in the data [16]. Therefore, in order to overcome the shortcomings of SLPs, a feed-forward neural network based on ANNs having one or several hidden layers called MLP has been proposed [2].

In this approach, we fed each audio sample into the librosa package [13] and extracted the following numerical features: 1) Mel-frequency cepstral coefficients 2) Power spectrum 3) Spectral contrast 4) Tonal centroid features 5) Roll-off frequency 6) Spectral flatness respectively which are input to the MLP network whose structure is outlined in Sect. 6.2 and presented in Fig. 7 for emotion detection.

4.2 Time-Series Model

Recurrent Neural Network. RNN, a deep learning model is usually adapted to sequential data and is one of most sought of approach for natural language processing (NLP) and speech recognition problems [4]. RNNs are a form of (ANNs) with advantages including modelling of time series while catering to structural dependency in long sequential data [5]. With an additional time delay unit along with feedback connections, information from prior state can be utilized for next state in RNN.

In this approach we feed each audio file into the librosa and fetch the Mel-frequency cepstral coefficients sampled at specific window size which are set as input to the RNN.

4.3 Computer Vision

In our computer vision approach, we first decide the image dimensions. We tried all the models which we will mention here later for both RGB and grayscale versions of the image. We also tried different image dimensions for simulations. Figure 1 shows overall architecture for emotion classification system using computer vision models.

VGG-16. VGG16, a famous CNN model was submitted to ILSVRC-2014 [19]. The input of convolution layer is 224×224 RGB image. This image passes through stacked convolutional layers. With a convolution stride of 1 pixel, the spatial resolution is made to be preserved after convolution. Five max-pooling layers with a window size of 22 pixels and stride factor 2 are used to perform spatial pooling which follows few of the convolutional layers. Max-pooling is

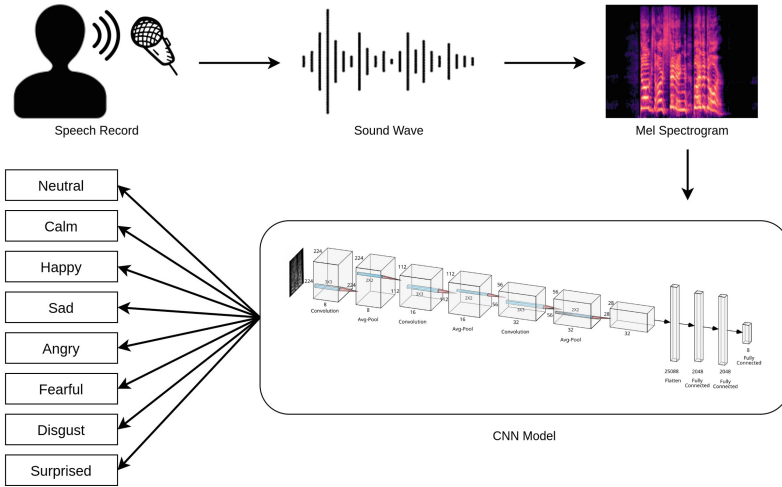


Fig. 1. Overall architecture of speech emotion recognition system using CNN models.

performed over a 2×2 pixel window, with stride 2. Three fully connected layers with 4096, 4096 and 1000 channels followed by a softmax layer are stacked after the convolutional layers. The first two have 4096 channels in each, the third contains 1000 channels and the final layer is a soft max layer. ReLU activation function has been used for all hidden layers. Figure 2 shows the overall architecture of VGG-16 network. We have also tried VGG-16 model with average pooling instead of max pooling; results for the same are listed with name “VGG-16 AvgPool” in Table 1.

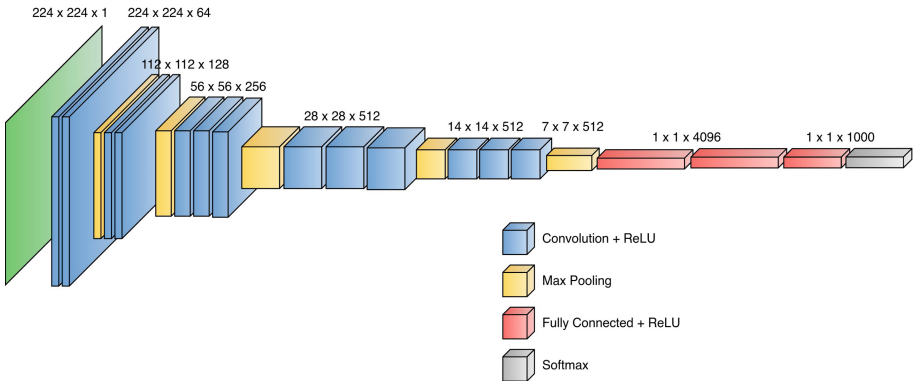


Fig. 2. VGG16 architecture

Inception V3. Inception V3, a third variant deep CNN by google for image classification is based on “Rethinking the Inception Architecture for Computer Vision” [21]. The model comprises of several building blocks including symmetric and asymmetric convolutions, average and max pooling, concatenation, dropouts along with fully connected layers as shown in Fig. 3.

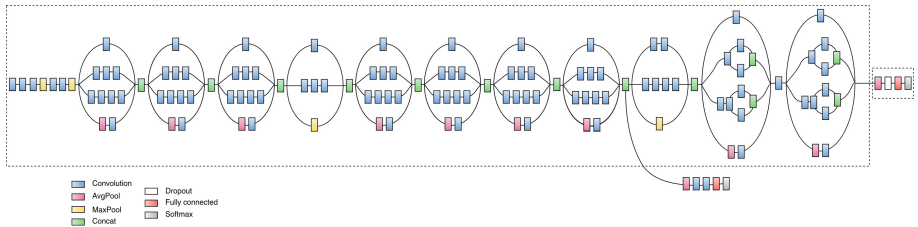


Fig. 3. Inception-V3 architecture

5 Experimental Setup

5.1 Experimental Settings

The entire dataset used for experimentation in this paper uses audio signals described in Sect. 3. Random combination of the signals are performed to acquire a shuffled dataset. Each of the signals are fed into librosa package with default parameters to generate Mel Spectrogram images of dimensions 224×224 which are used as input for the CNN models. The images are then split into training, validation and testing datasets in the ratio 7:1:2. We conduct several trials and report the average values of the evaluation metrics. Python language using “Keras” library through “TensorFlow” back-end via functional API is used for implementing the network architecture.

5.2 Network Structure of CNN-X

The proposed structure of CNN-X is comprised of a set of convolution, pooling and fully connected layers. Three pairs of convolution-pooling layers are stacked together to construct a network for effective features extraction. Average pooling is applied in the CNN-X structure. While designing the network, we stopped the pooling operations after the third average pooling layer as the activations after this layer converged. Figure 4 presents the output activations after the fourth pooling layer.

It can be observed that, around 20 kernels in 4th convolution layer does not learn any significant features; while the activation outputs for the 3rd convolution layer represented in Fig. 5 suggests most of the 32 kernels learn meaningful features which can be further captured using subsequent fully connected layers.

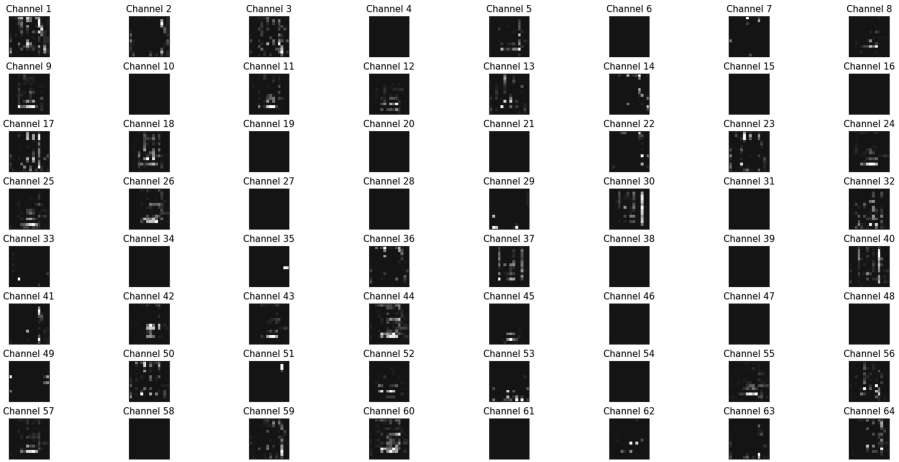


Fig. 4. Activations after fourth pooling layers

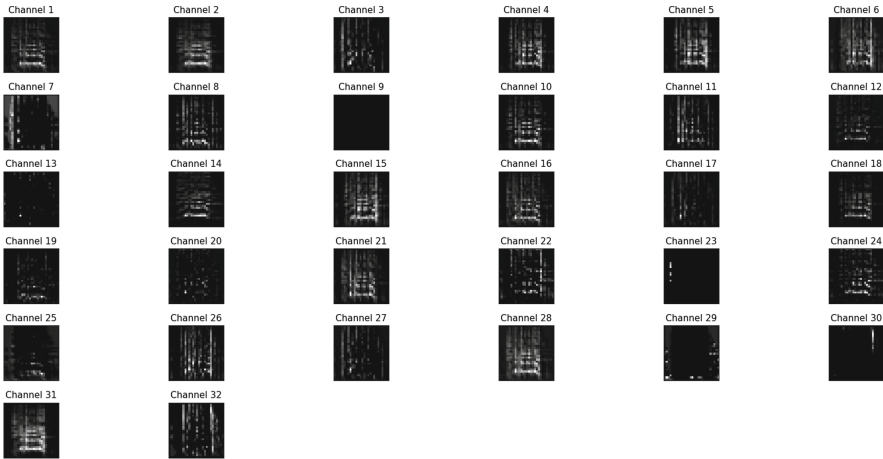


Fig. 5. Activations after third pooling layers

All the convolution layers are applied with the same kernel size which holds true for pooling layers as well. Figure 6 represents the overall architecture of the proposed CNN-X model with each layer named by their specific operations performed.

The first hidden layer is a convolutional layer with 8 channels and a kernel size of 3×3 , it takes as input an image of dimensions 224×224 . These are then fed into an average pooling layer with kernel size 2×2 . This results in an output dimensions of 112×112 which is fed to another convolution layer with a kernel of size 3×3 and 16 channels. The outputs are then fed to an average pooling layer with kernel size 2×2 generating outputs with dimension $56 \times$

56 with 16 channels. Finally the third convolution layer with a kernel of size 3×3 and 32 channels is fed with the input from the previous average pooling layer to produce an output dimension of $56 \times 56 \times 32$. This is followed by an average pooling layer with kernel of 2×2 to output dimensions of $28 \times 28 \times 32$ which are flattened into a 25,088 long one-dimensional feature vector. The feature vector is then passed through two fully connected layers with 2048 hidden neurons each followed by a fully connected layer with 8 hidden neurons subjected to the softmax activation function to predict the probabilities of the 8 emotions respectively.

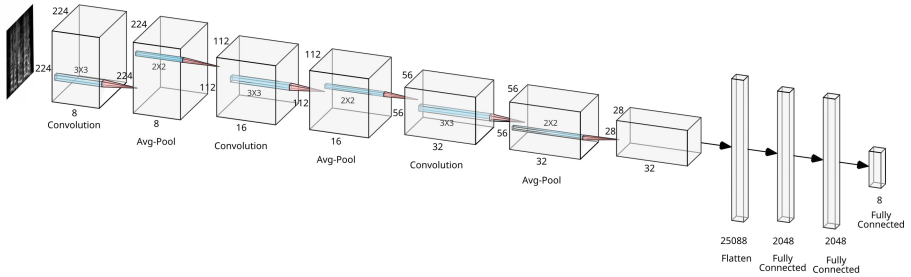


Fig. 6. CNN-X layer wise architecture

5.3 Hyper-parameter Tuning

In our experiments, we tune a specific set of hyper-parameters which play a significant role in learning high-level features representing different emotions. With a batch size of 32, a learning rate of 0.001, momentum of 0.8 and rmsprop as optimizer the network is trained for 20 epochs. Each of the convolution operation and fully connected layers except the last one are treated with the ReLU activation function.

5.4 Evaluation Metrics

Accuracy, precision, recall and F1-score are taken as the benchmark performance indicators on the emotion recognition dataset for all the model comparisons. We are solving a multi-class classification problem where we consider unweighted mean parameter for calculating the average for this classification. The performance indicators for evaluating the models are done using a standard confusion matrix [1].

6 Result and Discussion

6.1 Selecting the Appropriate Input Image Dimensions

It has been widely acknowledged that, using the right dimensions of the input image has a significant effect on feature extraction, evaluation metrics and results

of the network. Thus, for obtaining the optimal trade-off between the evaluation metrics of our problem and the image size, we have performed simulations using various image dimensions on different computer vision methods described in Sect. 4.3. Table 1 presents the comparison of test accuracy among different computer vision approaches with respect to different input image dimensions. Grayscale and RGB versions are used for selecting the appropriate image dimensions.

Table 1. Comparison between various computer vision approaches

Name	Num channels	Image size	Accuracy	Precision	Recall	F1-score
VGG-16	1	224×224	64.93%	0.69	0.64	0.65
	3	224×224	55.90%	0.61	0.55	0.53
VGG-16 AvgPool	1	224×224	54.17%	0.60	0.55	0.53
	3	224×224	48.96%	0.53	0.48	0.47
InceptionV3	1	224×224	46.18%	0.47	0.47	0.47
	3	224×224	47.92%	0.45	0.46	0.43

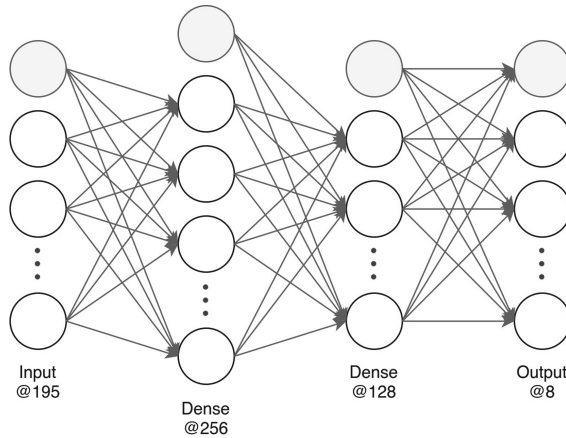
Table 2. Results for various input image sizes on CNN-X with 2 Convolution + Pooling and 3 Fully Connected layers.

Num channels	Image size	Accuracy	Precision	Recall	F1-score
1	32×32	65.97%	0.67	0.66	0.64
	64×64	81.94%	0.81	0.80	0.80
	128×128	77.08%	0.77	0.77	0.77
	224×224	63.89%	0.65	0.64	0.64
3	32×32	60.10%	0.61	0.60	0.59
	64×64	63.89%	0.67	0.63	0.61
	128×128	76.04%	0.76	0.76	0.76
	224×224	68.05%	0.66	0.67	0.66

CNN-X with a 6 layers (*3 Convolution + Average Pooling, 3 Fully Connected*) architecture is a shallow architecture over VGG-16 with a 16 layers and InceptionV3 with 48 layers. Thus, with a shallow configuration we are able to achieve an improved accuracy by a factor of 18.06% and 35.07% over VGG-16 and InceptionV3 models. Our experiments suggests the effectiveness of the model with an input dimension of 224×224 grayscale image. These dimensions are used as input to our proposed CNN-X model. It can be noted from Table 2 and Table 3 that, when we reduce the number of convolution layers beyond 3, the performance decreases.

Table 3. Results for various input image sizes on CNN-X with 3 Convolution + Pooling and 3 Fully Connected layers

Num channels	Image size	Accuracy	Precision	Recall	F1-score
1	32×32	53.12%	0.60	0.54	0.53
	64×64	75.00%	0.74	0.74	0.74
	128×128	79.17%	0.80	0.78	0.77
	224×224	82.99%	0.82	0.82	0.82
3	32×32	50.00%	0.52	0.47	0.48
	64×64	61.11%	0.62	0.60	0.59
	128×128	67.01%	0.68	0.67	0.66
	224×224	70.14%	0.73	0.70	0.69

**Fig. 7.** Overall architecture numerical approach**Table 4.** Accuracy comparison between proposed approach and other baseline architectures

Model	Accuracy	Precision	Recall	F1-score
Time-series (RNN)	52.08%	0.46	0.48	0.47
Numerical (MLP)	75.00%	0.71	0.69	0.70
Computer-vision (CNN-X)	82.99%	0.82	0.82	0.82

6.2 Comparison with Other Baseline Architectures

We perform a set of experiments to take into account other baselines with hand-crafted features described in Sect. 4.1 and Sect. 4.2 respectively. Separate features for each approach are extracted and fed into multi-layer perceptron (MLP) and recurrent neural network respectively (RNN). Figure 7 represents the network

structure of the numerical approach utilized in this paper. A feature set of size 195 is fed into the MLP followed by two dense layers with 256 and 128 neurons. The features extracted are then fed into the final fully connected layer with 8 hidden neurons respectively. In the time-series approach we take samples of specific window size from the “Mel-frequency cepstral coefficients” feature of the audio file and feed into the RNN. Table 4 presents the result comparison between the proposed computer vision approach (CNN-X) and other baselines. It can be observed that CNN-X comfortably overpowers the numerical approach (MLP) by a factor of 7.99% and time-series approach (RNN) by 30.91% suggesting the effectiveness of CNN-X over the other approaches.

6.3 Comparison Among Computer Vision Models

Table 5, presents the comparison of the classification accuracies among the computer vision baseline architectures described in Sect. 4.3. It can be observed that CNN-X, our proposed approach comfortably out-performs VGG-16 and InceptionV3 by significant margins of 18.06% and 35.07% respectively. This shows the effectiveness of our proposed approach to achieve a higher accuracy while training the model with a shallow architecture over deep architectures of VGG-16 and InceptionV3; suggesting CNN-X to be a viable approach given the memory constraints and image dimensions trade-off.

Table 5. Comparison between best results obtained from various computer vision approaches.

Model	Accuracy	Precision	Recall	F1-score
VGG-16	64.93%	0.69	0.64	0.65
InceptionV3	47.92%	0.45	0.46	0.43
CNN-X	82.99%	0.82	0.82	0.82

6.4 Comparison with State-of-the-art Approaches

Table 6 presents the accuracy comparison between the proposed approach and state-of-the-art benchmarks on the same dataset. It can be observed that CNN-X without any bias restriction on the number of classes to be predicted significantly outperforms two other benchmarks with no restrictions: 1) Livingstone and Rasso [11] approach by a factor of 20.99% and 2) Munot and Nenkova [15] by 26.61% respectively. Baobin Li et al. [3] with a prediction probability for 6 classes, Aseef Iqbal and Kakon Barua [9] with a restriction of 4 classes classification come near to CNN-X in terms of accuracy and are marginally less by a factor of 3.99% and 2.99% respectively. However, our proposed approach is able to achieve better accuracy for all the 8 class prediction problem suggesting the capability and efficiency of CNN-X over state-of-the-art benchmarks.

Table 6. Accuracy comparison between proposed approach and other state-of-the-art benchmarks

Authors	Year	Bias	Accuracy
Baobin Li et al. [3]	2017	6 classes	79%
Livingstone and Russo [11]	2018	–	62%
Aseef Iqbal and Kakon Barua [9]	2019	4 classes	80%
Munot and Nenkova [15]	2019	–	56.38%
Proposed CNN-X	2020	–	82.99%

7 Conclusion and Future Work

With the advent of IoT for smart home solutions, HMI based human emotion recognition via speech recognition will play a significant role in the future. RAVDESS dataset is a significant step towards building a multi-model environment, with use of both audio-visual data. In this work, we have tried to use the audio-only data from the RAVDESS information set to create a visual understanding for accurate emotion detection using three different types of baseline architectures namely, numerical approach, time-series model and the popular computer vision approaches. Through our result we have been able to show that computer vision approaches supersedes the other two techniques. Also, within the three computer vision approaches, CNN-X, a rather shallow approach works much better than the already established pre-trained approaches with much deeper neural networks. Thus, under the observed setting we can conclude that for any real-time smarter systems, shallow networks can provide as a more accurate and faster solution, specially when processed under limited infrastructure. In future, we would also like to consider the visual expression enacted by the different actors to suppose different emotions while delivering a speech and create a multi-task network under a similar shallow neural network setting.

References

1. Behera, S., Choubey, A., Kanani, C.S., Patel, Y.S., Misra, R., Sillitti, A.: Ensemble trees learning based improved predictive maintenance using IIoT for turbofan engines. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 842–850 (2019)
2. Chen, J.F., Do, Q.H., Hsieh, H.N.: Training artificial neural networks by a hybrid PSO-CS algorithm. *Algorithms* **8**(2), 292–308 (2015)
3. Gao, Y., Li, B., Wang, N., Zhu, T.: Speech emotion recognition using local and global features. In: International Conference on Brain Informatics, pp. 3–13. Springer (2017)
4. Goldberg, Y.: Neural network methods for natural language processing. *Synthesis Lect. Hum. Lang. Technol.* **10**(1), 1–309 (2017)
5. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint [arXiv:1308.0850](https://arxiv.org/abs/1308.0850) (2013)

6. Heidari, A.A., Abbaspour, R.A.: Enhanced chaotic grey wolf optimizer for real-world optimization problems: a comparative study. In: *Handbook of Research on Emergent Applications of Optimization Algorithms*, pp. 693–727. IGI Global (2018)
7. Heidari, A.A., Faris, H., Aljarah, I., Mirjalili, S.: An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft. Comput.* **23**(17), 7941–7958 (2019)
8. Iqbal, A., Barua, K.: A real-time emotion recognition from speech using gradient boosting. In: *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–5, February 2019. <https://doi.org/10.1109/ECACE.2019.8679271>
9. Iqbal, A., Barua, K.: A real-time emotion recognition from speech using gradient boosting. In: *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–5. IEEE (2019)
10. Karray, F., Alemzadeh, M., Saleh, J.A., Arab, M.N.: *Human-computer interaction: overview on state of the art* (2008)
11. Livingstone, S.R., Russo, F.A.: The ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. *PloS one* **13**(5), e0196391 (2018)
12. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133 (1943)
13. McFee, B., Rael, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: *librosa: audio and music signal analysis in python*. In: *Proceedings of the 14th python in science conference*, vol. 8 (2015)
14. Moore, R.: Presence: a human-inspired architecture for speech-based human-machine interaction. *IEEE Trans. Comput.* **56**(9), 1176–1188 (2007)
15. Munot, R., Nenkova, A.: Emotion impacts speech recognition performance. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 16–21 (2019)
16. Ojha, V.K., Abraham, A., Sňášel, V.: Metaheuristic design of feedforward neural networks: a review of two decades of research. *Eng. Appl. Artif. Intell.* **60**, 97–116 (2017)
17. Reddy, D.R.: Speech recognition by machine: a review. *Proc. IEEE* **64**(4), 501–531 (1976)
18. Sheikhan, M., Gharavian, D., Ashoftedel, F.: Using DTW neural-based MFCC warping to improve emotional speech recognition. *Neural Comput. Appl.* **21**(7), 1765–1773 (2012)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)* (2014)
20. Sun, Y., Zhou, Y., Zhao, Q., Yan, Y.: Acoustic feature optimization for emotion affected speech recognition. In: *2009 International Conference on Information Engineering and Computer Science*, pp. 1–4. IEEE (2009)
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826 (2016)
22. Trujillo, M.C.R., Alarcón, T.E., Dalmau, O.S., Ojeda, A.Z.: Segmentation of carbon nanotube images through an artificial neural network. *Soft. Comput.* **21**(3), 611–625 (2017)
23. Yu, Y.: Research on speech recognition technology and its application. In: *2012 International Conference on Computer Science and Electronics Engineering*, vol. 1, pp. 306–309. IEEE (2012)



k Stacked Bidirectional LSTM for Resource Usage Prediction in Cloud Data Centers

Yashwant Singh Patel^{1(✉)}, Rishabh Jaiswal², Savyasachi Pandey³,
and Rajiv Misra¹

¹ Department of Computer Science and Engineering, Indian Institute of Technology
Patna, Patna, India

{yashwant.pcs17,rajivm}@iitp.ac.in

² Department of Computer Science and Engineering, Medi-Caps Institute of
Technology and Management, Indore, India

³ Department of Computer Science and Engineering, SRM Institute of Science
and Technology, Kattankulathur, India

Abstract. Cloud computing leverages virtualization as the most popular technique to deploy enterprise applications on virtual machines. Since the cloud system dynamically adapts to workload changes depending on the time of the day. It is required to ensure elasticity as a robust technique to efficiently model the changing workload requirements. However, it is an extremely challenging task, as several users may enter and depart from the cloud system over time. Predicting the different resource usage metrics of dynamically arriving jobs can help the cloud service providers (CSPs) in better capacity planning to fulfill the service level agreements (SLAs). In this paper, we propose a k clustering-based stacked bidirectional LSTM (BiLSTM) deep learners to model the multivariate resource usage predictions for highly varying cloud workloads. We evaluate the proposed model on the Google cluster trace and validate its performance with the current approaches.

Keywords: Cloud workload prediction · Deep learning · Time-series data · Long short-term memory (LSTM) · K-means

1 Introduction

Cloud computing is growing rapidly as a successful paradigm for providing IT infrastructures, on-demand computing resources, and services [1]. Notably, the extensive adoption of virtualization technologies to deploy diverse applications on virtual machines (VMs) has led to the development of large scale data centers. Moreover, the workload at cloud data centers plays a unique role in maximizing the energy-efficiency and assuring the quality of services (QoS) [2]. Since the cloud workload dynamically changes, as a large number of cloud users may

instantaneously demand the services at one period, and few users may request the resources during other periods. If the cloud service provider (CSP) is unable to provide adequate resources in the course of high utilization periods, it will drastically reduce the QoS and generate the service-level agreement (SLA) violation. However, if the CSPs offer too much and excess cloud resources all the time, it may consume more energy and generate an additional cost, particularly in the time of low-demand periods. Due to the resource demand uncertainties at different time slots, it is complicated for the CSP to perform efficient resource allocation. Therefore, elasticity has become one of the key features of the cloud to adapt itself to meet the varying workload demands automatically. It allows the provisioning of dynamic resources based on the workload of VMs to cut down the overall cost and to maximize the QoS. Hence, the workload prediction helps the CSPs in better decision making and dynamic scaling of resources. Resource prediction problem in the cloud can be conceptualized as a time series forecasting problem, where we can analyze the past cloud resource usage and predict future demand for resources. In literature, various statistical models such as ARIMA (Autoregressive integrated moving average) [3], Feed-forward neural network [4], and Markov models [5,6] have proposed for forecasting of resource usage in the cloud data centers. Most of the existing methods have presumed that resource usage observed in a long period are independent of each other. But Ghorbani et al. [7] discovered the long range dependence issue in the cloud using Google cluster trace [8]. Song et al. [9] presented a univariate Long-short term memory (LSTM) based network to solve the problem of long range dependence. They have used only the CPU usage values to predict future trends in CPU usage. But the different resource metrics are inter-related to each other and may also affect the performance of desired resource metrics. Thus, the analysis of multivariate time series prediction is necessary to obtain correct behavior. In this work, we have investigated the multivariate time series based deep BiLSTM model on analyzing the effect of multiple resource metrics. We have applied different resource metrics for CPU and memory resource usage such as memory utilization, maximum memory utilization, assigned memory utilization, disk space utilization, maximum CPU utilization, cycles per instruction (CPI) and also the memory accesses per instruction in addition to CPU utilization. In this paper, first, we divide the time series data of resource usage into several equal-length sequences. We apply the K-means clustering algorithm for the classification of sub-sequences into different classes. Finally, we construct k stacked BiLSTM models for all the clusters. We use the Google cluster trace for cloud workloads modeling.

2 Related Work

Zhenhuan et al. [5] have proposed predictive elastic resource scaling (PRESS) approach, which applies signature and state-driven schemes for future resource utilization prediction. The state drive scheme applies a discrete-time markov chain model for short term prediction of future resource values in the Google cluster trace. For a resource metric such as CPU or memory usage, it divides the values into k equal size bins, where bin represents a different state. It uses

the transition probability matrix Tp of $k \times k$ matrix, where any element Tp_{ij} at row i and j represents the conditional probability for transition from state i to state j . For predicting the future value, it uses Chapman-Kolmogorov equation as:

$$\psi_t = \psi_{(t-1)}Tp \tag{1}$$

Here ψ_t and $\psi_{(t-1)}$ represents the probability distribution of state at time slot t and $t - 1$ respectively.

AGILE [6] is a extension of PRESS technique. It uses wavelets along with the Markov model for medium-term demand prediction of resources for cloud. The key concept of the AGILE approach is to first break down the time-series demand into a set of wavelet driven signals. After that, it performs predictions on individual break down signals for future resource estimation using the Markov model. At last, the future resource demand is synthesized by combining the predictions of all the individual signals from wavelets. Zhang et al. [3] have proposed the ARIMA based model to forecast the value of resource usage. It uses the last t observations of H_t to predict H_{t+1} , which is a estimate resource usage at time $t + 1$.

$$H_{t+1} = \xi_0 H_t + .. + \xi_{n-1} H_{t-m+1} + \epsilon_{t+1} + \delta_0 \epsilon_t + .. + \delta_{q-1} \epsilon_{t+1-q}, \tag{2}$$

where the term $\xi + i$ and δ_j denotes constants. ϵ_t is the independent error term. The other parameters m and q denotes the total number of lags or the last measured resource usage values and total error terms respectively. ARIMA has become a popular time series prediction model but it fails to detect the non linear patterns present in the time series data. Caglar et al. [4] proposed an iOverbook model. It applies a two-layer feedforward neural network for prediction. It generalizes the linear and non linear correlation connecting the input and output having a single hidden layer.

$$y'_t = \sum_{t=1}^m \eta_t \mathcal{Y} \left(\sum_{j=1}^n \omega_{jt} y_j \right) \tag{3}$$

Here y represents input, n denotes the total number of lags, m represents the total neurons present in the hidden layer, $\mathcal{Y}(\cdot)$ denotes the activation function, and η, ω denotes the training weight parameters. The techniques, as mentioned earlier, take the assumption that the time series is memory-less & stationary. Although Ghorbani et al. [7] studied the presence of long range dependence (LRD) in the workloads, the LRD is a circumstance that may arise during the time-series analysis, where several past time lags affect an increase or decrease present in the next-step value. It is mostly related to the decay rate of statistical dependence having 2 points with an escalating time-interval. The LRD exists if the dependence decays more gradually than an exponential decay, mostly a power-like decay. Intuitively, LSTM models are more suitable to handle the long-range dependence. Song et al. [9], used the LSTM network for forecasting the mean load over successive upcoming time-intervals and also the multi step ahead specific load. However, they have used the past trends of CPU resource usage

only. In [10], the authors have observed that to perform prediction of demand in the department of emergency at Western Australia, the multivariate vector autoregressive moving average (ARMA) approaches to deliver high performance than applying univariate ARMA schemes. Therefore, we use multi-variate time series analysis for future CPU and memory resource utilization demands. We analyze the effect of various resource utilization metrics such as memory utilization, assigned memory utilization, disk I/O time, and used disk space, etc. along with the CPU utilization. We build multivariate LSTM for future cloud workload prediction. Current literature is mostly focused on the unidirectional LSTM to predict future events. However, it only preserves the information of past resource usage. In this paper, we present the k stacked BiLSTM model for out-of-sample multivariate resource usage predictions. The stacked BiLSTM model runs the input in two ways, one from past to future and another one from future to past. Wakuya et al. [12] used the bi-directional computation for time-series forecasting of laser data. Experimentally, it is observed that the quality of the prediction model is far superior to the unidirectional model. Zhao et al. [13] applied a bidirectional time series model based on the extreme learning machines approach for wind farm production. Through simulations, it is observed that the forecast accuracy is improved in the bidirectional mechanism. Gupta et al. [11] proposed multivariate LSTM and BiLSTM models for cloud workload prediction using Google cluster trace. The application of stacked Bi-LSTM deep learners has not been presented for resource utilization of future cloud workloads. To enhance the prediction accuracy, we first apply K-means clustering to classify the time series data into k clusters then apply k stacked Bi-LSTM deep learners on each cluster. Further, we analyze the prediction performance of the proposed model with the other existing deep learning approaches.

3 Architecture of Proposed k Stacked BiLSTM Model

3.1 BiLSTM (Bidirectional LSTM)

Bi-LSTM deep learning model is a combination of Bi-directional Recurrent Neural Networks (BiRNN) and Long Short-Term Memory (LSTM) networks. It takes the input sequence in both forward as well as backward directions. This enables better learning and handling of data dependent on long-range. It is shown that the bidirectional networks are better than the unidirectional LSTM. Figure 1 shows BiLSTM architecture. Here, \vec{h} is the forward layer based output sequence, which can be iteratively calculated by applying inputs in form of a positive series from time $T - n$ to time $T - 1$. On the other hand, the backward-layer output sequence, \overleftarrow{h} , is determined via the reversed inputs from time $T - n$ to time $T - 1$. By using Eq. (4), we can calculate the forward and backward-layer outputs. Finally, Y_T is the processed output generated by the BiLSTM layer, where each element is calculated by using the following equation:

$$y_t = \sigma(\vec{h}, \overleftarrow{h}) \quad (4)$$

where σ function integrates the two output sequences. Similar to the LSTM layer, the final output of a BiLSTM layer can be denoted through a vector, $Y_T = [y_{T-n}, \dots, y_{T-1}]$, in which the y_{T-1} represents the predicted value for the next-time iteration.

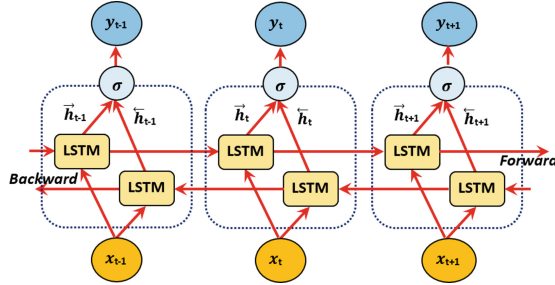


Fig. 1. Structure of BiLSTM

3.2 K-Means Clustering

K-means clustering is one of the popular unsupervised learning algorithms to identify groups of patterns in the form of clusters. Each feature set in the data is visualized as a point in the n -dimensional coordinate system. All the points closer to a particular cluster centroid belong to the same cluster and hence represent the same patterns in the time series. The algorithm is as follows:

1. Repeat until centroid stops moving
 - a. Select any k no. of points (cluster centroids) and name them as C_1, C_2 , etc.
 - b. Calculate distances of all the points from each centroid.
 - c. Assign each point to that cluster from which its distance is minimum.
 - d. For each cluster calculate new centroid using the below formula:-

$$c(i) = \text{sum}(p(i))/\text{count}(p(i))$$
 where i is the dimension
 p is a point and
 c is centroid

3.3 Proposed k Stacked BiLSTM Deep Learners

The existing works have shown that the deep LSTM architectures with several stacked LSTM hidden layers are more effective than the classical neural network model. In the deep LSTM model, the output of one hidden LSTM layer will be given as input for the successive hidden-layers. Such stacked layers model

can improve the computational power of neural networks. This paper proposes a k stacked BiLSTM deep learners, which uses the BiLSTMs to capture various useful information from spatial time series data and the LSTM layer as the top layer of the architecture which only utilize learned features from lower layers, to calculate and generate the predicted values. In Fig. 2, we have illustrated the architecture of the proposed deep learning model. This model is based on the assumption that the cluster data contains a group of patterns, and recognizing those groups and training against them, an individual deep learning model for each will result in better predictions.

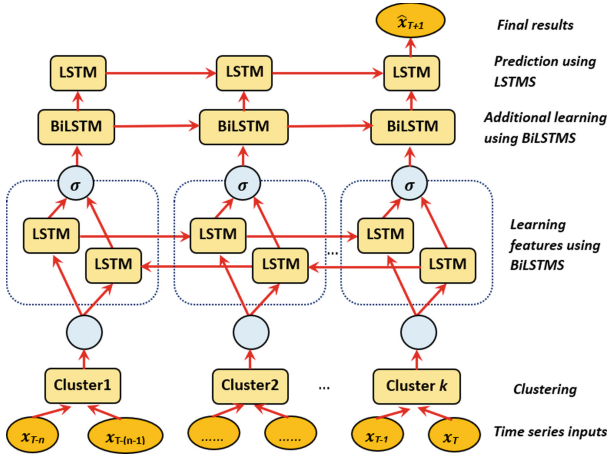


Fig. 2. Proposed architecture

The step-by-step explanation of the proposed model is shown in Algorithm 1. In our proposed model, we first divide our data into train and test sets so that we can calculate out-of-sample prediction. We classify the train set into k number of clusters using an unsupervised learning model i.e., K-Means. Here, choosing the right value of k is an essential factor for success. If the value of k is low, there might be some undetected patterns left in the data and scattered over different clusters. If k is very high, there might be redundancy in patterns, and also, a large number of models will have to be trained, which is computationally expensive. The value of k is chosen by the trial-and-error method. This model is saved and further used for classifying samples of the test set. Each of these clusters is trained separately, using stacked BiLSTM deep learners. So, there are k stacked BiLSTM deep learners for learning k clustered datasets. Each stacked BiLSTM contains a BiLSTM layer as the first feature-learning layer and, in top of that, an LSTM layer as the final layer. To make full use of input data, we apply additional learning using BiLSTM layers in the middle. The k BiLSTM deep learners are capable of predicting multiple steps of future data using the historical data. The detailed explanation of input data and the performance of k stacked BiLSTM deep learners is discussed in the experimental results section.

Algorithm 1: Proposed Algorithm

```

1 Input: No. of clusters  $n_c$ ; Multi-variate time series  $S_{in} \leftarrow \{T_1, T_2, T_3, \dots, T_n\}$ 
   where  $T_i \leftarrow \{x_1, x_2, x_3, \dots, x_{n_f}\}$  and  $n_f$  is no. of features; No. of iterations  $N$ ;
2 Output: New improvised prediction model  $P_{final}$ 
3 Divide the time series into two parts:
4 Training Set  $S_{train} \leftarrow \{T_i\}: i \in (0, l)$  and  $l = 0.8 * n$ ;
5 Test Set  $S_{test} \leftarrow \{T_j\}: j \in [l, n]$ ;
6 Let Set  $Y_{train}$  be a set of training outputs/targets
7 Let Set  $C \leftarrow \{C_1, C_2, C_3, \dots, C_{n_c}\}$  be a set of clusters where  $C_i \leftarrow \{T_i\}: T_i \in S_{train}$ , and  $C_i \cap C_j = \phi$  if  $i \neq j$ 
8 Let  $P_{final} \leftarrow \{P_1, P_2, P_3, \dots, P_{n_c}\}$ , where  $P_i$  is a weak model.
9 while ( $N \neq 0$ ):
10 begin
11   for  $X$  in  $S_{train}$ :
12     begin
13       if  $X \in C_i$ :
14          $Y_{h1} \leftarrow h1_i(W1, X)$ , where  $Y_{h1} = \{y_{11}, y_{12}, y_{13}, \dots, y_{1m}\} : m \neq n$ 
15          $Y_{h2} \leftarrow h2_i(Y_{h1})$ , where  $Y_{h2} = \{y_{21}, y_{22}, y_{23}, \dots, y_{2q}\} : q \neq m$ 
16          $Y_{h3} \leftarrow h3_i(Y_{h2})$ , where  $Y_{h3}$  is the target output.
17         Calculate difference between predicted and target value.
18          $\varepsilon = Y_{h3} - Y_{target, i}$ 
19         Adjust weights using  $\varepsilon$  as  $\Delta w = \alpha * \varepsilon * h3_i'(Y_{h2})$   $w_{new} = w_{old} + \Delta w$ 
20         Adjust intermediate weights for hidden layers using back-propagation
21        $N = N - 1$ 

```

4 Experiment Results Analysis and Comparison

In this study, we compare the prediction performance of the proposed model with different deep learning approaches for CPU and memory usage prediction of cloud workloads. We have used the dataset of the Google cluster trace. It contains the running information of a cluster having 12,500 machines and shows 29 days running time trace of various jobs dynamically entering, executing, and leaving the Google cluster at different time instants. In this study, we have used 11 different resource usage metrics for CPU and memory usage prediction. We have taken the sample size of 86,880 samples for training the deep learning models. Then we have generated multi-step forecasts for the next 90, 180, and 270 steps. The resource usage values of different jobs are aggregated at 5 min time period. As the classical time series models like Holt-winter, ARIMA are not worthy of prediction for upcoming resource usage. Thus, we apply deep learning models such as RNN, CNN, GRU, LSTM, Bi-LSTM, and Multi-layer for prediction of cloud resources. Table 1 presents the hyper-parameter settings of the proposed model. To analyze the performance of the multi-step ahead prediction, we have calculated the RMSE of out-of-sample prediction and compare the overall performance of the proposed model with the current deep learning

Table 1. Hyperparameter settings of the proposed model

Architecture details	
Parameter	Value
Default hidden layers	3
Units in hidden layers	60, 120, 60
Training step/epochs	70
Batch size	100
Loss function	Mean squared error
Optimizer	Adam
Activation function	tanh
Input layer neurons	12
BiLSTM layer neurons	60, 120, 60
Final layer neurons	1

models. Table 2 shows the RMSE performance of models for CPU and memory usage, respectively. As shown in the table, it can be identified that the proposed model generates the best predictions in comparison with other models. In Fig. 3 and 4, we have analyzed the iterative multi-step ahead resource utilization predictions of the model for (a) 90 (b) 180, and (c) 270 steps ahead. As shown in the figures, the proposed model accurately captures the trends of CPU and memory resource usage. Figure 5 shows the overall RMSE performance of all the models. The k stacked BiLSTM deep learners make use of the high coupling exist in both forward as well as backward time series dependencies and thus give superior prediction performance in comparison with the other models.

Table 2. Performance comparison of models for CPU and memory prediction

Model	Steps for CPU prediction				Steps for memory prediction			
	90	180	270	Overall	90	180	270	Overall
ARIMA	0.00667	0.00674	0.00640	0.00640	0.00488	0.00488	0.00485	0.00485
Linear regression	0.00565	0.00571	0.00573	0.00573	0.00424	0.00425	0.00425	0.00425
CNN	0.09004	0.09102	0.09139	0.09547	0.9121	0.91225	0.9123	0.91239
GRU	0.00541	0.00547	0.00549	0.00561	0.00263	0.00269	0.00274	0.00285
LSTM	0.00540	0.00546	0.00547	0.00559	0.00263	0.00267	0.00271	0.00283
BiLSTM	0.00542	0.00547	0.00549	0.00562	0.00266	0.00273	0.00277	0.00287
Proposed	0.00539	0.00544	0.00545	0.00555	0.00265	0.00266	0.00270	0.00281

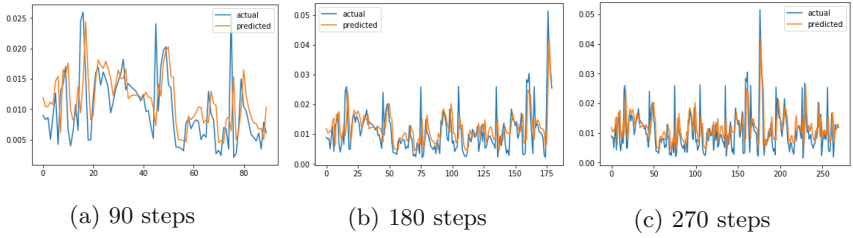


Fig. 3. Multi steps ahead CPU prediction using k stacked BiLSTM where X-axis represents *No. of Steps*, and Y-axis represents *CPU* values

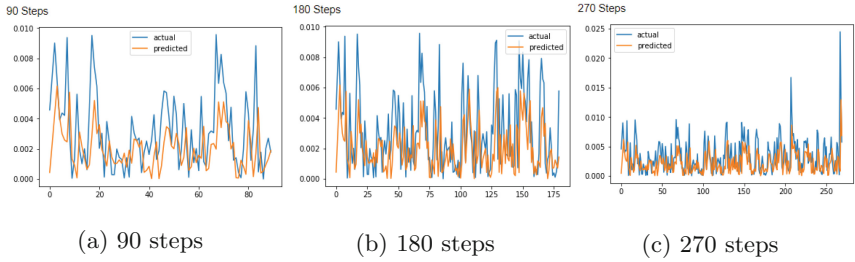


Fig. 4. Multi steps ahead memory prediction using k stacked BiLSTM where X-axis represents *No. of Steps*, and Y-axis represents *Memory* values

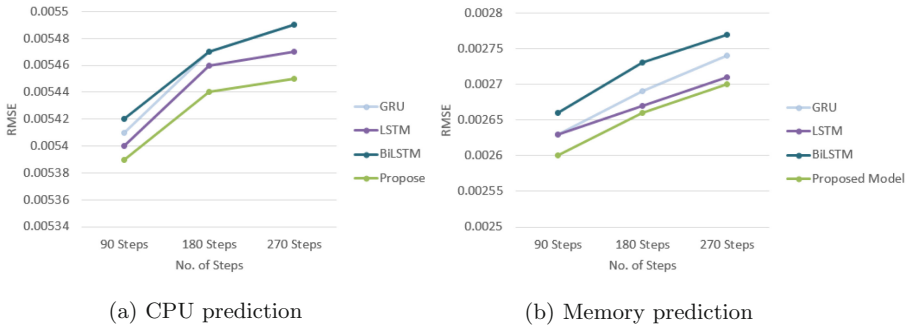


Fig. 5. Performance comparison of models

5 Conclusions

Resource usage prediction is a critical aspect for better load balancing, scheduling, and dynamic scaling of cloud resources. In this paper, we propose an architecture of k stacked BiLSTM deep learners considering both forward and backward dependencies of cloud resource usage. We have validated the proposed model on a publicly available Google cluster trace. Through experimental results, we have observed that the proposed network can model the long range dependencies present in time series data and also efficient to capture the spatial-temporal features from the multi-variate resource usage dataset.

Acknowledgements. The work is partially supported by Department of Science and Technology (DST), Government of India under ICPS Programme through the Project No.: DST/ICPS/CPS-Individual/2018/403(G), “Low-cost Energy-Efficient Cloud for Cyber-Physical Disaster Management Systems”. The first author, Yashwant Singh Patel, acknowledges Visvesvaraya Ph.D. Scheme for Electronics and IT under Ministry of Electronics and Information Technology (MeitY), Government of India for supporting this research.

References

1. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. *Internet Serv. Appl.* **1**(1), 7–18 (2010)
2. Kaur, T., Chana, I.: Energy efficiency techniques in cloud computing: a survey and taxonomy. *ACM Comput. Surv.* **48**(2), 22:1-22:46 (2015)
3. Zhang, Q., et al.: Dynamic energy-aware capacity provisioning for cloud computing environments. In: *International Conference on Autonomic Computing (ICAC)*, pp. 145–154. ACM, New York (2012)
4. Caglar, F., Gokhale, A.: iOverbook: intelligent resource overbooking to support soft real-time applications in the cloud. In: *IEEE 7th International Conference on Cloud Computing (CLOUD)*, pp. 538–545. IEEE, Anchorage (2014)
5. Gong, Z., Gu, X., Wilkes, J.: PRESS: PRedictive elastic ReSource scaling for cloud systems. In: *International Conference on Network and Service Management (CNSM)*, pp. 9–16. IEEE, Niagara Falls (2010)
6. Nguyen, H., Shen, Z., Gu, X., Subbiah, S., Wilkes, J.: AGILE: elastic distributed resource scaling for infrastructure-as-a-service. In: *10th International Conference on Autonomic Computing (ICAC)*, pp. 69–82. USENIX, San Jose (2013)
7. Ghorbani, M., Wang, Y., Xue, Y., Pedram, M., Bogdan, P.: Prediction and control of bursty cloud workloads: a fractal framework. In: *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp. 1–9 (2014)
8. Reiss, C., Wilkes, J., Hellerstein, J.L.: Google cluster-usage traces: format + schema, November 2011
9. Song, B., Yu, Y., Zhou, Y., et al.: Host load prediction with long short-term memory in cloud computing. *J. Supercomput.* **74**, 6554–6568 (2018)
10. Aboagye-Sarfo, P., et al.: A comparison of multivariate and univariate time series approaches to modelling and forecasting emergency department demand in Western Australia. *J. Biomed. Inform.* **57**, 62–73 (2015)

11. Gupta, S., Dinesh, D.A.: Resource usage prediction of cloud workloads using deep bidirectional long short term memory networks. In: International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–6. IEEE, Bhubaneswar (2017)
12. Wakuya, H., Shida, K.: Bi-directionalization of neural computing architecture for time series prediction. III. Application to laser intensity time record Data Set A. In: International Joint Conference on Neural Networks (IJCNN), pp. 2098–2103. IEEE, Washington (2001)
13. Zhao, Y., Ye, L., Li, Z., Song, X., Lang, Y., Su, J.: A novel bidirectional mechanism based on time series model for wind power forecasting. *Appl. Energy* **177**, 793–803 (2016)



Offline Bengali Handwritten Sentence Recognition Using BiLSTM and CTC Networks

M. A. Muhaimin Sakib, Omar Sharif, and Mohammed Moshiul Hoque^(✉)

Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh
{muhaimin.sakib,omar.sharif,moshiul_240}@cuet.ac.bd

Abstract. It is a very challenging task to recognize unconstrained Bengali handwritten text due to its cursive nature. This paper introduces an offline technique of recognizing handwritten Bengali sentences based on BiLSTM architecture and connectionist temporal classification (CTC) output layer. The traditional approach of detecting handwritten sentence recognition rises the computational complexity due to the difficulty of combining isolated characters. The proposed method only segments the sentences into words, recognizes separately and combines them to represents the output sentence. The proposed technique is evaluated with 450 Bengali sentences and achieved 92% accuracy for recognizing sentences considering 15 distinct writing styles.

Keywords: Pattern recognition · Handwritten recognition · Word separation · Feature extraction · Bidirectional long short term memory

1 Introduction

Recognizing handwriting is the specialized research issue in the scope of pattern recognition which recognizes handwritten texts by analyzing various patterns observed in hand written scripts. This is the aptitude of computers to understand and translate handwritten text into electronic form or data. The advantage of transforming physical written documents into electronic form is that digital documents are more comfortable to store, retrieve and manipulate. Besides this, it is easy to store, edit and access digital document compared to a handwritten text document. Moreover, physical documents take a tremendous amount of office space, and much valuable knowledge gets lost because documents never get translated into digital format. Although significant advancements have been made in recognizing handwritten in English and other languages, it is in the initial stage in Bengali language [8]. The process of handwritten identification can be segregated into two schemes: online and offline. In the online approach, the motion of pen tip is recorded and using those coordinates the handwriting is recognized [5]. Only an image containing the handwritten text provides the

necessary information and the offline technique have to identify from the image [1]. This paper concerned on the offline approach.

A significant amount of work has conducted to recognize Bengali handwritten digits, vowels, and consonants using fuzzy logic and neural network. This research proposes a deep neural network architecture due to its large number of features consideration and better accuracy than the general neural network architectures [16]. The task of recognizing Bengali handwriting is quite challenging due to its various writing styles, shortage of resources, presence of uncorrelated segments and broken characters. Furthermore, characters in a word connected by a line called “Matra” in Bengali handwriting which makes them more challenging to segment the words into isolated characters. To the best of our knowledge, no significant advances have been made so far in recognizing handwritten Bengali sentences. The suggested technique present in this paper can recognize Bengali handwritten sentences by segmenting it into constituent words and then recognize the individual words without segmenting them into constituent characters.

2 Related Work

There are no significant research conducted yet to recognize the Bengali handwritten sentences several approaches are already available that deals with the recognition of handwritten Bengali digits, characters and words respectively [1, 15, 17].

There are several methods for recognizing handwritten characters. Some methods are used fuzzy logic [6, 10] while others use neural networks [3, 7]. There are also methods using genetic algorithms described in [9, 11]. Alom et al. [2, 3] used different deep learning techniques to recognize Bengali handwritten digits and characters. Digit recognition system gets maximum 98.78% accuracy by using CNN with dropout and Gabor filters. Deep convolution neural networks can recognize characters with approximately 98% accuracy. The system proposed by Adak et al. [1] is achieved 86.96% accuracy in detecting handwritten cursive Bangla words. Performance of this system can be enhanced by employing the recurrent neural network. A new method can identify handwritten Bengali words using a set of fuzzy linguistic rules [6]. The online handwritten recognition system has proposed by Bhattacharya et al. [5], which used HMM-based classifier and N-pen++ features. This method achieved 92.33% accuracy with 20000 words. Recent work focused on LSTM and CNN models to classify handwritten English words [4]. This work used CNN model to classify words and LSTM model to segment the characters. An automated system that detects English sentence by analyzing gestures of the mouse in real-time based on ANN with notable accuracy shown in [13]. Graves et al. [7] presented an alternative method for sequence labelling tasks where data have long-range bidirectional interdependencies and hard to segment based on novel type RNN technique. Majority of the existing work focused on recognition of Bengali digits, characters or words. None of the previous work focused to recognise Bengali handwritten sentences. This work proposes a machine learning-based framework that can recognize the handwritten Bengali sentences.

3 Proposed Handwriting Recognizer

Figure 1 represents the proposed framework of Bengali handwritten sentence recognition. The framework consists of two main parts: learning and recognition. Both of them consists of several smaller parts which explained below.

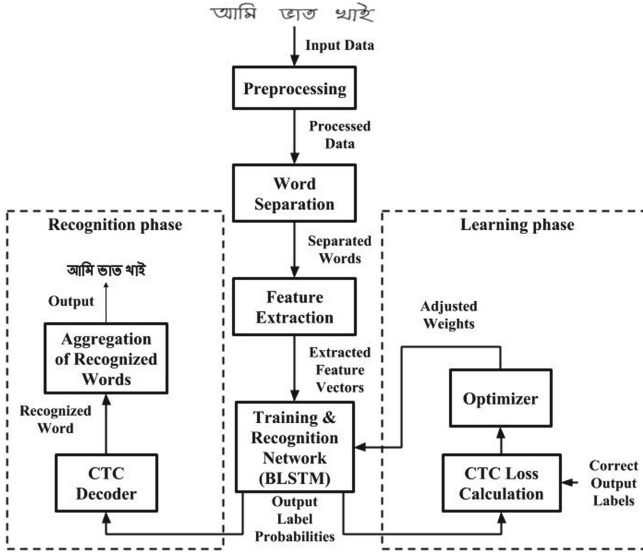


Fig. 1. Proposed framework for recognizing Bengali handwritten sentences

3.1 Preprocessing

RGB image of the handwritten sentence takes as the input of the suggested technique. The input image is further processed to extract relevant features by applying following steps. **Step 1:** the input RGB image I_{rgb} is translated to gray-scale image I_{gray} [14]. **Step 2:** the gray-scale image I_{gray} is transformed to binary image I_{binary} using Otsu's method [12]. **Step 3:** the binary image I_{binary} is cropped so that there are no redundant pixels at the beginning, end, top or bottom of the image. These steps are summarized in Fig. 2.

3.2 Word Separation

The column wise density of black pixel (d_{col}) is considered to separate words in an image which is done in following two steps.

- Step 1: a sliding window of height ($H = \text{image height}$), width ($W = 1 \text{ pixel}$) is taken. This window is slid over the image (I) to right for calculating black pixel counts (C_{black}) in each window position. Pseudo-code for this step is shown in Algorithm 1.

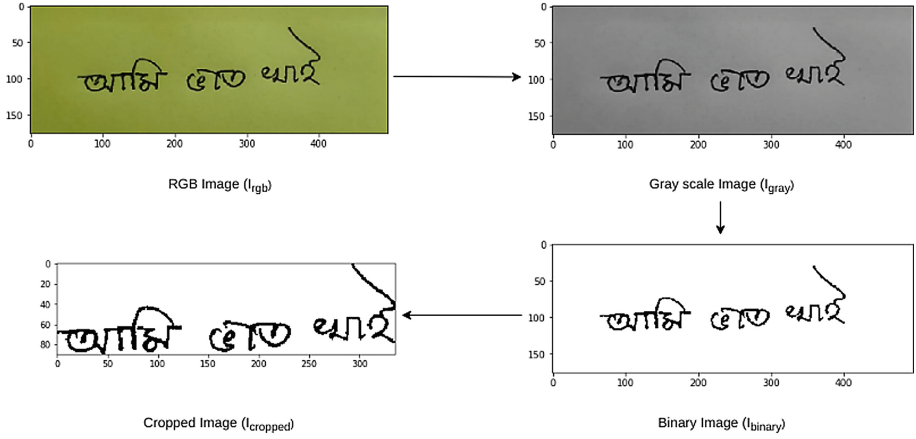


Fig. 2. Output images after preprocessing the sentence ‘ আমি ভাত খাই ’

Algorithm 1. Count the number of black pixels in a column

- 1: Initialize variables, $i \leftarrow 0, L \leftarrow image_width$
 - 2: Calculate C_{black} in i^{th} column
 - 3: **if** $i = L - 1$ **then**
 - 4: *exit*
 - 5: **else**
 - 6: $i \leftarrow i + 1$
 - 7: *go to step 2*
 - 8: **end if**
-

- Step 2: count the number of consecutive window positions C_{window} where $C_{black} = 0$. The image is segmented if the following equation holds,

$$C_{window} \geq T_{max} \quad (1)$$

Here T_{max} is the maximum value C_{window} can have before we segment the image. The success of these algorithms depends on the following two assumptions,

$$D_{min} \geq T_{max}, D_{max} < T_{max} \quad (2)$$

Here, D_{min} is the minimum distance between consecutive words and D_{max} is the maximum distance between consecutive characters. Algorithm 2 shows the pseudo-code for performing segmentation and the word separation process is depicted in Fig. 3.

3.3 Feature Extraction

The feature vectors enumerated for each word image W_{img} . Nine features are calculated in each window position by using a sliding window along the width of

Algorithm 2. Word Segmentation

```

1: Initialize variables,  $i \leftarrow 0, C_{window} \leftarrow 0$ 
2:  $C_{black} \leftarrow$  no. of black pixels at  $i^{th}$  column
3: if  $C_{black} = 0$  then
4:    $C_{window} \geq C_{window} + 1$ 
5: else
6:    $C_{window} \leftarrow 0$ 
7: end if
8: if  $C_{window} = T_{max}$  then
9:   segment the image
10: else
11:    $i \leftarrow i + 1$ 
12: end if
13: if  $i = image\_width$  then
14:   exit
15: else
16:   go to step 2
17: end if

```

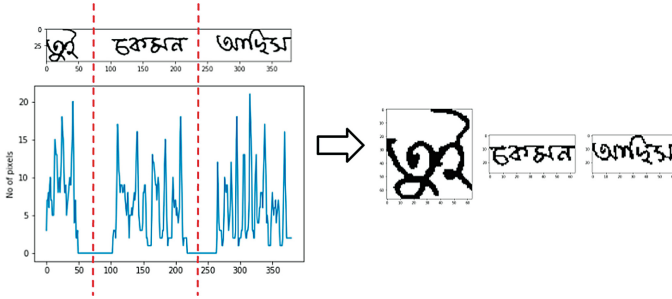


Fig. 3. Segmentation of S_{img} (left image) into W_{img} (right image).

the image. The mean number of black pixels at each window position, the centre of gravity of the pixels, the second-order vertical moment of the centre of gravity and the positions of the lowermost and uppermost black pixels are calculated by using Eqs. 3–6 respectively.

$$f_1(t) = \frac{1}{m} \sum_{y=1}^m p(t, y) \quad (3)$$

$$f_2(t) = \frac{1}{m} \sum_{y=1}^m y \cdot p(t, y) \quad (4)$$

$$f_3(t) = \frac{1}{m^2} \sum_{y=1}^m y^2 \cdot p(t, y) \quad (5)$$

$$f_4(t) = y_u, f_5(t) = y_l \quad (6)$$

Equations 7–9 calculates gradient of uppermost and lowermost black pixels, number of black-white transitions and number of black pixels inside the window respectively.

$$f_6(t) = \frac{f_4(t)}{t}, f_7(t) = \frac{f_5(t)}{t} \tag{7}$$

$$f_8(t) = \sum_{y=1}^m (p(t, y) \oplus p(t, y - 1)) \tag{8}$$

$$f_9(t) = \sum_{y=1}^m p(t, y) \tag{9}$$

Figure 4 illustrates the visual representation of features.

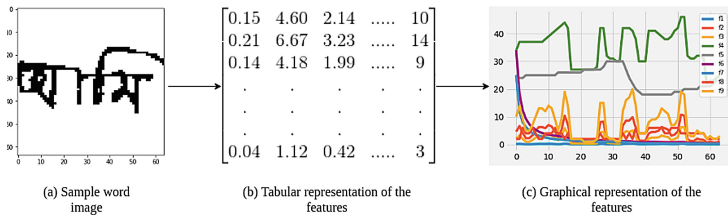


Fig. 4. Extracted feature vectors form input image of word ‘আমি’.

3.4 Training and Recognition Network

The input to the BiLSTM network is a (64 × 9) matrix. This layer produces a probability distribution for each of the 59(+blank) character present in Bengali alphabet. Therefore, the output of this network is a 64 × 60 matrix which is fed to the CTC layer. Figure 5 illustrates the input and corresponding output of the BiLSTM network.

Learning Phase. The system learns using the output of the BiLSTM network and weights of the network are adjusted to enhance the recognition performance. This phase consist of two basic parts: CTC loss and optimizer.

- **CTC loss Calculation:** CTC loss function calculate the loss using the output matrix of the BiLSTM network and the correct output labels. The loss value is calculated as the negative *log* of the sum of probability of all paths that correspond to the correct sequence of labels.

$$loss = -\ln \sum_{i=1}^x p(q_i) \tag{10}$$

Loss value is calculated by Eq. 10 where $q_1, q_2, q_3, \dots, q_x$ are all the paths that correspond to correct sequence of labels.

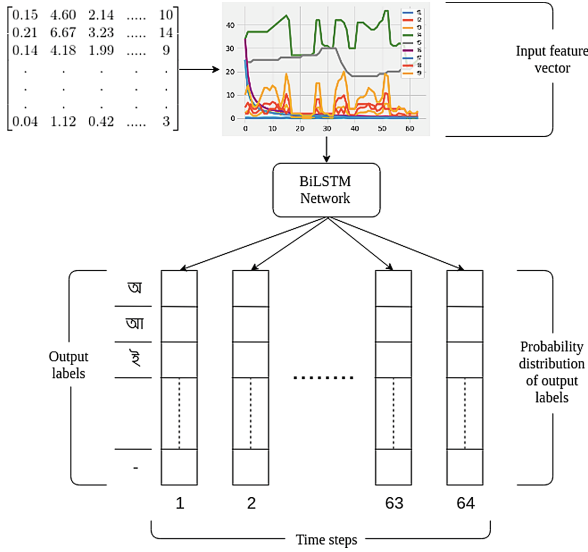


Fig. 5. Input and output of BiLSTM network.

- **Optimizer:** The optimizer try to minimize loss as well as adjust the weights of the BiLSTM network by considering the calculated CTC loss. The adjusted weights of the network is enables it to perform better.

Recognition Phase. In preprocessing step, a sentence image S_{img} is converted into several word images w_1, w_2, \dots, w_l . The proposed system is recognize these word images w_i to produce final output which is the recognised sentence. The recognition is done in two steps: CTC decoding and aggregation.

- **CTC Decoder:** An unknown input sequence u represents the probability distribution of output labels, produced by the BiLSTM network. This is translated by CTC decoder which choose the sequence of labels L^* with the maximum conditional probability calculated by Eq. 11 in BiLSTM network.

$$L^* = \max_L p(L|u) \tag{11}$$

This sequence of labels represented by L^* represents our recognised word.

- **Aggregation of Words:** the recognized words $L_{w1}^*, L_{w2}^*, \dots, L_{wl}^*$ are concatenated using a space character (G) in between them to form final output sentence F_s of length S_l . Where S_l represents the number of words the sentence consists of. Using Eq. 12 we get our final final sentence by aggregating words.

$$F_s = \begin{cases} \sum L_{wi}^* + G & \text{if } i < S_l \\ \sum L_{wi}^* & \text{if } i = S_l \end{cases} \tag{12}$$

4 Evaluation Measures

The proposed system is evaluated at word level as well as sentence level in terms of edit distance, segmentation error rate, label error rate, word accuracy, and sentence accuracy respectively.

- **Edit Distance:** It is the deviation of output sequence ($w_1w_2..w_n$) from the actual sequence of labels ($g_1g_2..g_n$). It can be expressed as in Eq. 13.

$$E_d = \gamma(w_1w_2..w_n, g_1g_2..g_n) \quad (13)$$

This returns the minimum number of substitution, insertion or deletion required to convert $w_1w_2..w_n$ to $g_1g_2..g_n$.

- **Segmentation Error Rate:** It denotes the percentage of sentences that the system failed to correctly segment. If total number of sentence is T_s and number of correctly segmented sentence is C_s then segmentation error S_e is given by Eq. 14.

$$S_e = \frac{C_s}{T_s} \quad (14)$$

- **Label Error Rate:** It represents the percentage of characters that the system failed to correctly identify in a sentence. If R is a sample set of recognized words and X is the set of actual words then label error rate (L_e) is found by Eq. 15.

$$L_e = \sum_{x \in R, y \in X} \frac{\gamma(x, y)}{\text{len}(y)} \quad (15)$$

Here, $\text{len}(y)$ gives the number of characters in word y .

- **Word Accuracy:** It gives the percentage of words that the system able to correctly predict. A word w is considered correctly predicted if $E_d = 0$.
- **Sentence Accuracy:** It represents the percentage of correctly predicted sentences. Sentence accuracy S_a is calculated by the ratio of correctly predicted sentence C_p and total number of sentences T_s .

5 Experimental Results

Due to scarcity of resources and benchmark dataset on Bengali handwritten we developed a dataset. The summary of the dataset shown in the Table 1.

The proposed system trained the classifier on the word level. The BiLSTM-CTC model used in this experimentation consists of three layers, each with 64 BiLSTM cells. Each BiLSTM cell consists of 512 hidden nodes. The network is trained for 4000 iterations on the training set before evaluation. We used RMSProp optimizer to adjust the weights of our network, and for decoding the, we used CTC beam search decoding algorithm with a beamwidth of 80. Table 1 also indicates that the overall accuracy of the proposed technique at the sentence level. The result indicates that the system can predict the Bengali handwritten sentences with 92% accuracy.

Table 1. Dataset statistics

	Training	Testing	Accuracy
Number of sentences	300	150	92%
Number of words	930	465	
Avg. no. of words per sentence	3.7	3.5	
Writing styles	15	15	
Age range	19–24	19–24	

The system also evaluated by considering different sentence lengths and different writing styles to see how accuracy varies depending on various contexts. The performance of the system for four different sentence lengths shown in Fig. 6. This result indicates that as the length of the sentence increases, segmentation error also increases and accuracy decreases as expected.

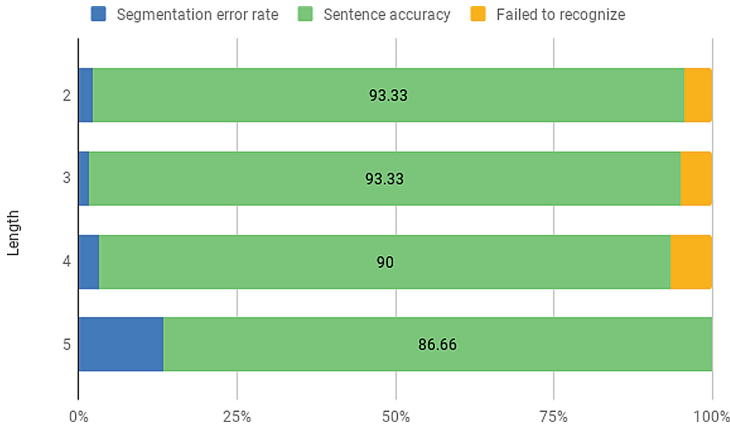


Fig. 6. Accuracy versus sentence length.

Table 2 shows samples of 15 different writing styles and their corresponding accuracy. We can see that the system performs well for most of the writing styles. For the styles 10 and 14, the system recognized only 70% and 50% of the sentences respectively.

A comparative analysis is performed between the proposed technique and CNN. The CNN architecture consisted of 3 convolution, 2 max pooling, 2 fully connected and a dropout layers. Table 3 displays the analysis of comparison. The result shows that BiLSTM-CTC networks are more accurate to predict the Bengali handwritten sentences than CNN. In BiLSTM-CTC, the output nodes and labels are in equal numbers for the Bengali language. On the other hand, the CNN output layer considers as many nodes as there are words in Bengali vocabulary which can be quite large.

Table 2. Performance of the system for various writing styles.

Style	Sample images	Recognized sentence	Word accuracy	Sentence accuracy
1	আমি ভাত খাই	আমি ভাত খাই	100%	100%
2	খাটি তোমাকে ভালবাসি	খাটি তোমাকে ভালবাসি	96.66%	90%
3	বিশের নিচ্ছি	বিশের নিচ্ছি	91.66%	80%
4	পুলিশকে ডাক	পুলিশকে ডাক	100%	100%
5	একটি ভাষা যথেষ্ট নয়	একটি ভাষা যথেষ্ট নয়	100%	100%
6	সে কোন দেশের লোক	সে কোন দেশের লোক	100%	100%
7	আপনি কি বাংলা বলতে পারেন	আপনি কি বাংলা বলতে পারেন	100%	100%
8	মাফ করবেন	মাফ করবেন	100%	100%
9	আমার নাম রহিম	আমার নাম রহিম	100%	90%
10	তুমি কেমন আছো	তুমি কেমন আছো	100%	70%
11	পুলিশকে ডাক	পুলিশকে ডাক	100%	100%
12	বিদায় নিচ্ছি	বিদায় নিচ্ছি	100%	100%
13	মাফ করবেন	মাফ করবেন	100%	100%
14	সে কেমন তোন লোক	সে কেমন তোন লোক	79.63%	50%
15	আমি ভাত খাই	আমি ভাত খাই	100%	100%

Table 3. Comparative analysis between BiLSTM-CTC and CNN at word level.

Architecture	Total words	Correctly predicted	Word accuracy
BiLSTM-CTC	465	378	81.29%
CNN	465	351	75.48%

6 Conclusion

This work presents a deep neural network-based technique that can recognize the Bengali handwritten sentences. The evaluation results indicated that BiLSTM-CTC network is more effective in recognizing Bengali handwritten sentences. This architecture enables the system to recognize words without segmenting them into constituent characters. The overall accuracy of the system can be enhanced by using a different set of features and increase the number of handwriting styles in the training dataset. This present implementation can be extended to recognize handwritten paragraphs by including more features in preprocessing and segmentation phases.

References

1. Adak, C., Chaudhuri, B.B., Blumenstein, M.: Offline cursive Bengali word recognition using CNNs with a recurrent model. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 429–434. IEEE (2016)
2. Alom, M.Z., Sidike, P., Hasan, M., Taha, T.M., Asari, V.K.: Handwritten bangla character recognition using the state-of-the-art deep convolutional neural networks. *Comput. Intell. Neurosci.* **2018** (2018)
3. Alom, M.Z., Sidike, P., Taha, T.M., Asari, V.K.: Handwritten Bangla digit recognition using deep learning. arXiv preprint [arXiv:1705.02680](https://arxiv.org/abs/1705.02680) (2017)
4. Balci, B., Saadati, D., Shiferaw, D.: Handwritten text recognition using deep learning. CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring (2017)
5. Bhattacharya, N., Roy, P.P., Pal, U., Setua, S.K.: Online Bangla handwritten word recognition. *Malays. J. Comput. Sci.* **31**(4), 300–310 (2018)
6. Chowdhury, K., Alam, L., Sarmin, S., Arefin, S., Hoque, M.M.: A fuzzy features based online handwritten Bangla word recognition framework. In: 2015 18th International Conference on Computer and Information Technology (ICCIT), pp. 484–489. IEEE (2015)
7. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(5), 855–868 (2009)
8. Hussain, R., Raza, A., Siddiqi, I., Khurshid, K., Djeddi, C.: A comprehensive survey of handwritten document benchmarks: structure, usage and evaluation. *EURASIP J. Image Video Process.* **2015**(1), 46 (2015)
9. Kala, R., Vazirani, H., Shukla, A., Tiwari, R.: Offline handwriting recognition using genetic algorithm. arXiv preprint [arXiv:1004.3257](https://arxiv.org/abs/1004.3257) (2010)
10. Malaviya, A., Peters, L.: Extracting meaningful handwriting features with fuzzy aggregation method. In: 1995 Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 2, pp. 841–844. IEEE (1995)
11. Oliveira, L.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In: Null, p. 10568. IEEE (2002)
12. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
13. Parwej, F.: English sentence recognition using artificial neural network through mouse-based gestures. arXiv preprint [arXiv:1301.4659](https://arxiv.org/abs/1301.4659) (2013)
14. Saravanan, C.: Color image to grayscale image conversion. In: 2010 Second International Conference on Computer Engineering and Applications, vol.2, pp. 196–199. IEEE (2010)
15. Sazal, M.M.R., Biswas, S.K., Amin, M.F., Murase, K.: Bangla handwritten character recognition using deep belief network. In: 2013 International Conference on Electrical Information and Communication Technology (EICT), pp. 1–5. IEEE (2014)
16. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015)
17. Shawon, A., Rahman, M.J.U., Mahmud, F., Zaman, M.A.: Bangla handwritten digit recognition using deep CNN for large and unbiased dataset. In: 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–6. IEEE (2018)



Robotics Vehicle System Using Simple Hand Gesture

Sanjay Misra¹(✉), Modupe Odusami¹, Olusola Abayomi-Alli¹, Olaoluwa Oseni¹, Robertas Damasevicius², and Ravin Ahuja³

¹ Department of Electrical and Information Engineering, Covenant University, Ota, Nigeria
{Sanjy.misra, modupe.odusami, olusola.abayomi-alli}@covenantuniversity.edu.ng, olaoluwa.oseni@stu.cu.edu.ng

² Kaunas University of Technology, Kaunas, Lithuania

³ ShriVishwakarma Skill University, Gurgaon, India

Abstract. The development of robots has enabled humans to allow them to carry out simple tasks and activities for easier and faster living. Robotic vehicles are found in our houses and industries and would continue to be integrated into more aspects of human living. In the present age of technology, most robotic systems found in industries and our everyday lives are controlled by a traditional input method or are pre-programmed. This makes them a little tedious for operation by a common man without technical know-how or a certain level of expertise. Due to this, there is a need for a natural mode of communication between humans and robots. This study develops a robotic vehicle system that can be controlled by a user's simple hand gestures. The system employs the use of an accelerometer to detect hand gestures and then processed by Arduino boards to relay instructions to the robotic vehicle. The development of a hand gesture control system for a robotic vehicle provides a more natural mode of communication for human-robot interaction. This technology would increase the applications of robotic vehicles in various sectors through its implementation.

Keywords: Control system · Hand gesture · Robotics · Robot interaction

1 Introduction

Robotics is a growing field in engineering. It is concerned with the design, construction, and control of robotic devices. The word robot comes from a Slavic word *robota*, which means labor or to work. A robot is a machine, usually programmable by a computer that is capable of carrying out a series of complex tasks and activities simultaneously [1, 2]. Technological advancements over the years have introduced robotic devices that handle day-to-day operations for human beings. Domestic or service robots can be found in houses in the form of vacuum cleaners, automatic pool cleaners, window washers, and many more. Robots also find applications in various sectors of the economy handling operations that were previously handled by humans. In the automobile sector, robots handle a large portion of the assembly process of cars and trucks. Construction robots

help in lifting heavy objects in building sites, and automated tractors help to till the land and plant crops in what is known as mechanized agriculture [3].

These robotic devices are interacted with using quite a several interfaces because they still need to be controlled by humans. This control can be either through wired or wireless connections. Robots are used in a variety of fields and require a control mechanism, also known as human-robot interaction that feels natural and easy to the human operator. This control mechanism has to be able to recognize the instructions from the operator, interpret the instructions, and deliver to the robot to get the desired result [4]. Currently, the control mechanisms employed to control robots include a traditional remote control where a button is pressed, and the robot performs a function. Some other control systems require a high level of expertise because a computer program has to be written to get the desired output from the robot. Therefore, the human operator has to be trained to use the robotic device especially those applied in the industrial and construction sectors.

This study is motivated by the increasing usage of robots and robotic vehicles for performing more tasks for faster and easier operations. Also, traditional methods of control of robotic vehicles such as a joystick, remote control are unnatural modes of communication that need the training to use. Hand gestures provide a natural mode of communication because people generally use their hands to communicate ideas visually. This can be exploited to provide an easier way for control of robots and also opens up the possibilities of more applications of robots in our activities. This study aims at providing a natural mode of communication because people generally use their hands to communicate ideas visually. This can be exploited to provide an easier way for control of robots and also opens up the possibilities of more applications of robots in our activities. In this study, we developed a control system for a robotic vehicle based on human hand gestures by focusing on the need for a natural mode of communication between the human operator and the robot being controlled. The main contribution of this study is the provision of the system architecture of the hand gesture control system for a robotic vehicle whereby Radio Frequency transmission was used to transfer data between the transmitter and receiver unit. The rest of the paper is sectioned as follows: Sect. 2 presents the background of the proposed study. Section 3 discusses the related works including state of the art methodologies for formalizing gesture control models. Next, Sect. 4 presents the hardware and software applied in the design and implementation of the proposed model. Section 4 presents results analysis and discussion. Section 5 concludes and suggests future areas of research.

2 Background

This section discusses the development and importance of robotic vehicles, gives an overview of human-robot interaction and current technologies being applied for robot control, and introduces gesture control as a method of human-robot interaction and how it can be implemented on a control system.

2.1 Importance of Robotics

A robotic vehicle can be described as a vehicle that moves autonomously on the ground, in the air, under the sea, or in space [5]. These robots move without direct human operation and have sensors and resources that allow them to move and perform tasks on their power. Autonomous mobile robots have the ability to moving and not be confined to a particular location. Despite their independent mobility, in applications where mobile robots are utilized, the ability to control or direct a mobile robot by a trained human operator is sometimes useful [6]. The robotic vehicle is usually linked with the operator control system through cable or wireless media in the case of higher bandwidth communications. Robotic vehicles are also referred to as mobile robots. Mobile robots in their physical and mechanical construction can be legged, wheeled, or specially tracked for navigating specific terrains. One importance of robotic vehicles is that they can go to environments where humans cannot due to inhabitable circumstances. Robotic vehicles are very important in the space sector for interplanetary exploration. The Mars Rover is a popular robotic vehicle and the latest version was launched by NASA (National Aeronautics and Space Administration) in 2011.

2.2 Human-Robot Interaction

Human-robot interaction (HRI) is a research field dedicated to the study, understanding, and evaluation of machines and robots when they are working with or for humans [7]. It examines the mode of control and communication of instructions and information from the human operator to the machine or robot to achieve a specific task. Interaction indicates communication of instructions and feedback between the human and the robot. Interaction can either be remote that is, the human and robot are in two different locations or proximate which means that the human and robot are in close positions to each other. HRI involves discovering the capabilities of humans and robots and designing and implementing technologies that allow desirable interaction between them [8]. A good human-robot interaction system should be able to receive the input from the human operator, transform them into instructions, and relay to the robot to perform the given task. Traditional methods of robotic vehicle control include remote control systems: a button is pressed and the robot moves in the pre-programmed direction and joystick control where the robotic vehicle moves in the direction moved by the joystick.

General modes of communication and HRI are considered unnatural because they require training and a certain level of expertise to use and operate. This project aims to show that natural modes of interaction can be incorporated into a robotic vehicle with wheels. The natural interface is an interface that allows a human operator to instruct a machine or robot in a simple way that can be used by non-skilled operators with little training. The interface suggested in this study is hand gesture control. Hand gestures of the human operator are recorded by a unit and programmed to the robot parameters to enable control.

2.3 Gesture Control

Gesture control is the ability to recognize and interpret the movements of the human body to interact with and control a computer or robotic system without direct physical

contact [9]. The interface of these systems is known as the natural user interface (NUI). The human language would be more understandable by computers, robots, and machines and creates a more immersive user experience for the operators. The development of the natural interface of control using gestures would enable non-expert users to operate robotic vehicles for different services. Gesture control is most important in situations where the machine or the computer has to be operated remotely and without physical contact. This technology makes that possible. Gestures can be defined as the movement of the body and limbs that expresses an idea or sentiment [10].

Gesture recognition and control can be achieved through different technologies, the most common being vision-based systems. Vision-based systems for gesture control employ the use of a camera to detect hand movements of the human operator and interpret them as instructions for the robot. A virtual 3D model of the human hand is constructed to calibrate the system with acceptable gestures, and then the camera tries to match the hand movements of the user with previously recorded gestures from the model [11]. The vision-based systems have also been used in the area of sign language recognition, where a camera records hand motions and matches them with pre-recorded sign language words to convert them into speech or text [12]. This research would be using an accelerometer sensor system due to some shortcomings detected in vision-based systems. The accelerometer would detect movements on the human operator's hand and move the robotic vehicle according to the interpreted instruction.

3 Related Works

Different researchers have carried out studies and projects on how gesture recognition and control can be incorporated into systems for better human interaction. Authors in [13] proposed a system that implements a motion sensor that would enable a drone to be controlled by a user's simple hand gestures. A LEAP motion sensor is a computer hardware sensor that recognizes hand and finger movements as input. It was used in this project at the control section and it was used to operate a Parrot AR Drone 2.0. Python programming language was used to interact between the LEAP sensor and the drone. The program allows the sensor to be able to recognize different hand motions and gestures made by the users and these are then interpreted to the drone. A gesture is made by the human operator which is recorded and recognized by the motion sensor. A platform acts as an interface that interprets and relays instructions from the motion sensor to the robot. In most systems, it consists of a microcontroller and a communication system between the input and output systems.

Nayak et al. [14] proposed a glove-based system for hand gesture control based on a simple accelerometer to sense hand gestures of the human operators. The study implemented an ADXL345 accelerometer sensor and Arduino mega on the hand glove that served as the transmitter system, a ZigBee communication module for input and output connection, and a motor driver IC on the receiver system to interact with the robot according to values received from the transmitter. The accelerometer sensor records tilting movements of the user's hands and the Arduino reads the analog values provided and relays to the receiver system through the ZigBee module. The result shows that the Arduino microcontroller can be used as a platform for reading values from the

accelerometer sensor, interpreting them, and relaying the instructions to the robotic vehicle. The main challenge of this particular project was that the components used in the transmitter section of the system were quite large to be fitted in a simple hand glove.

Hand gesture control has continued to be studied as a method of human-computer interaction, and more research has been carried out in recent years on how to implement this on wheelchair control systems for disabled people. Authors in [15] developed a prototype for a wheelchair based on hand gesture control. The prototype operated with a transmitter system on the hands of the user and a receiver system on the wheels of the wheelchair. They implemented an ADXL335 accelerometer sensor and an Arduino ATMEGA microcontroller on the glove that would be the transmitter section of the prototype. Authors in [16] developed a hand gesture control system for a robotic arm and used a ZigBee module for communication between the input and output systems. The ZigBee module has a transmission rate of 256 kb/sec over a length of fewer than 10 m. This restricts the range at which the machine can be operated by the human operator. Gesture recognition for human-machine interaction can also be achieved with Image Processing techniques [17] The paper made use of an Intel RealSense 3D sensor to track hand movements through a camera. The sensor recognizes the hand movements of the user and translates it to instructions that are relayed to the robotic vehicle. Hand gesture recognition through image processing and hand tracking technology hasn't been fully developed, and factors such as lighting and environmental conditions can affect the input to the system. Authors in [18] employed the use of different image processing techniques for hand gesture recognition and interpretation and were able to control a robotic arm using gesture recognition technology. There are several other related works [19–22] available in literature but we are not including in detail those not related to hand gesture.

4 System Design and Implementation

The proposed system consists of two units namely the transmitter unit and the receiver unit as depicted in Fig. 1.

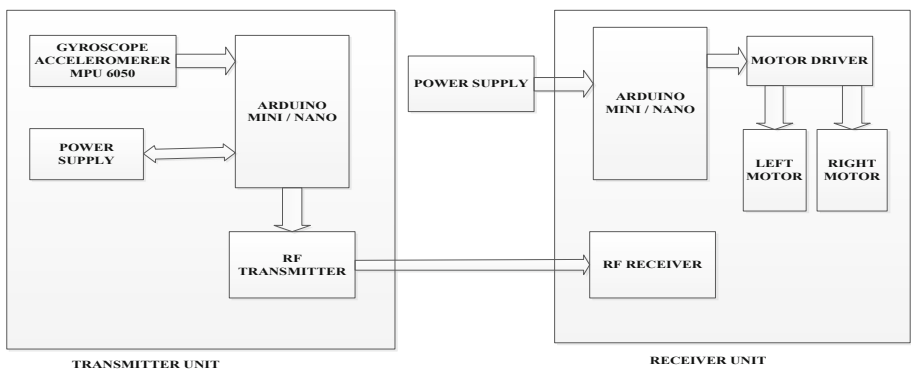


Fig. 1. Block diagram of proposed system

The transmitter unit consists of an MPU 6050 gyroscope, the Arduino Nano microcontroller, and the Radio Frequency (RF) transmission module being used for communication. The circuit diagram for the transmitter unit is depicted in Fig. 2.

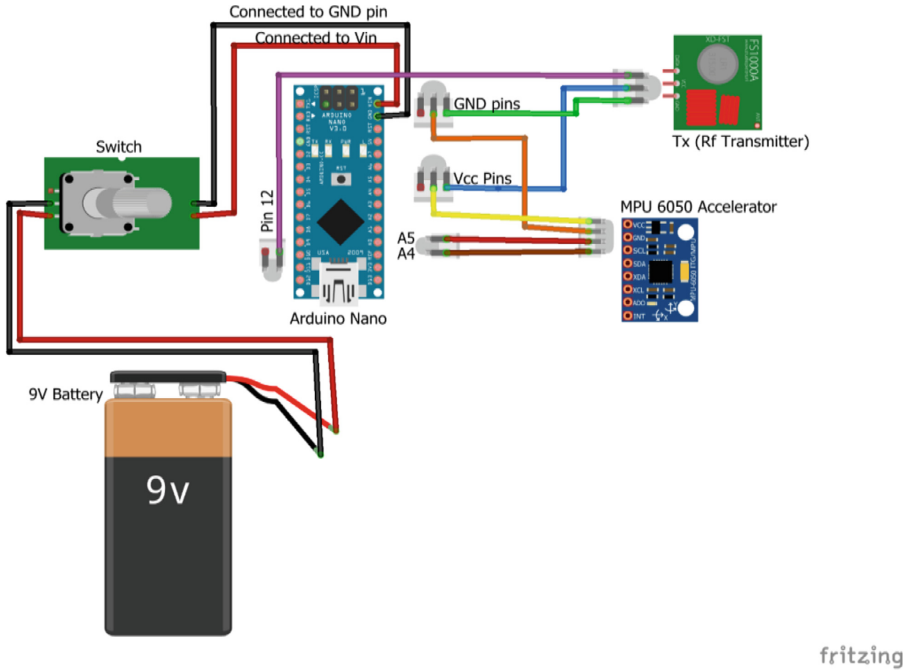


Fig. 2. Transmitter unit circuit diagram

From Fig. 2, MPU6050 is a micro electro mechanical system component that consists of a 3-axis accelerometer and gyroscope embedded in it. The component is used to measure and record changes in position, displacement, velocity, and orientation of a system or object. It measures its coordinates by recording X, Y, and Z-axis values. Movement can be detected and recorded when these values change over time. The MPU 6050 is interfaced with the Arduino Nano so that it can send these recorded values to it and the microcontroller stores certain values as instruction sets for the receiver module. The Arduino Nano is programmed using Arduino IDE software and the language is based on the C higher-level programming language. The Arduino is interfaced with the MPU 6050 using an accelerometer library and the RF module using a Virtual. wireh library that enables radio frequency wireless transmission. The Arduino receives the movement readings from the MPU 6050 through its data pins and defines instruction sets based on the readings and then sends these instructions to the RF module for transmission. The specification for Arduino Nano is shown in Table 1.

Table 1. Arduino nano specifications

Component	Specifications
Microcontroller	ATmega328P – 8-bit AVR family microcontroller
Operating voltage	5 V
Recommended input voltage for vin pin	7–12 V
Analog input pins	6 (A0–A5)
Digital I/O pins	14 (Out of which 6 provide PWM output)
DC current on I/O pins	40 mA
DC current on 3.3 V Pin	50 mA
Flash memory	32 KB (2 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (clock speed)	16 MHz
Communication	IIC, SPI, USART

The radiofrequency module (RF Module), as shown in Fig. 2 is an electronic device used to transmit and receive radio frequency signals over a particular distance. It consists of a Rf transmitter and Rf receiver. The transmitter is placed on the transmitter system on the glove and is used to communicate the data from the accelerometer to the robotic vehicle. The receiver of the module is used in the receiver circuit of the car to receive the signals being transmitted to the robotic vehicle. In this study, the RF transmitter is interfaced with the Nano board and receives instructions to be transmitted. At the same time, the RF receiver is coupled with the Arduino Nano on the receiver circuit on the robotic vehicle to relay instructions that have been sent. The RF module used in this project transmits and receives signals at a frequency of 433 MHz. Through RF communication; this project aims to implement the control system over a distance of 20–30 m.

4.1 Transmitter Unit Flowchart and Algorithm

For the transmitter unit, a gesture sensing code was used in the interfacing of the Arduino and the accelerometer. The gesture sensing flowchart is shown in Fig. 3.

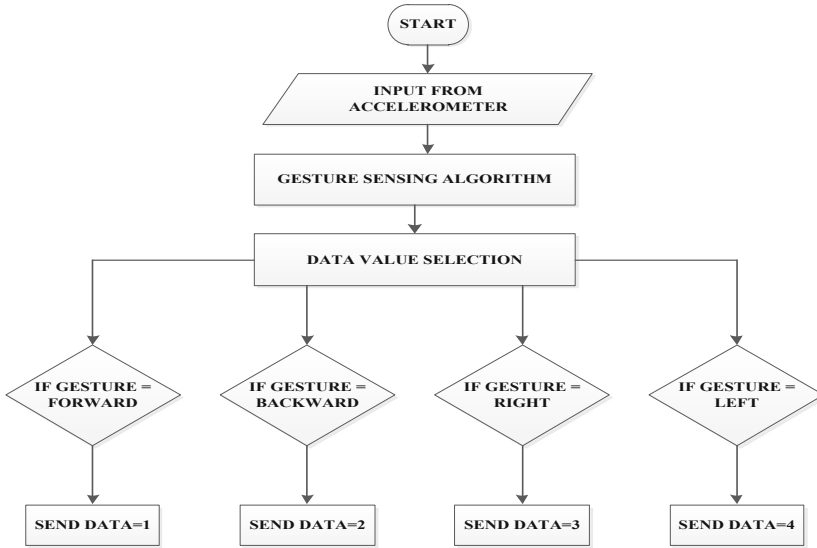


Fig. 3. Transmitter unit flowchart

4.2 Transmitter Unit Flowchart and Algorithm

- Step 1: Start System and Initiate libraries for accelerometer and rf module
- Step 2: Record values of coordinates of MPU 6050 as X-axis, Y-axis, and Z-axis
- Step 3: Wait to detect changes in coordinate values.
- Step 4: If coordinates are unchanged, value of data sent = 0
- Step 5: If Y axis > 1 and $-5 < = X \text{ axis} < = 5$, data = 1 else go to step 6
- Step 6: If Y axis < -1 and $-5 < = X \text{ axis} < = 5$, data = 2 else go to step 7
- Step 7: If X axis < -1 and $-5 < = Y \text{ axis} < = 5$, data = 3 else go to step 8
- Step 8: If X axis > 1 and $-5 < = Y \text{ axis} < = 5$, data = 4 else go to step 4
- Step 9: Send data value to transmitter. STOP.

4.3 Receiver Unit

The hardware for the receiver unit is a robotic vehicle chassis with 4 wheels and 2 motors. It consists of the receiver part of the RF transmission module, an Arduino Uno that serves as the microcontroller for the circuit and a motor driver integrated circuit to enable movement of the vehicle. The circuit diagram for the receiver unit is shown in Fig. 4.

From Fig. 4, the Arduino Uno is used in the receiver circuit of this project to interpret the signals received through the RF module from the transmitter circuit. These signals

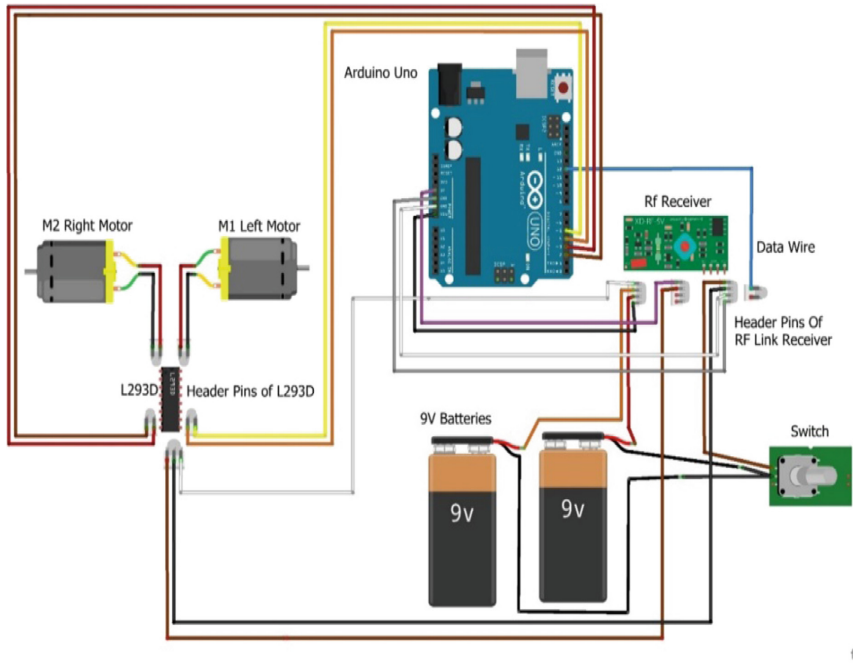


Fig. 4. Receiver unit circuit diagram

would be used to control the motor driver which then enables the robotic vehicle to move. The microcontroller board would serve as the brain of the receiver circuit. It receives the data value being sent from the transmitter circuit and translates it into the desired function to the motor driver to enable movement of the robotic vehicle. In this study, there are two motors with two pins: m1p1, m1p2, m2p1, and m2p2. These pins can have one of two states, HIGH or LOW. When the first pins of both motors are HIGH, the car moves forward. When the pins 2 of both motors are HIGH, the car moves backward. These indicate that current flow is in the opposite direction, therefore, changing the direction of the DC motor movements. When pin 1 of motor 1 and pin 2 of motor 2 are HIGH, the car moves right and when pin 2 of motor 1 and pin 1 of motor 2 are HIGH, the car moves left.

4.4 Receiver Unit Algorithm and Flowchart

The receiver unit is controlled by the Arduino Uno board. A flowchart of the unit is shown in Fig. 5.

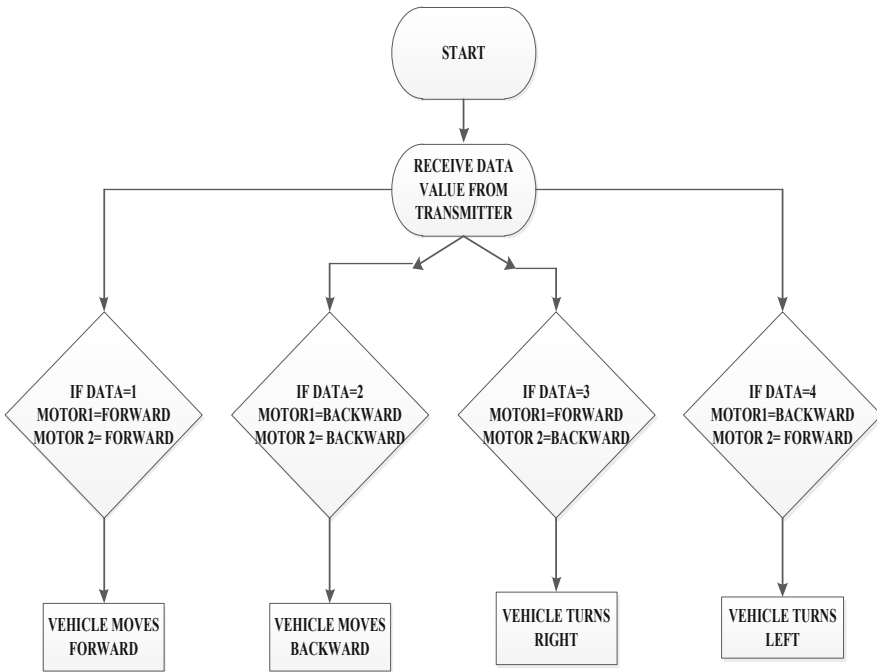


Fig. 5. Receiver unit flowchart

4.5 Receiver Unit Algorithm

Step 1: Start and enable the Rf library.

Step 2: Wait to receive data value sent from transmitter circuit

Step 3: If data = 0, go to step 2 else go to step 4

Step 4: If data = 1, set m1p1 and m2p1 to HIGH and m1p2 and m2p2 to LOW, else go to step 5

Step 5: If data = 2, set m1p1 and m2p1 to LOW and m1p2 and m2p2 to HIGH, else go to step 6

Step 6: If data = 3, set m1p1 and m2p2 to HIGH and m1p2 and m2p1 to LOW, else go to step 7

Step 7: If data = 4, set m1p2 and m2p1 to HIGH and m1p1 and m2p2 to LOW, else go to step 2

4.6 Transmitter Implementation

The transmitter unit of the proposed system is set up on a simple hand glove to be worn by the human operator. The human operator wears the hand glove and uses the necessary gestures to move the robotic vehicle. The gestures accepted by the transmitter unit are shown in Figs. 6, 7, 8 and 9.



Fig. 6. Gesture for forwarding movement



Fig. 7. Gesture for reverse movement



Fig. 8. Gesture of turning right



Fig. 9. Gesture for turning left

4.7 Receiver Implementation

The receiver unit is a robotic chassis with 2 motors and 4 wheels. The two DC motors are controlled to enable movement of the robotic vehicle. It receives the instruction from the transmitter unit and moves in the required direction. The receiver unit is depicted in Fig. 10.

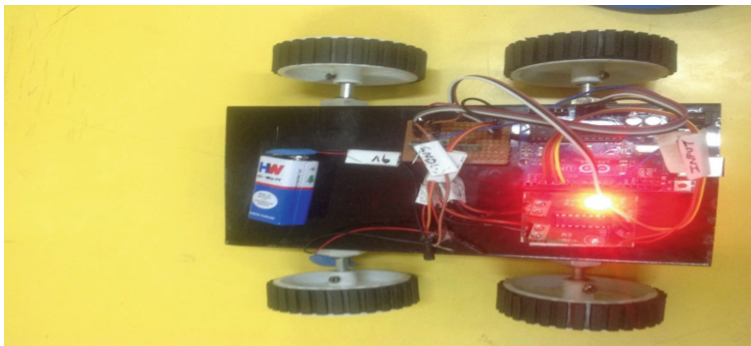


Fig. 10. Receiver unit

The proposed system was tested, and the robotic vehicle was able to be controlled by the accelerometer-based transmitter unit. The transmitter unit was able to recognize the hand gestures specified by the transmitter code as input commands, and the receiver unit was able to receive instructions and move the robotic vehicle according to the input commands received. The range of transmission was also tested, and the radio frequency allowed human operator control over a distance of 15 m.

5 Conclusion and Future Work

In this study, the development of a hand gesture control system for a robotic vehicle provides a more natural mode of communication for human–robot interaction. The technology would increase the applications of robotic vehicles in various sectors through its implementation. The implementation of a hand gesture control system allows for more flexibility in robotic vehicle operation with little training. However, the future development of the proposed system is to fully integrate hand gesture control as the preferred method of robot-human interaction.

Acknowledgements. We acknowledge the support and sponsorship provided by Covenant University through the Centre for Research, Innovation, and Discovery (CUCRID).

References

1. Ben-Ari, M., Mondada, F.: Robots and their applications. In: Ben-Ari, M., Mondada, F. (eds.) *Elements of Robotics*. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-62533-1_1
2. Yinka-Banjo, C., Daniel, O., Misra, S., Jonathan, O., Florez, H.: Comparative analysis of three obstacle detection and avoidance algorithms for a compact differential drive robot IN V-Rep. In: Florez, H., Leon, M., Diaz-Nafria, J., Belli, S. (eds.) *Applied Informatics. ICAI 2019. Communications in Computer and Information Science*, vol. 1051, pp. 357–369. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32475-9_26
3. Husti, I.: Possibilities of using robots in agriculture. *Hungar. Agric. Eng.* **35**, 59–67 (2019)
4. Bartneck, C., Hu, J.: Rapid prototyping for interactive robots. In: 8th Conference on Intelligent Autonomous Systems (IAS-8), Amsterdam, Netherlands (2004)
5. Vantsevich, V.V., Blundell, M.V. (eds.): *Advanced Autonomous Vehicle Design for Severe Environments*, vol. 44. IOS Press (2015)
6. Stančić, I., Musić, J., Grujić, T.: Gesture recognition system for real-time mobile robot control based on inertial sensors and motion strings. *Eng. Appl. Artif. Intell.* **66**, 33–48 (2017)
7. Coronado, E., Villalobos, J., Bruno, B., Mastrogiovanni, F.: Gesture-based robot control: design challenges and evaluation with humans. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2761–2767. IEEE (2017)
8. Goodrich, M.A., Schultz, A.C.: Human-robot interaction: a survey. *Found. Trends® Hum.-Comput. Interact.* **1**(3), 203–275 (2008)
9. Aggarwal, L., Gaur, V., Verma, P.: Design and implementation of a wireless gesture controlled robotic arm with vision. *Int. J. Comput. Appl.* **79**(13), 24–30 (2013)
10. Gesture | Definition of Gesture by Merriam-Webster. <https://www.merriam-webster.com/dictionary/gesture>. Accessed 22 Mar 2019

11. Yousefi, S., Li, H.: 3D hand gesture analysis through a real-time gesture search engine. *Int. J. Adv. Robot. Syst.* **12**(6), 67 (2015)
12. Fang, G., Gao, W., Zhao, D.: Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Trans. Syst. Man Cybern.-Part a: Syst. Hum.* **37**(1), 1–9 (2006)
13. Sarkar, A., Patel, K.A., Ram, R.G., Capoor, G.K.: Gesture control of drones using a motion controller. In: 2016 International Conference on Industrial Informatics and Computer Systems (CIICS), pp. 1–5. IEEE (2016)
14. Nayak, S., Nalini, J., Deepak, B.B.V.L.: Development of gesture controlled robot using a 3-axis accelerometer. *JoCI* **23**, 34 (2016)
15. Shaari, A.R., Jani, M.N.M., Yunus, A.S.M.: A smart wheelchair prototype based on hand gesture control. *J. Eng. Sci. Res.* **1**(2), 209–214 (2017)
16. Mapuskar, A., Kharade, A., Kedari, K., Shah, S., Gaikwad, K.: Robot controlled using hand motion recognition. *Int. Res. J. Eng. Technol.* **4**(4), 424–427 (2017)
17. Vilaça, R., Ramos, J., Sepúlveda, J., Esteves, J.S.: Mobile platform motion control system based on human gestures. *Int. J. Mechatron. Appl. Mech.* **1**, 267–273 (2017)
18. Achari, S.M., Mirji, S.G., Desai, C.P., Hulasogi, M.S., Awari, S.P.: Gesture-based wireless control of robotic hand using image processing. *Int. Res. J. Eng. Technol. (IRJET)* **5**(5), 3339–3345 (2018)
19. Azeta, A.A., Misra, S., Azeta, V.I., Osamor, V.C.: Determining suitability of speech-enabled examination result management system. *Wirel. Netw.* **25**(6), 3657–3664 (2019)
20. Olanrewaju, L., Oyebiyi, O., Misra, S., Maskeliunas, R., Damasevicius, R.: Secure ear biometrics using circular kernel principal component analysis, Chebyshev transform hashing and Bose–Chaudhuri–Hocquenghem error-correcting codes. *Signal Image Video Process.* **14**, 1–9 (2020)
21. Camalan, S., Sengul, G., Misra, S., Maskeliunas, R., Damaševičius, R.: Gender detection using 3D anthropometric measurements by Kinect. *Metrol. Meas. Syst.* **25**(2), 253–267 (2018)
22. Temiatse, O.S., Misra, S., Dhawale, C., Ahuja, R., Matthews, V.: Image enhancement of lemon grasses using image processing techniques (histogram equalization). In: Panda, B., Sharma, S., Roy, N. (eds.) *Recent Developments in Science, Engineering and Technology*, vol. 799, pp. 298–308. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-8527-7_24



An Intelligent Recommendation System Based on Collaborative Filtering and Grid Structure

Animesh Chandra Roy^(✉) and Mohammad Shamsul Arefin^(✉)

Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh
sarefin@cuet.ac.bd

Abstract. The rapid growth of location-based networks during the twenty-first century has greatly increased, so there is a need of providing suggestions to persons about their interest activities. Nowadays, location-based social networks (LBSN) become a common platform for users to share interests. In this paper, our main concern is to design a recommendation system that will provide suggestions to the user according to their interests. We have developed a framework based on Collaborative Filtering (CF) that analyses user activities to find the similar user. CF helps us to enrich each user profile by rating unvisited places which we can include in their interest hierarchy. Then we calculate the similarity of the user profile with the Point of Interest (POI) extracted from the user's current location and make recommendations. Here Grid Structure is used to analyse the POIs extracted from Google.

Keywords: Recommendation system · Collaborative Filtering (CF) · Grid structure

1 Introduction

Information Retrieval (IR) technologies have gained outstanding prevalence in the last two decades with the explosion of massive online information repositories. Recommendation system (RS) is one of the most effective and efficient features that has been initiated by the LBSNs. Based on the user's interest this system can recommend the users about different places and activities which are matches their interest [7]. An efficient and accurate recommender system is very much needed in this highly social network era because of the rapid improvement of Internet Technology. This system collects the user information which can include location, entertainment, activities, games, and traveling destination, etc. based on their choices. After considering this information Recommender systems provide a recommendation to the user. This prediction reflects on the user's preference history. These recommendations make the Social network more strengthen because it connects one to another through their common interest. The recommendation can also provide a suitable place, where a group of friends can meet together according to their preferred type of places and previously tagged in places. As RS considers the user profile, it calculates the similarity between different profiles and suggests the best

matching. In this paper, we have developed a framework based on grid structure to make a suggestion, in which each recommendation is generated after calculating the similarity of users categorized preferences and geotagged places.

1.1 Motivating Example

Because of the availability of modern GPS technology people tends to use it in their daily life to make it easier. Most of the existing collaborative filtering methods compare between the similar user and taking into account their check-in spots to provide recommendations. Whenever a user comes to visit a new place, it is necessary to suggest some POIs according to the user's interests. Since the existing works do not consider the semantic analysis of each user's interests[10]. By analyzing their choices we can easily avoid irrelevant places to recommend.

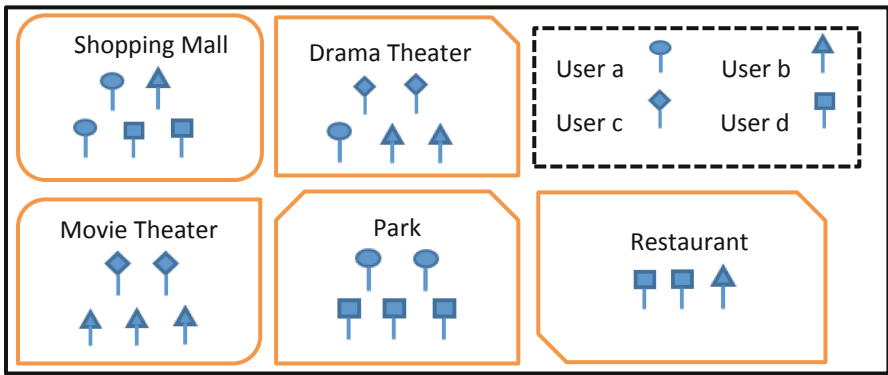


Fig. 1. Users check-in points

Now, consider four user's check-in points in Fig. 1. Here each user's check-in is pointed as a point of interest (POI) and their visiting summary is presented by Table 1. Here, user b seems like to visit 'Movie Theater' and 'Drama-theater' a lot. So our system analyses the behavior and suggests an 'entertainment' type place when the user comes to a new visiting place. Thus it can avoid unnecessary suggestions that do not match with users interest. Our proposed method works as follows:

- At first, we create a user profile based on their previous check-in history and make the hierarchy of their choices.
- Next, we extract all the possible POIs from the user's current location and divide them into grids.
- Then we measure the similarity score between the user's profile and extracted POIs from the grid.
- Finally, we select top-scoring POIs to suggest with the shortest distance.

Table 1. User check-in information

Place	User a	User b	User c	User d
Shopping mall	2	1	0	2
Drama-theater	1	2	2	0
Movie theater	0	3	2	0
Park	2	0	0	3
Restaurant	0	1	0	2

The rest of the paper is described as follows: In Sect. 2, some related works have been discussed. In Sect. 3, there are some preliminary topics related to our system. In Sect. 4 briefly we describe our framework. Section 5 describes the dataset and implementation of our system. Finally, Sect. 6 concludes and gives an idea about future research.

2 Related Works

Ashbrook et al., [1] proposed a predictive system to locate the future movements of the users. It uses Global Positioning System (GPS) to collect location data and other information such as other people's presence. These data are then clustered into different scales and then the Markov model introduced to incorporate those locations for prediction. Herlocker et al., [2] they reviewed some key metrics for evaluating a recommender system such as RMSE, Precision, Recall, Prediction-rating correlation etc. Adomavicius et al., [3] described filtering methods into three categories: content-based, collaborative, and hybrid recommendation approaches. They also showed some limitations and future extensions of a recommender system. Horozov et al., [4] introduced an enhanced collaborative filtering method using the location parameter generating recommendations. Park et al., [5] introduced a personalized recommendation system based on Bayesian Networks (BN), which uses the user's location, surrounding context, and time. Considering user requests this map-based system provides services by displaying onto the minimap. Chow et al., [6] provides a comprehensive system that covers three dimensions of location-based services such as, newsfeed, news ranking, and recommendations. They work on the designing of location and rank aware query operators, materializing query answers, and providing privacy-aware query processing.

In [7, 8, 10] used user-based CF method and a hierarchical-graph-based similarity measurement (HGSM), is introduced to create a model on each user's check-in history, and measuring the similarity of check-in activities between users. This framework is built on three factors, 1) the sequence data of an individual's outdoor movements, 2) the volume of a visited geospatial region, and 3) the ordered data of geographic spaces. They used a content-based collaborating filtering method to find the suggestions. Lee et al., [9] used a semantic approach to measure the similarity using the location. In [11] a machine learning technique is used for predicting a user's location. The dataset consists of the user's points of interest (POI) or venues based on their social activities and interests. To solve data sparsity, they proposed a Probabilistic Neural Network

which gave better results considering the other two types of neural networks. Mu et al., [12] proposed a collaborative method to provide service recommendations based on the service properties. At first, this framework builds a preference model of service property based on each user's information. Then, measures the service similarity score of two services of each user. Finally, the Pearson correlation coefficient of the similarity score of two services shows the final result. In [13] a novel approach was introduced by using a user-location vector to represent the relationship between user and POI. Liao et al., [14] proposed a strategy for recommendation using tensor factorization. At first, the user's information is extracted using Latent Dirichlet Allocation(LDA) to generate a probability distribution for the extracted information. Secondly, the user's check-in information is separated into a different category. Finally, the singular value decomposition (SVD) algorithm is applied to for POI recommendation. In [15] they proposed a recommendation after considering user rating and item attributes. Here, a weighted control coefficient is used to find the nearest neighbors.

3 Prelimineries

3.1 Check-In Information

Check-in information defines those physical places where users usually visit more often and share their feelings. These days online social services, such as Instagram, Facebook, and Tripadvisor provides such kind of check-in option for user using a mobile application. Global Positioning System (GPS) allows us to locate those places.

3.2 Point of Interest (POI)

Point of Interest (POI), defines the categorical partitioning of all the user's check-in information. Each check-in data is classified into a specific category presented by a POI. Mainly some GPS oriented software, GIS technology uses the POI concept. POI can be defined by geometrical coordinate as the latitude and the longitude of a specific place.

Table 2. POI data

Location	Rating	Co-ordinate	Category
Banani coffee house	4.0	<23.777176, 90.399452>	Food
Chittagong complex	3.75	<22.341900, 91.815536>	Shopping
Shapla movie theater	4.25	<25.744860, 89.275589>	Entertainment

For example, in Table 2, it shows some contents of a POI database. Here POI database contains location, co-ordinates, rating, and category of this information. The proposed method intended to extract the category of each user and find similarities with the new location. The similarity will help to recommend new places for the users to visit.

4 System Architecture and Design

Our proposed framework of the recommender system consists of three main parts: finding user interest, measuring similarity, and generating recommendations. Recommendation generation section divided into two components: recommend to a single user and for multiple users.

4.1 Discover Interest Using Density-Based Spatial Clustering

To identify users' interests, we must study on check-in places.

Nowadays, people used to visit a variety of places for different reasons. Some places like a movie theater, park, restaurant are visited with a lot of interests. Some of them are visited for regular activities such as colleges, universities. More often people rarely visit a place with interest because of its accessibility.

We used the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm for clustering the check-in spots. It makes clusters with high-density regions which are separated from one another. In figure, P1, P2, : : , P8 are the clusters. Then we locate the gravity center for each cluster and store it in the user's table. Any spot that is closest to the center of a cluster named CAFE will be categorized into Food.

Two parameters (ϵ , $MinPts$) are necessary for DBSCAN: ϵ defines a value if one point can be considered as a neighbor and $MinPts$ defines a minimum number of data points within ϵ radius. A point with more data points than ϵ is a core point and the least data point is a border point for a cluster. Steps for DBSCAN:

- Select an unvisited point and extract all data points within ϵ , and find the core point.
- Form a cluster with the selected points if they are not assigned to any other cluster.
- Recursively add density connected points until a border point is found and create a new cluster. The point which doesn't belong to a cluster is a noise.

Then, a Semantic Hierarchical Category-graph Framework (SHCF) [10] was applied shown in Fig. 2. SHCF consists of three layers, clusters from DBSCAN are in the bottom layer, the middle and top layer generates the general category of the POIs. From the figure we can see that, in "layer 1", these are check-in clusters. In "layer 2", these are the sub-type category nodes. And, the top "layer 3", contains the super-type interest categories. Thus SHCF generates each user's activity profile based on their interests. Whenever a user comes to a new place, this framework helps to find the best suitable suggestions for the user.

4.2 Collaborative Filtering (CF)

Collaborative filtering is a well-known mechanism for finding recommendations. To experiment with CF we collect user's check-in information with their reaction to those places. It filters out those items that can be liked by the user based on the reactions by similar users. It also helps us to enrich each user's interest profile previously generated by SHCF. Generally, the CF algorithm produces a rating on unvisited places based on

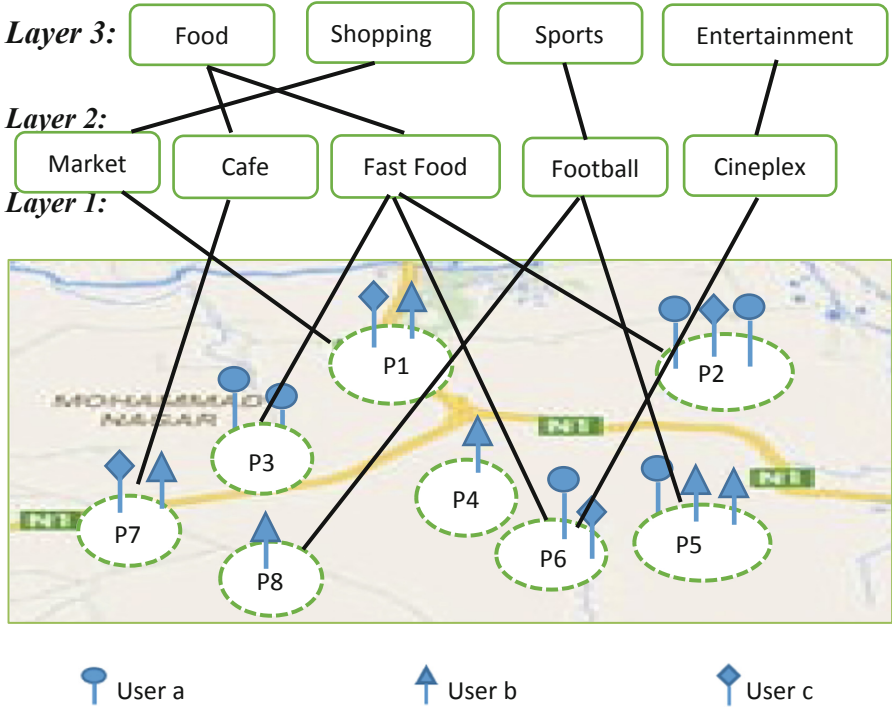


Fig. 2. Semantic Hierarchical Category-graph Framework (SHCF)

the dataset of previously rated check-ins by similar users. If we consider the rating of unvisited places is $r_{u,p}$, where for user u and place p is calculated by an aggregate of the ratings of N similar users.

$$r_{u,p} = \text{aggr}_{u^i \in u^\wedge} r_{u^i,p} \tag{1}$$

Where u^\wedge defines the N similar users to user u who have rated.

4.2.1 Discovering Similar Users and Unvisited Places

To measure the similarity of user and their unvisited spots we used Pearson Correlation Coefficient. It represents the relationship between the two users on a scale ratio or an interval. The coefficient ranges in between $+1$ to -1 , where $+1$ indicates positive relationship, -1 represents perfect negative and 0 defines no relationship. Pearson correlation coefficient formula works as follows:

$$r_{x,y} = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{(N \sum x_i^2 - (\sum x_i)^2)(N \sum y_i^2 - (\sum y_i)^2)}} \tag{2}$$

Where, $r_{x,y}$ defines the similarity score between user x and y .

User visits different places with various rates of reaction. Here we try to find the similarity score between the user. Based on the similarity score it can say that an unvisited

place might be a point of interest for a particular user. Thus it helps us to update the user profile combining both previously visited and unvisited places.

Table 3 shows the matrix representation of user to user similarity score.

Table 3. Similarity matrix

Similarity score	u_a	u_b	u_c	...	u_n
u_a	1	$SimScr_{a,b}$	$SimScr_{a,c}$...	$SimScr_{a,n}$
u_b	$SimScr_{b,a}$	1	$SimScr_{b,c}$...	$SimScr_{b,n}$
u_c	$SimScr_{c,a}$	$SimScr_{c,b}$	1	...	$SimScr_{c,n}$
...
u_n	$SimScr_{n,a}$	$SimScr_{n,b}$	$SimScr_{n,c}$...	1

4.2.2 Grid Structure Generation

Before calculating the similarity between the user profile and POI spots extracted from google, we divide each type of geolocated (latitude & longitude) POIs into different grids. Each geolocation is converted into a cartesian coordinate and used in a grid-based data structure. Figure 3 shows the grid structure of our system where each grid contains various kinds of POI that are extracted from Google place API. Now we construct a Table 3 from this grid structure that shows the detailed information of the grid. For example, G13 contains three POIs of the type 't' for theater, 'c' for cafe, and 'p' for the park. Some of the grid-like G12, G2, etc. does not contain any single POI, thus we can neglect those for our calculation. Now we have to measure the similarity of these listed grids and each user profile for recommendation generation. The algorithm describes the grid structure generation process.

Algorithm 1: Dividing given points into N * N grid

Input: places & N. Places contains each of the place geo-location and their type. N is the value of N * N grid.

Output: Location of the POIs in a N * N grid.

Begin

for all poi \in places do

 poi.cartesian \leftarrow geolocation_to_cartesian (poi.geolocation)

end for

 minx \leftarrow minimum(point.cartesian.x); for all point \in places

 maxx \leftarrow maximum(point.cartesian.x); for all point \in places

 miny \leftarrow minimum(point.cartesian.y); for all point \in places

 maxy \leftarrow maximum(point.cartesian.y); for all point \in places

 gridsizeX \leftarrow (maxx - minx)/N; Size of grid in X axis

 gridsizeY \leftarrow (maxy - miny)/N; Size of grid in Y axis

End

4.3 Recommendation Generation

We measure the user interest by combining the user's check-in history and CF-based similarity correlation. The user profile also defines the priority rate for each choice of interest. Now we have to generate recommendations for a user. We need to divide extracted POI related geo-location surrounding on user's current location into different grids and made a list of POI table. We calculate the similarity between grid wise POIs and user's interest profile. A higher similarity score on a POI defines the required recommendations.









<i>G31</i>	<i>G32</i>	<i>G33</i>  _t  _t
<i>G21</i>	<i>G22</i>  _c  _t	<i>G23</i>
<i>G11</i>  _p	<i>G12</i>	<i>G13</i>  _t  _p  _c

Fig. 3. Grid structure

Formula for score generation is as follows:

$$poi.score = p_s * poi.type * r \quad (3)$$

Where,

p_s is the priority score of user choices,

r is rating of POI.

Here we consider the rating of each POIs because in case of the same type of suggestions it will help us to find a better solution. The minimum distance between the user and suggested places is one of our concerns for the optimal solution. Before making the suggestion the manhattan distance between user location consisting grid and the grid contains the POI is calculated.

$$poi.man_{dis} = |poi.grid.x - user.grid.x| + |poi.grid.y - user.grid.y| \quad (4)$$

$$poi.score = (p_s * poi.type * r) / poi.man_{dis} \quad (5)$$

5 Experiment

In our experiment we take check-in places from real world and make the recommendations of our system.

5.1 DataSet and Implementation

We used a synthetic dataset where each user contains information about user_name, spot_type, latitude, longitude, rating to generate check-in spots. Table 4 shows the check-in dataset.

Table 4. Check-in information

User_name	Spot_type	Latitude	Longitude	Rating
Smith	Shoe store	-53.8343	64.99585	4.2
Carter	Park	-82.4145	-94.8644	3.5
Robinson	Spa	-11.7012	-4.23816	4.4
Bailey	Art gallery	6.171611	49.443224	4.5

In our work, the total number of check-in spots is 57,753 of 35,515 users. We considered 15,000 users who checked more than 20 spots and used DBSCAN clustering around a 100-m radius. In Pearson correlation formula we used 200 users profile for similarity calculation.

5.2 Evaluation

Precision and Recall are very well-known metrics to demonstrate a predictive system. To evaluate our system we considered 10 users for testing. There can be four possible outcome for place suggestion; true positive (t_p) - accurately predicted the recommendation, false positive (f_p) - predicted a recommendation that does not belong to a user interest, false negative (f_n) - failed to predict the actual places, true negative (t_n) - successfully avoids the places that are not on the profile.

$$precision = \frac{t_p}{t_p + f_p} = 0.9 \quad \text{and} \quad recall = \frac{t_p}{t_p + f_n} = 0.9$$

The precision and recall value shows a very good outcome.

5.3 Example of Recommendation

We tested our work in Chittagong town, Bangladesh for different users. Our system extracts the user's current location and provides appropriate recommendations. For example, Fig. 4 shows some restaurants nearest to a user as a recommendation.



Fig. 4. Recommendation for a user

6 Conclusion

In this paper, we introduced Density-Based Spatial Clustering (DBSC) and a semantic hierarchical category graph framework (SHCF) to categorize the spots visited by the user. We also used Collaborative filtering to find out an unvisited place that might be a choice of activity for a user. we have tried to find the nearest possible places from the user's current location for a recommendation, which was a success.

Our work does not cover an option to suggest a group of users for a meeting place according to their similarity. In the future, it can be considered which will give more visiting options to the user.

References

1. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Springer, London (2003)
2. Herlocker, L.J., Konstan, A.J., Terveen, A.G., Riedl, T.J.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**(1), 5–53 (2004)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
4. Horozov, T., Narasimhan, N., Vasudevan, V.: Using location for personalized POI recommendations in mobile environments. In: *Proceedings of the Symposium on Applications and the Internet* (2006)
5. Park, M., Hong, J., Cho, S.: Location-based recommendation system using Bayesian user's preference model in mobile devices. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) *UIC 2007. LNCS*, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73549-6_110
6. Chow, C., Bao, J., Mokbel, F.M.: Towards location-based social networking services. In: *ACM LBSN*, 2 November 2010
7. Zheng, Y., Zhang, L., Xie, X.: Recommending friends and locations based on individual location history. *ACM Trans. Web* **5**(1), Article 5 (2011)
8. Berjani, B., Strufe, T.: A recommendation system for spots in location-based online social networks. *ACM*, 10 April 2011
9. Lee, M., Chung, C.: A user similarity calculation based on the location for social network services. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) *DASF AA 2011. LNCS*, vol. 6587, pp. 38–52. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20149-3_5
10. Chen, H., Arefin, M.S., Chen, Z., Morimoto, Y.: Place recommendation based on users check-in history for location-based services. *Int. J. Netw. Comput.* **3**(2), 228–243 (2013)
11. Kosmides, P., Remoundou, C., Demestichas, K., Loumiotis, J., Adamopoulou, E., Theologou, M.: A location recommender system for location-based social networks. In: *Proceedings of Mathematics and Computers in Sciences and in Industry* (2014)
12. Mu, W., Meng, F., Chu, D.: A collaborative filtering recommendation algorithm based on user preferences on service properties. In: *Proceedings of Service Sciences* (2014)
13. Ajantha, D., Vijay, J., Sridhar, R.: A user-location vector based approach for personalised tourism and travel recommendation. In: *International Conference on Big Data Analytics and computational Intelligence (ICBDACI)* (2017)
14. Liao, G., Jiang, S., Zhou, Z., Wan, C., Liu, X.: POI recommendation of location-based social networks using tensor factorization. In: *19th IEEE International Conference on Mobile Data Management* (2018)
15. Li, Z., Huang M., Zhang Y.: A collaborative filtering algorithm of calculating similarity based on item rating and attributes. In: *14th Web Information Systems and Applications Conference* (2017)



SMOTE Based Weighted Kernel Extreme Learning Machine for Imbalanced Classification Problems

Roshani Choudhary^(✉) and Sanyam Shukla

Maulana Azad National Institute of Technology, Bhopal, India
<http://www.manit.ac.in/content/drsanyam-shukla>

Abstract. Most of the classification problems in the real-world suffer from class imbalance. The performance of traditional classification algorithms is biased towards the majority class while handling a class imbalance problem. Several methods have been used to handle the class imbalance problems. These methods are based on three main approaches which are data level, algorithmic, and ensemble. Weighted Kernel-based Synthetic Minority Oversampling Technique (WK-SMOTE) is a recently proposed method that utilizes the benefits of both the data level approach and algorithmic approach. Inspired by the idea and performance of WK-SMOTE this work proposes a novel Synthetic Minority Oversampling based Weighted Kernelized Extreme Learning Machine (SMOTE-WKELM). SMOTE-WKELM is a variant of Weighted Kernelized Extreme Learning Machine (WKELM), which uses SMOTE for oversampling of the minority class instances. Experiments are performed on 15 datasets with varying imbalance ratios, downloaded from keel dataset repository for performance evaluation. The results on these datasets show that the proposed method performs better than the other state of the art methods in consideration.

Keywords: Classification · Class imbalance · Oversampling · Synthetic minority oversampling technique · Weighted extreme learning machine

1 Class Imbalance Classification

A classification problem is said to be class imbalanced when the samples in one class outnumber the samples of other classes. The class which has a greater number of samples is known as the majority class and the class with a smaller number of instances is called the minority class. Traditional classification algorithms are biased towards the majority class samples, so the minority class samples are misclassified more often [5]. In real-world applications of classification, the minority class samples are more important like cancer classification [11], medical applications [13], and intrusion detection system etc. So correct classification of minority

instances is required. Several methods have been proposed to handle class imbalance dataset classification which can be classified mainly as data preprocessing, algorithmic and ensemble methods [5].

Algorithmic methods make changes in the classification algorithm to handle class imbalance learning. Various algorithmic variants of traditional classification algorithms have been proposed to solve the imbalance problem. Weighted extreme learning machine (WELM) [15], class-specific cost regularized extreme learning machine (CCRELM) [12], cost-sensitive extreme learning machine (CSELM) [9] and cost-sensitive kernel extreme learning machine CSKELM [10] are some algorithmic variants of the (ELM) extreme learning machine [7]. Data preprocessing methods try to reduce the imbalance ratio of the problem before it is used for developing the classification model. Oversampling and under-sampling are the two popular data preprocessing methods used for class imbalanced learning. Random oversampling and SMOTE [3] are two commonly used methods for oversampling the minority class.

This work proposes a SMOTE-WKELM, which is a hybrid of WKELM and SMOTE. The following subsections provides a brief description of the existing techniques related to the proposed work.

1.1 Weighted Extreme Learning Machine (WELM)

WELM [15] is a algorithmic method which assigns more weight to minority class instances. WELM minimizes weighted least squares error and uses ridge regularization for finding the optimal solution. Two weighting schemes are used in WELM to calculate the weight matrix \mathbf{W} utilizing the class proportion. The proposed work uses weighting scheme1 which is defined as:

$$W_{ii} = \frac{1}{t_k}. \quad (1)$$

Here, t_k is the total number of instances belongs to the k^{th} class. For N training samples $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$. We have $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ as the input feature vector and $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$ as the output label vector. The number of hidden neurons will be set equal to L . Here, vector/matrix transpose is denoted by superscript T . The weight matrix from input to hidden layer is represented by $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_j, \dots, \mathbf{a}_L]^T \in R^{L \times n}$, Where $\mathbf{a}_j = [a_{j1}, a_{j2}, \dots, a_{jn}]$ are the weights connecting the input neurons to the j^{th} hidden neuron. During training time these weights are randomly generated and are not changed further. The hidden neurons bias matrix is denoted by $\mathbf{b} = [b_1, b_2, \dots, b_j, \dots, b_L]^T \in R^L$, Where b_j is the bias of the j^{th} hidden neuron. For i^{th} training/testing sample i.e. x_i , the hidden layer output $h(x_i)$ is calculated as follows:

$$h(\mathbf{x}_i) = G(\mathbf{a}\mathbf{x}_i + \mathbf{b}) \quad (2)$$

Here, $G(\cdot)$ is the activation function used in the hidden layer.

The optimization function of WELM as given in [15] can be reproduced below:

$$\text{Minimize: } \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \sum_{i=1}^N W_{ii} \|\xi_i\|^2 \tag{3}$$

$$\text{Subject to: } \mathbf{h}(\mathbf{x}_i)\beta = \mathbf{t}_i^T - \xi_i^T, \quad i = 1, \dots, N.$$

Using the KKT condition, the result of (3) is determined in [15] as follows:

$$\beta = \begin{cases} \mathbf{H}^T \left(\frac{I}{C} + \mathbf{W}\mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{W}\mathbf{T}, & \text{when } N < L \\ \left(\frac{I}{C} + \mathbf{H}^T\mathbf{W}\mathbf{H} \right)^{-1} \mathbf{H}^T\mathbf{W}\mathbf{T}, & \text{when } N > L. \end{cases} \tag{4}$$

WKELM - WKELM [6] uses the Gaussian kernel function which maps the input data to the feature space as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \tag{5}$$

Here, σ denotes the kernel width parameter, \mathbf{x}_i denotes the i^{th} sample and \mathbf{x}_j denotes the j^{th} centroid $i, j \in 1, 2, \dots, N$. The number of Gaussian kernel function used in KELM [6] was equal to the number of training samples.

The kernel matrix of the hidden layer is given by the following Eq.

$$\mathbf{K} = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \dots & K(x_1, x_N) \\ K(x_2, x_1) & K(x_2, x_2) & \dots & K(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_N, x_1) & K(x_N, x_2) & \dots & K(x_N, x_N) \end{bmatrix}_{N \times N} \tag{6}$$

Here, $K(\mathbf{x}_i, \mathbf{x}_j)$ represents the output of the j^{th} hidden neuron for the i^{th} input \mathbf{x}_i , here $i, j \in 1, 2, \dots, N$. Applying Mercer’s condition to define the kernel matrix of KELM as illustrated in the following Eq.

$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^T : \Omega_{ELM_{i,j}} = h(x_i).h(x_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

The output of the KWELM can be rewritten as follow:

$$\mathbf{f}(\mathbf{x}) = \text{sign} \left[\begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T \left(\frac{I}{C} + \mathbf{W}\Omega_{ELM} \right)^{-1} \mathbf{W}\mathbf{T} \right] \tag{8}$$

Compared to the Sigmoid node based WELM, KWELM has better classification performance [15].

1.2 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE [3] is a synthetic minority oversampling technique which is used to generate a synthetic sample on the line connecting two minority class samples $x_i, x_j \in X_{min}$. Where, X_{min} is the set of the instances belonging to minority class. A synthetic sample X_{new} can be created between x_i and x_j using following equation:

$$x_{new} = x_i + (x_j - x_i) * \lambda \quad (9)$$

Here λ is a random number between $[0, 1]$. The sample x_i is a randomly selected minority class instance and x_j is one of its K nearest neighbor belonging to the minority samples. In SMOTE, K can be decided according to the amount of oversampling required.

1.3 Weighted Kernel-Based Synthetic Minority Oversampling Technique (WKSMOTE)

Weighted kernel-based SMOTE (WK-SMOTE) [8] overcomes the limitation of SMOTE for nonlinear problems by oversampling the minority class in the feature space of support vector machine (SVM) classifier. The performance of SMOTE is limited on nonlinear separable problems especially those with high imbalance ratios [8].

2 Proposed Method

This work proposes SMOTE based WKELM (SMOTE-WKELM) to handle the class imbalance problem efficiently. SMOTE increases the number of minority class instances in the original training dataset by creating synthetic samples in the minority region. It further increases the decision boundary for the minority class into the majority class region. The proposed method does not always fully balance the training data because we have created a limited number of minority samples in the oversampling. This work used WKELM as the classifier, because if the training dataset is not balanced after oversampling then the classifier is still able to handle the class imbalance. The weights of majority and minority class instances are assigned using the Eq. 1. It has been shown in [3] that SMOTE is used to increase the significance of the specific region of the minority class in the decision boundary, which achieves better generalization performance in classification. So, the proposed SMOTE-WKELM combines the SMOTE technique with a cost-sensitive method i.e. WKELM which provides different weights to minority and majority class instances in case the SMOTE is not able to fully balance the training dataset. Algorithm 1 demonstrates the pseudo-code of the proposed SMOTE-WKELM.

Benefit of Combining SMOTE with WELM - For dataset with high imbalance like medical data or IDS (intrusion detection system) data, it is not feasible

to fully balance the dataset using any oversampling method. Fully balancing a highly imbalance dataset will lead to overfitting. So, this work uses SMOTE to oversample given dataset to reduce the class imbalance ratio followed by model generation using WKELM.

3 Experimental Setup and Result Analysis

To evaluate the performance of proposed method some experiments are performed. The performance of proposed algorithm is compared with some of the other state of art methods, which are KELM [14], WKELM [15], CCR-KELM [12] and WKSMOTE [8]. KELM, WKELM and CCR-KELM are the algorithmic variants of ELM [7] and WK-SMOTE [8] is a SVM [4] based classification method.

Algorithm 1. Proposed SMOTE-WKELM

Input: The training dataset: $\{(\mathbf{x}_i, \mathbf{t}_i) \mid \mathbf{x}_i \in R^n, \mathbf{t}_i \in R^m, i = 1, 2, \dots, N\}$,
 X_{min} : Minority class instances
 N_{min} : Number of minority class instances
 N_{max} : Number of majority class instances
 K : Number of nearest neighbours
 O : Maximum Number of synthetic data to be generated
 $O = N_{max} - N_{min}$
Output: SMOTE-WKELM model for classification
procedure SMOTE-WKELM
for $i = 1$ **to** N_{min} **do**
Find K nearest neighbours of instance $x_i \in X_{min}$ among other minority instances in X_{min}
 $x_j = KNN(x_i, X_{min})$
for $j = 1$ **to** K **do**
Generate a synthetic sample x_{new} between x_i and x_j using equation 9
Add x_{new} to set S
end for
end for
if $(O > (N * K))$
Add all synthetic samples in S to training set X
 $X_{train} = X \cup S$
else
Randomly select O samples from S and add to Training set X
 $X_{train} = X \cup S(random(O))$
end if
Determine the output kernel matrix for training data in X_{train} using Eq. 9
Determine the final class prediction by employing Eq. 8
return β
end procedure

3.1 Dataset Specification

The experiments were performed to evaluate the performance of proposed SMOTE-WKELM classifier by using 15 benchmark binary class imbalanced datasets available in the 5-fold cross validation format. These datasets are obtained from KEEL data repository [1, 2], which is an on-line dataset repository. Our work uses normalized dataset, normalization is done to map the numbers within a range 1 and -1 for all attributes.

Table 1. G-mean obtained on various datasets using different algorithms

Dataset	KELM	WKELM	CCR-KELM	WKSMOTE	σ_1	C_1	SMOTE-KELM	σ_2	C_2	SMOTE-WKELM
abalone9-18	76.56	89.76	71.35	91.94	8	12	84.82	4	14	93.12
ecoli-0-1_vs-5	88.36	88.36	88.36	88	0	0	88.36	0	10	91.34
glass0	85.35	85.65	85.35	78	-4	-2	85.77	-2	12	85.78
glass1	79.38	80.35	76.6	73	-8	-18	79.03	-8	-18	79.03
glass2	66.28	82.59	66.59	79	-2	8	81.14	10	40	82.47
glass4	87.22	91.17	87.22	89	-12	-18	92.46	-12	-18	92.46
glass6	92.84	94.04	93.77	90	8	22	94.32	6	26	94.04
pima	71.28	75.58	73.61	74	0	-4	76.41	0	4	76.70
vehicle0	99.36	99.30	97.81	75.52	2	8	99.46	2	16	99.46
vehicle2	99.22	99.45	99.13	99.24	0	10	99.53	2	20	99.53
yeast-1_vs-7	68.32	77.72	68.32	76	10	46	69.01	-2	6	79.85
yeast-2_vs-4	86.27	91.88	86.27	80	-2	4	86.27	2	16	92.00
yeast-2_vs-8	78.91	77.89	78.91	78.57	-2	22	78.91	0	18	82.62
shuttle-c2-vs-c4	94.14	100.00	100	100	-12	-18	94.14	0	-18	100.00
pageblocks-13vs4	97.89	98.07	97.84	97.38	-8	-18	97.89	0	10	100.00

3.2 Parameters Settings

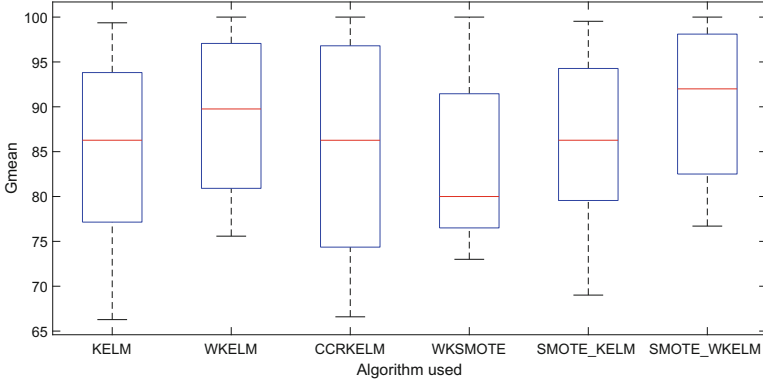
The proposed SMOTE-WKELM uses the Gaussian kernel function to maps the input feature space to the higher dimensional space. Here, the number of kernels is set equal to the number of the training instances after oversampling the training data. This work shows the average performance of the five-folds obtained from the five-fold cross-validation. The best values of the regularization coefficient denoted as C and the kernel width parameter denoted as σ are identified by grid search on $\{2^{-18}, 2^{-16}, \dots, 2^{48}, 2^{50}\}$ and $\{2^{-18}, 2^{-16}, \dots, 2^{18}, 2^{20}\}$ during the model selection phase. This work uses only training dataset for tuning these parameters. SMOTE based minority oversampling approach requires to select the number of nearest neighbors i.e. K . In this work, K is set equal to three as recommended in the original implementation of SMOTE [12].

3.3 Experimental Results

Accuracy is not an appropriate performance matrix for evaluation of a class imbalance problem [5]. This work uses G-mean for performance evaluation. Table 1 shows the G-mean results of various algorithms on the datasets used

Table 2. T-test statistical results on G-mean given in Table 1

Methods compared	Stats	p	h(0.05)
KELM vs SMOTE_KELM	[-4.790501476442684; -0.026458523557312]	0.04782802	1
WKELM vs SMOTE_KELM	[-0.092822253024254; 3.333862253024252]	0.061965218	0
CCR_KELM vs SMOTE_KELM	[-5.325967179341631; 0.476340512674957]	0.094658886	0
WKSMOTE vs SMOTE_KELM	[-6.680571290222234; 1.633611290222229]	0.213948125	0
KELM vs SMOTE_WKELM	[-8.191328453710561; -2.076484879622766]	2.89E-03	1
WKELM vs SMOTE_WKELM	[-2.007231741897072; -0.202581591436261]	0.01992916	1
CCR_KELM vs SMOTE_WKELM	[-8.674443046522690; -1.626036953477315]	7.32E-03	1
WKSMOTE vs SMOTE_WKELM	[-8.554394960904572; -1.943418372428765]	4.26E-03	1

**Fig. 1.** Boxplot for Gmean on various datasets given in Table 1

for comparison. In Table 1 σ_1 and C_1 are the parameters of SMOTE-KELM and σ_2 and C_2 are the parameters of SMOTE-WKELM. The significance of proposed SMOTE-WKELM is shown by performing T-test on the Gmean-results provided in Table 1. The T-test results are provided in Table 2, which shows that proposed SMOTE-WKELM outperforms KELM, WKELM, CCRKELM, WKSMOTE. Figure 1 shows the box-plot diagram for G-mean obtained on the various datasets in use. It can be observed from the box-plot diagram in Fig. 1 that proposed SMOTE-WKELM has more average G-mean compared to other methods. Also the degree of dispersion for the proposed SMOTE-WKELM is less than other methods in consideration.

4 Conclusion and Future Work

The proposed SMOTE based WKELM utilizes benefit of both oversampling and cost sensitive algorithm. To reduce the class imbalance present in a problem, this work creates synthetic samples of minority class using SMOTE. To handle the class imbalance present after oversampling, this work uses WKELM which is a cost sensitive algorithm. The various experiments performed to evaluate the performance of proposed work shows that the proposed method performs

better than other methods in consideration. In the proposed work the synthetic minority samples created using SMOTE are generated in the input space. The proposed method is limited to solve binary class imbalance problems, in future it can be extended to solve multi-class imbalance problems. In future, this work can be extended to generate the synthetic samples in the kernel space, like WK-SMOTE.

References

1. Alcalá-Fdez, J., et al.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Multiple-Valued Log. Soft Comput.* **17**, 255–287 (2010)
2. Alcalá-Fdez, J., et al.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.* **13**, 307–318 (2009)
3. Chawla, N., Bowyer, K., Hall, L.O., Philip Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)* **16**, 321–357 (2002)
4. Corinna, C., Vladimir, V.: Support-Vector Networks, vol. 20, pp. 273–297. Kluwer Academic Publishers (1995)
5. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
6. Huang, G., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **42**(2), 513–529 (2012)
7. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1), 489–501 (2006). *Neural Networks*
8. Mathew, J., Pang, C.K., Luo, M., Leong, W.H.: Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(9), 4065–4076 (2018)
9. Raghuvanshi, B.S., Shukla, S.: Class-specific extreme learning machine for handling binary class imbalance problem. *Neural Netw.* **105**, 206–217 (2018)
10. Raghuvanshi, B.S., Shukla, S.: Class-specific kernelized extreme learning machine for binary class imbalance learning. *Appl. Soft Comput.* **73**, 1026–1038 (2018)
11. Schaefer, G., Nakashima, T.: Strategies for addressing class imbalance in ensemble classification of thermography breast cancer features. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 2362–2367, May 2015
12. Xiao, W., Zhang, J., Li, Y., Zhang, S., Yang, W.: Class-specific cost regulation extreme learning machine for imbalanced classification. *Neurocomputing* **261**, 70–82 (2017). *Advances in Extreme Learning Machines (ELM 2015)*
13. Yu, H., Ni, J.: An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **11**(4), 657–666 (2014)
14. Zeng, Y., Xu, X., Shen, D., Fang, Y., Xiao, Z.: Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Trans. Intell. Transp. Syst.* **18**(6), 1647–1653 (2017)
15. Zong, W., Huang, G.B., Chen, Y.: Weighted extreme learning machine for imbalance learning. *Neurocomputing* **101**, 229–242 (2013)



Bluetooth 5 and Docker Container: Together We Can Move a Step Forward Towards IOT

Mohammad Equebal Hussain¹(✉) and Rashid Hussain²

¹ Computer Science and Engineering, Suresh Gyan Vihar University, Mahal, Jagatpura, Jaipur 302025, Rajasthan, India

mdequebal.60508@mygyanvihar.com

² Department of Computer Science and Engineering, Suresh Gyan Vihar University, Mahal, Jagatpura, Jaipur 302025, Rajasthan, India

rashid.hussain@mygyanvihar.com

Abstract. Virtual wireless sensor network (WSN) is an approach towards automated monitoring system with minimum cost and maximum throughput. vWSN is an approach to convert physical WSN [1] into virtual device using VM and docker container combined with Bluetooth 5, this designed can be further transformed into on demand IOT device that is efficient, secure and real time device used for monitoring system. In this manuscript we will propose the end to end design to achieve the desired result having lot of advantages. This model can be deployed on the cloud. We will also discuss the security aspect of vWSN.

Keywords: Internet of Things (IoT) · Bluetooth 5 · Docker container · Security · Virtual wireless sensor network (vWSN) · Virtual machine (VM)

1 Introduction

Recent transformation from physical server to virtual machine to docker container. The former has many advantages over its predecessor. Standalone server was expensive, running at low potentials therefore wastage of resource, Later the virtual machine (VM) came into picture which allows multiple applications to run concurrently on single server. Going forward VM's are also not perfect because every VM requires dedicated OS. Every OS consumes memory, storage, CPU as well as license, continuous upgrade and monitoring. This model has other challenges like slow boot, portability to cloud etc. Docker container share single OS therefore it is fast, portable, elastic and cost effective. Docker enables separating application from infrastructure which helps shipping, testing and deploying much quicker and faster. Bluetooth 5 with docker can be used to design an efficient, portable, low power consumable device called virtual WSN.

2 Background

Docker container is not a replacement of VM but analogous to it. VM and container are not mutually exclusive. Both can exist together. The major difference is that container

doesn't need full OS. Containers are based on linux kernel namespace, control groups (cgroup) and Docker. Major components of docker are server (daemon), REST API and CLI (client). Docker registry is used to store docker images. Running instance of docker image is called container.

Bluetooth is a technology for short distance wireless communication. It doesn't depend on IP or internet. Data exchange is done instantly between any Bluetooth enable devices [2] over a short distance ranging from 10 to 100 m. Bluetooth 5 is latest version released in 2016 after V4.x, V5.0 has many improvements like performance and functionality, privacy and security, less power consumption (BLE – Bluetooth Low Energy), double speed (48 Mbit/s) and 4 times distance range (300 m) as well as support for IoT.

IoT (Internet of Things) refers to scenario [8] where connectivity and computing extends to sensors and other items normally not considered as computer, allowing generating, exchange and consuming data with minimal or no human intervention. Each object is provided a unique id (UID), IP address in order to transfer data over internet. IoT devices share collected data to an IoT gateway in order to send it to cloud or local server for further processing. Addressing new and unique security challenge [15] is also a priority.

3 Related Work

Many research works has been proposed in the area of WSN but few of them discuss about end to end design using virtual WSN. Sensor nodes collect data and transfer it to other node using internet or mobile network. The critical aspect is the network lifetime due to power consumption either by battery or any other source [5]. In this paper we propose an efficient method by separating control plane and data plane using virtual device. Theoretical analysis is presented in this approach. The authors in [6] proposed wireless sensor network coverage problem. Authors in [7] proposed energy efficient data routing communication protocol to maximize the energy and lifetime of sensor nodes.

4 Problem Formulation

Virtual wireless sensor network (vWSN) is fairly new concept taking the advantage of Bluetooth and dockernization to propose an IoT device. The design proposed in this manuscript is not just specific to any single application but equally applicable for monitoring and recording physical conditions of environment, weather, temperature, sound, pollutions, insects monitoring [3, 18], health monitoring [4], cyber security etc.

In this paper we propose a method for virtual WSN based on Bluetooth for communication, docker container for data processing using Internet of Things (IoT) technique [14].

We will also discuss about Bluetooth-5, major protocol change from its predecessor version 4.x and its advantages, about IoT and Docker container.

4.1 About Bluetooth 5

Bluetooth 5 is a short-range wireless communication technology for data transmission up to 2 Mbps using radio waves between two or more devices. It is developed specially to support IoT use case scenario [9]. Any file type such as photo, videos, documents, music or real time data can be transmitted over a short distance ranging from 10 to 100 m. Bluetooth technology is expanding to support IoT and machine to machine (M2M) communication [10] using Bluetooth Low Energy (BLE) which consumes less power [8]. This technology is used in many IoT applications like smart home automation and a traffic monitoring system.

Any Bluetooth device consist either an inbuilt adaptor or external card connected to device. Communications between devices are done using pre-defined embedded instructions. When devices come in each other's radio range, the Link Management Protocol (LMP) performs link setup, negotiation of packet size, segmentation and reassembling. The Service Delivery Protocol (SDP) enables devices to join network. Each device has a unique global ID which is exchanged between the devices to setup the connection. Bluetooth 5 uses mesh topology to communicate between devices within the network (Fig. 1).

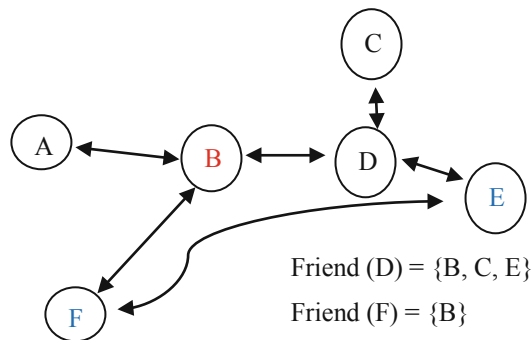


Fig. 1. Mesh network [11]

Suppose Node-F intend to send message to E, then F will send message (destination: E) to its entire neighbor i.e. node-B. Since B is not the destination hence it will re-broadcast the message to its entire neighboring i.e. node-A and node-D. A will discard the message, but node-D will forward the message to Node-E because D is a friend node of E.

4.2 Application of Bluetooth 5

Following are the major application of Bluetooth technology.

- a. peripheral devices keyboard, mouse, printer
- b. Healthcare and medical, motion detection
- c. Internet of Things (IoT) wireless sensors etc.

Advantages:

- a. Multiple devices can communicate to each other.
- b. Location based service
- c. More data transfer in short duration due to high speed
- d. Bluetooth 5 uses FEC (Forward Error Correction) technique to recover the data by receiver in case an error occurs during transmission [12].

5 Proposed System Architecture and Design

Major components:

- (i) Docker Engine – it consists of pre-configured pool of agents. Each agent represents one IoT device. It handles connection of device with agent. Agent talks to device collect data and pass it to other component for further processing [13].
- (ii) Record Parser -This is required to parse various fields of records generated by the device in order to store in database.
- (iii) In memory database. Redis will be used.
- (iv) HDFS uploader - its role is to upload parsed file to HDFS database.
- (v) Socket connection – TCP//IP wherever required.

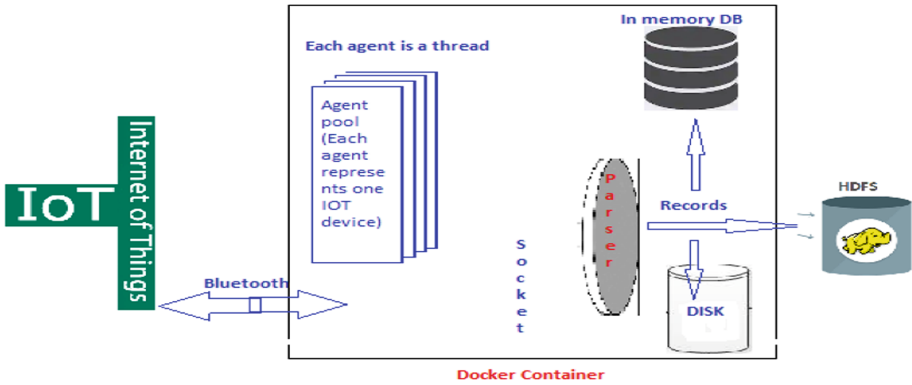


Fig. 2. Different components of proposed design

Docker container is the core backend component which collects data from IoT devices. It handles.

- Registration and De-registration.
- Filtering and distribution of work for parsing.
- Store records into in memory database.
- Define thread (Management thread, worker thread, Filter thread)

- Use RADIUS for authorization and authentication purpose wherever needed in order to assure security. RADIUS server is usually run as daemon process and communicates over UDP protocol.
- RADIUS server supports a variety of methods to authenticate a user.
- Response from server is Accept, Reject or Challenge.

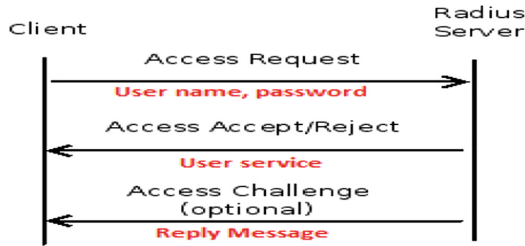


Fig. 3. Basic Authentication mechanism using RADIUS protocol.

5.1 vWSN as Virtual Machine vs. VWSN as Docker Container

Each VM has its own operating system whereas docker container focuses on deployment of application inside container, which can be packed and run as a single service. Single docker layer is shared among all containers within the same host. Authors in [16] proposed security implementation for this model (Fig. 4).

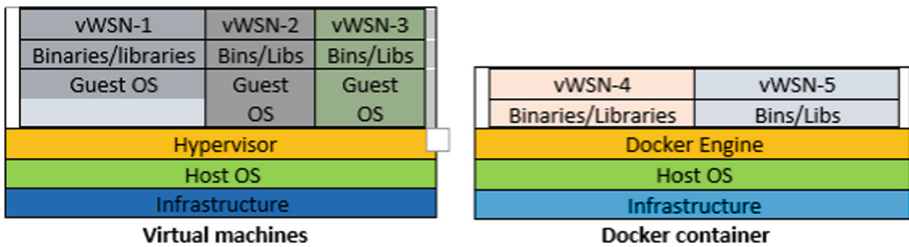


Fig. 4. VM vs. Docker container. <https://www.taksatech.com/containers-vs-vm/>

5.2 Docker Software Architecture

Docker underlying technology use Linux kernel to deliver its functionality Docker is based on namespace to provide isolation layers among container running in same host. Docker engine also relies on cgroups. This helps allocate resource like memory, CPU to a specific container (Fig. 5).

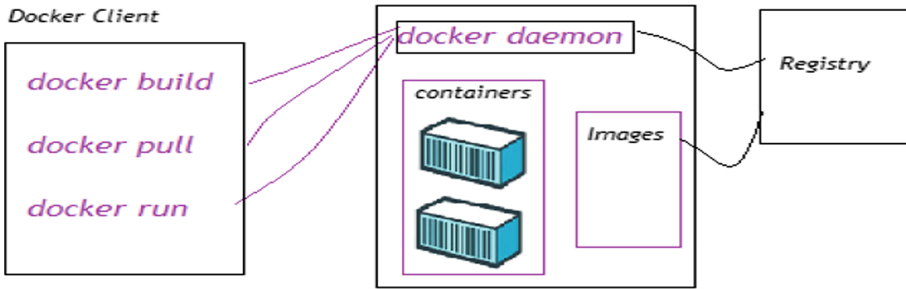


Fig. 5. Docker architecture. <https://www.taksatech.com/containers-vs-vms/>

Docker client is used to interact with docker daemon using various commands like docker build, docker run etc.

Docker daemon manages docker objects like containers, images, network and storage. It also listens to client request as shown in the above diagram.

Docker registry stores images. Docker pull, docker run and docker push commands are used to pull and push the images from docker hub registry.

Docker image is read only file consists of set of instructions to create a container. Dockerfile is used to create own docker image. Each instruction in Dockerfile creates a layer in image.

Docker is developed on the concept of namespace. Resources are visible only to those processes which run on the same namespace. Process running in one namespace can't see process running on another namespace. This provides a kind of virtualization and isolation of resource. Each container runs in its own namespace. This support is provided by the kernel which knows about the namespace and during API call; kernel makes sure that process can only access resource of its own namespace [13].

Docker hub is on the cloud registry service used to host various docker images built by other communities. We can also upload our own docker built images to docker hub.

Docker compose is used to define application using multiple container. Docker engine is used for building docker images.

Download and run docker image using below command.

```
$> docker run -it <image> /bin/bash
```

Displaying docker images.

```
$> docker images
```

```
REPOSITORY TAG IMAGE ID CREATED SIZE
virtual-wsn mhussain-1.0 b85ae2d81253 2 s 40 MB
```

To remove docker image.

```
$> docker rmi < image ID>
```


5.3 Building vWSN Image from Dockerfile

Dockerfile is a text file having series of instruction to build image. Some of the important commands are FROM, RUN, COPY, ADD, CMD, ENTRYPOINT, WORKDIR, USER, VOLUME etc. Start/stop/create/delete/move is some of the CLI command.

Dockerfile

```
FROM ubuntu: 1.0.18b1
MAINTAINER Mohammad Equebal Hussain
RUN apt-get update \
    && apt-get install -y vwsn \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/* \
    && echo "daemon off;" >> /etc/vwsn/wsn.conf
ADD default /etc/vwsn/sites-available/default EXPOSE 8080
CMD ["vwsn"]
```

Docker Build

```
docker build -t appliance/vwsn:1.0 -t appliance/vwsn: latest
```

Docker Push

```
Docker push appliance/vwsn: latest.
```

6 vWSN as Internet of Things (IOT)

IoT platform [17] helps to connect hardware, handle different communication protocols, Collect, visualize, and analyze data, provide security and authentication for users and devices as well as Integrate with other web services. IoT is a broad field and includes variety of applications. Many sensors/devices will collect data, Bluetooth as a connectivity technology that enables those sensors and devices to pass data to the backend (either docker application running locally or on the cloud). Activities like storage and data processing take place separately rather than on device, because large amount of computational power is extremely expensive and energy intensive therefore data from all the sensors and devices can be passed to the backend docker application running on the cloud is important for aggregating, processing and make intelligent decision. This gives high scalability. One of the biggest advantages of IoT platform is time to market.

One of the crucial parts is overall user experience which could be achieved using alerts and notification. When user get alert, how to get alert and what happens if alert is received but user doesn't react within the time frame.

There are some concerns need to be addressed while using cloud service. For example, Data ownership (actual owner vs. cloud provider), Latency (in IoT application, time is critical for taking decision) and potential crash (if connection broken then IoT application won't work).

6.1 IOT Elements

IOT provides many facilities and benefits to the user. In order to deliver functionality of IOT following elements are required [9].

Identify \Rightarrow Sense \Rightarrow Communication \Rightarrow Compute Services \Rightarrow Semantics

Identification refers to name and unique address (Ipv6).

Sensing refers to collecting information using sensing device such as RFID, smart sensor, actuators, wearable devices etc.

Communication refers to send and receive messages, files and other information using Bluetooth [10], Wi-Fi [8] and Long-Term Evolution (LTE) [11].

Computation refers to removing unnecessary information and keep useful information for further processing.

Services refers to making appropriate decision from collected and processed information on time every time.

Semantics is one of the most important elements of IoT which refers to getting all information and making appropriate decision to send response back to the device.

IOT is a 3-layer architecture consisting of perception or sensor layer (Lowest), Network or transmission Layer (Middle) and Application layer (Top).

6.2 Security Issue in Decernized IoT Solution

RFID: Radio Frequency Identification has three parts. Tags, reader and database. Encrypted information is transmitted between devices either using Data encryption (DES -symmetric key, 56 bit) or Advance encryption (AES-128 bit).

Reading information from tags is not secure because it gives information without authenticating the readers.

vWSN: Wireless Sensor Network consists sensors, microcontrollers and virtual memory. vWSN architecture consists physical layer, link, network, transport and application layer. Most common attacks on WSN is Denial of Service (DOS), man in the middle and DDOS attack [12].

Without proper security mechanism, IoT application is not useful. Existing securities are hash based encryption, Identity management framework, Intrusion detection system, Access control mechanism (ACL). Each device in IoT network need unique ID in order to communicate to each other. Therefore, support of such large number of devices is a big challenge. Other challenge is the huge data getting generated by these IoT devices over a period. In order to communicate and connect, dependency of device on other IoT device is also a challenge. There are many reasons to cause securities issues.

To overcome security issues, authentication (To verify user's identity using credentials) and ACL are used.

7 Conclusion

Internet of Things (IoT) is an emerging idea to quickly find the path to enhance the quality of modern life by connecting various smart devices. IoT allows to automate everything around us to make life easier, better and secure. This paper presents proposed design of vWSN as an IoT device using Bluetooth connectivity, Docker containers, possible security threat and their solution along with, their feature and limitations. This work can be further extended to propose secure docker containers to solve major security issues like malicious images in docker registry hub, denial of service attacks using access control list (ACL) and data loss prevention (DLP) solution.

Acknowledgements. I would like to thank Dr. Rashid Hussain for his encouragement and support during the research.

References

1. Venkatasubramanian, K., Deng, G., Mukherjee, T.: Ayushman: a wireless sensor network based health monitoring infrastructure and testbed distributed computing in sensor systems. In: Prasanna, V.K., Iyengar, S.S., Spirakis, P.G., Welsh, M. (eds.) DCSS 2005. Lecture Notes in Computer Science, vol. 3560, pp. 406–407. Springer, Heidelberg (2005). https://doi.org/10.1007/11502593_39
2. Ng, H.S., Sim, M.L., Tan, C.M., Wong, C.C.: Wireless technologies for telemedicine. *BT Technol. J.* **24**(2), 130–137 (2006)
3. Soni, K., Hussain, R.: WSN based insect monitoring: pest detection through MATLAB-artificial neural network, vol. 5, no. 1. ISSN 2278-1315
4. Yuce, M.R., Ng, P.C., Khan, J.Y.: Monitoring of physiological parameters from multiple patients using wireless sensor network. *J. Med. Syst.* **32**, 433–441 (2008). <https://doi.org/10.1007/s10916-008-9149-5>
5. Cardei, M., Du, D.: Improving wireless sensor network lifetime through power aware organization. *Wirel. Netw.* **11**, 333–340 (2005). <https://doi.org/10.1007/s11276-005-6615-6>
6. Huang, C., Tseng, Y.: The coverage problem in a wireless sensor network. *Mob. Netw. Appl.* **10**, 519–528 (2005). <https://doi.org/10.1007/s11036-005-1564-y>
7. Kumar, R., Kumar, D.: Multi-objective fractional artificial bee colony algorithm to energy aware routing protocol in wireless sensor network. *Wirel. Netw.* **22**, 1461–1474 (2016). <https://doi.org/10.1007/s11276-015-1039-4>
8. Abbas, Z., Yoon, W.: A survey on energy conserving mechanisms for the internet of things: wireless networking aspects. *Sensors* **15**(10), 24818 (2015)
9. Chang, K.H.: Bluetooth: a viable solution for iot? [industry perspectives]. *IEEE Wirel. Commun.* **21**(6), 6–7 (2014)
10. Gazis, V.: A survey of standards for machine to machine (m2m) and the internet of things (IoT). *IEEE Commun. Surv. Tutor.* **PP**(99), 1 (2016)
11. <https://electricalfundablog.com/wp-content/uploads/2019/01/Bluetooth-topology.png>
12. Nafaa, A., Taleb, T., Murphy, L.: Forward error correction strategies for media streaming over wireless networks. *IEEE Commun. Mag.* **46**(1), 72–79 (2008)
13. Martin, J.P., Kandasamy, A., Chandrasekaran, K.: Exploring the support for high performance applications in the container runtime environment. *Hum.-Cent. Comput. Inf. Sci.* **8**, Article number: 1 (2018)

14. Ruchika, V.: Evaluation of Docker for IoT application. *Int. J. Recent Innov. Trends Comput. Commun.* **4**(6) 624–628 (2016)
15. Singh, V., Pandey, S.K.: Cloud computing: vulnerability and threat indications. In: Pant, M., Sharma, T.K., Basterrech, S., Banerjee, C. (eds.) *Performance Management of Integrated Systems and its Applications in Software Engineering*. AA, pp. 11–20. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-8253-6_2
16. Mukherjee, T., Sahana, S., Sarddar, D.: A noble approach toward security implementation in cloud virtualization framework. In: Kundu, S., Acharya, U.S., De, C.K., Mukherjee, S. (eds.) *Proceedings of the 2nd International Conference on Communication, Devices and Computing*. LNEE, vol. 602, pp. 541–552. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-0829-5_52
17. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* (2013). <https://doi.org/10.1016/j.future.2013.01.010>
18. Jha, T., Hussain, R.: WSN controlled insects monitoring: identification of onion thrips. *IJCAR* **6**, 5257–5260 (2014)



Statistical Analysis Based Feature Selection for Detection of Breast Cancer Using Thermograms

Vartika Mishra^(✉), Monika Lilhare, and Santanu Kumar Rath

National Institute of Technology, Rourkela 769008, Odisha, India
517CS1015@nitrkl.ac.in

Abstract. Breast Cancer is one of the most prevalent diseases among women. Its early diagnosis helps to increase the survival rate. Among many modalities, thermography is considered to be an early diagnostic procedure, which depicts the temperature values of the hot regions and further provides scopes in locating the tumor. In this work, features from Gray Level Co-occurrence Matrix (GLCM) and Gray Level Run Length Matrix (GLRLM) are extracted using the breast thermograms. Dimensionality reduction technique i.e., Autoencoder is applied to the extracted features. It gives the non-linear pixel intensities of the breast thermograms. Further, the reduced feature set is directed towards the statistical analysis of the features with three different methods viz. Filter, Wrapper and Embedded methods on the breast thermograms for the selection of best features set. Random Forest and Decision tree based classification algorithms are further applied for the features selected using three statistical tests. Among both the classifiers, Random forest with Recursive feature Elimination method gives a better performance in detecting the tumors between healthy and unhealthy breasts, giving an accuracy level of 81.63%.

Keywords: Breast cancer · Thermography · Feature reduction · Statistical analysis · Feature selection · Classification

1 Introduction

Breast cancer is a disease where the infected cell grows uncontrollably. It is mainly developed in the mammary glands of the breast forming tumor, which is invasive in nature [1]. It is a common cancer among the women, where in average, every third women is diagnosed worldwide. The survival rate can be increased if diagnosed early.

Various modalities are being followed for the detection of breast cancer like mammography, ultrasound, X-ray and thermography etc. Among all, thermography helps to increase the rate of survival as it detects the tumor at an early stage when compared with other modalities [2]. Unlike other modalities, the procedure is contact free, non-ionizing non-invasive in nature. It doesn't make the patient very uncomfortable while examination is conducted, as compared to other modalities used.

In thermography, the temperature of the surface is recorded and an image is captured in an infrared thermal camera which is further analyzed to locate the hot spot [3]. The

tumor cells have been observed to be higher in temperature, thus it makes easy through thermography to indicate a hot spot. The thermal camera displays the temperature ranging from hot to cold regions in an image. No radiations are passed towards the patient while examining the suspected area. In this work, the texture features GLCM and GLRLM texture features comprising of 27 features are extracted from the thermogram images. Then further dimensionality reduction technique i.e., Autoencoder is applied, through which eight best features are obtained. These eight features are statistically examined by filter, wrapper and embedded methods for selecting the best features critically. Pearson correlation, Recursive Feature Elimination and Lasso methods are further applied on them through which the best of five features are obtained from each method. Further, these five features from each three methods, are applied individually to the classification methodologies such as random forest and decision tree. Recursive feature elimination method when classified with random forest gives the best performance among all the other methods.

This paper is organized as follows: Sect. 2 gives the state of the art, Sect. 3 presents the details of dataset and the method applied, Sect. 4 gives result analysis and Sect. 5 concludes the entire work.

2 State of the Art

2.1 Literature Survey

For detection of cancerous tumors, it has been observed that the thermography is more helpful unlike other modalities like mammography and ultrasonography based on the research outcome over the past few years [4, 5].

Acharya et al. [6] have applied feed-forward artificial neural network (ANN) and SVM for classifying the breast thermogram images by extracting the spectral features of higher order with radon transform. When compared with SVM, the higher order spectral features give better performance w.r.t specificity, sensitivity and accuracy with ANN. Borchardt et al. [7] have applied temperature based features such as standard deviation, mean, range of temperature in the region and quantization of the higher tone extracted from the breast quadrants and the entire breast image. They gained an accuracy of 86% with SVM (Support Vector Machine) classification model.

Nicandro et al. [8] have considered features based on variations in temperature such as heat area under the breast, number of veins with higher temperature, temperature difference, geometry of the hot center, number of hottest points, histogram and age of the patient. Bayesian network is applied as the classification algorithm obtaining an accuracy of 77%. Pramanik et al. [9] have used the pseudo color based thermogram images from which they have extracted the statistical features by applying wavelet transform and used ANN for classification.

Sheeja V. et al. [10] have applied Curvelet based features representation for automated classification. Here statistical and Haralick features are applied to the SVM classifier obtaining an accuracy of 83.36% and 90.92% respectively. M.Etehad Tavakol et al. [11] analyzed and showed that the Highest Order Statistics (HOS) features were very efficient in differentiating among the healthy and unhealthy classes. The dataset comprised of 11 normal, 9 malignant and 12 benign cases. H Qi et al. [12] have distributed

the different intensities based on quantization in the thermograms by extracting higher-order statistics features elements. From the obtained set of features, asymmetry among the breast was analysed.

D. Sathish et al. [13] have applied polynomial curve fitting using shape features for segmenting the breasts. They have extracted histogram and GLCM based texture features from the segmented images and obtained an accuracy of 90% using SVM RBF classifier. Deepika S. et al. [14] have presented a survey on the breast cancer detection systems including image acquisition, feature extraction, classification methods and use of thermography in the past few decades. This survey helps in the knowledge for reliability of the breast cancer detection using thermography. Madhu et al. [15] have extracted non-vascular thermal features based on medical interpretability. They have considered features which differ from non-malignancy conditions including hormone sensitivity and certain benign conditions. This work is applied for 78 subjects. U. Raghavendra et al. [16] have proposed a novel method by combining the HOG and KLPP for breast cancer detection using thermography. It is observed that the combination of two method enables to extract the hidden signatures from the breast thermograms. Among many classifiers applied, Decision Tree gives the promising result with sensitivity of 96.66%. V Mishra and S. K. Rath [17] have selected the features using the unsupervised reduction technique i.e., Autoencoders and PCA for breast thermograms. They have applied different classifiers among which, Random Forest with PCA gives an accuracy of 95.45% in classifying between healthy and unhealthy breast.

2.2 Research Gap

The latest research has been done in recent years with the following mentioned literatures. V Madhavi and C.B. Thomas [18] they have extracted the texture features in identifying the presence of tumour. Multi-view thermograms are used for obtaining the significant texture features by applying Kernel principal component analysis (KPCA). Here, statistical based analysis is performed by applying t-test method. Gogoi U. et al. [19] have applied Mann–Whitney–Wilcoxon statistical test for selecting the best features, being extracted from the breast thermograms of DBT-TUJU and DMR dataset. Bonini et al. [20] have applied different statistical measures as features viz. Histogram, Higuchi function and methods of geostatistics: Geary coefficient, Moran Index and Ripley's K Function followed by applying Support Vector Machine for classification. Jeyanathan et al. [21] have categorized features in three major transforms namely wavelet, curvelet and contourlet transform. By applying the statistical test ANOVA and independent t test, the thermograms are analyzed based on statistical values.

It is observed that various researchers have applied feature extraction techniques from images; but the most important features are not obtained as the entire set gives redundant information and there is a probability of increase in the complexity of the algorithms. Hence, to obtain the best set of features and increase in the accuracy levels, this study is focused on feature selection based on statistical analysis. In this study, firstly the dimensionality reduction technique is applied by Autoencoders. Further these features are selected based on statistical test by applying filter, wrapper and embedded techniques. Each of the tests selects five features individually from the set of eight features reduced by Autoencoders.

2.3 Dataset Used

The dataset used in this work is obtained from the online sources comprising data of 56 subjects (640×480 pixels); each having 20 positional temperature matrix. The dataset belongs to the Database of Mastology Research (DMR) repository of Fluminense Federal University (UFF) Brazil [22]. The thermal Camera FLIR SC-620 with a spatial resolution of 640×480 , has been used for capturing the thermograms [23]. Among a total of 56 subjects, there are 19 healthy subjects and 37 unhealthy subjects. The thermograms are captured by maintaining a room temperature between 20°C and 22°C keeping windows closed and no flow of air towards the patient [23]. It is mentioned that there is one metre distance between the patient and the camera.

3 Methodology Approached

This work has been carried by analysis in this proposed study with the following steps: Firstly, pre- processing of the thermograms is done, then segmentation of the breast area is carried out, further by extracting the features from the breast thermograms, then selection of most relevant features using dimensionality reduction technique and statistical analysis and then classifying for the presence of tumor between healthy and unhealthy breast (see Fig. 1).

3.1 Preprocessing

The temperature matrix is pre-processed for obtaining the breast thermograms. The images are transformed to the pseudo colour images (see in Fig. 2a and 2b) respectively. Further the images are converted to the gray scale image as it requires less information to be given to the pixels.

3.2 Segmentation

In segmentation process, the region of interest of the breast is segmented by selecting the left coordinate and right coordinates of the breasts. Otsu's Thresholding is applied in the images as it utilizes the inner-class variance and the between variance to measure the two areas various levels of grey value [24].

3.3 Feature Extraction

In this work, the features are extracted from the breast thermograms based on GLCM and GLRLM matrices. The GLCM matrix [25] considers the spatial relationship of the pixel values in gray levels which statistically examines the texture of the features. It calculates the occurring of specific pixel values with respective spatial relationship extracting the different statistical measures from the matrix. Here, we have extracted 20 features from the GLCM matrix. The GLRLM matrix [26] looks for the pair of pixel of a particular grey color in a sequential direction which develops a 2D matrix. Here we have obtained seven features from the GLRLM matrix.

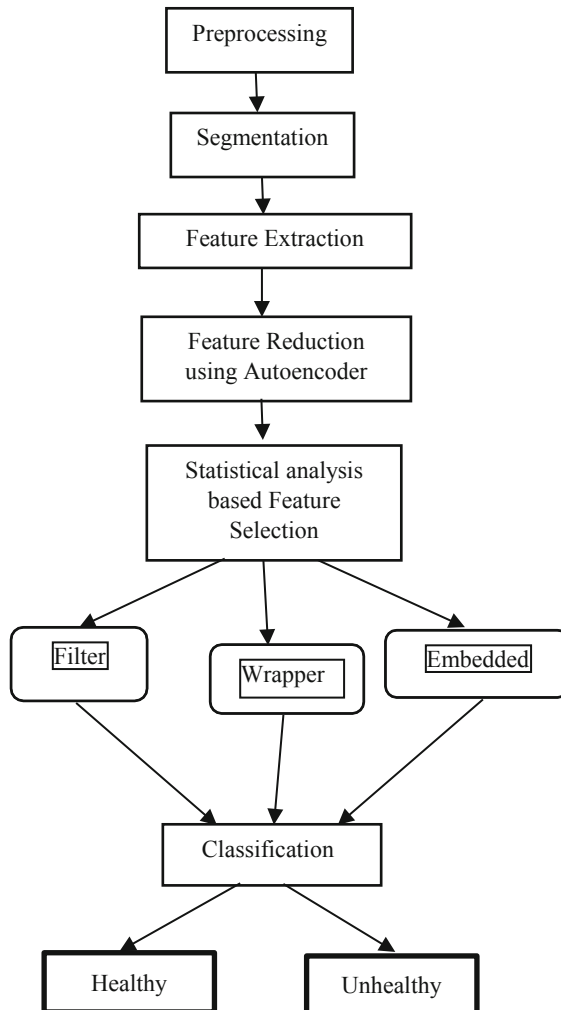


Fig. 1. Flow chart of the process

3.4 Feature Reduction

Feature extraction accounts for large amount of data from an image, which directs towards the selection and reduction features often needed [27]. The redundant features may hamper the results and thus may mis-guide towards ill-direction. Thus to limit this, dimensionality reduction techniques are applied. Here in this work autoencoder technique is applied for the dimensionality reduction purpose.

- Autoencoders.

Artificial neural network has different variations, one of its type is autoencoder which is used to learn efficient data coding in an unsupervised manner [28]. In this paper autoencoder is used to reduce the dimensionality of features of given dataset. Data in lower dimension can improve performance of the further tasks like classification. The architecture of autoencoder contains two parts: encoder and decoder. Encoder part has one input layer and hidden layers. It is used to learn and describe latent attributes of the input data. Decoder part has hidden layers and output layer. It is used to update the information provided by encoder iteratively during training through Back-propagation.

3.5 Statistical Analysis-Based Feature Selection

Statistical analysis is performed to obtain a summary of the data which has been collected, and also provide understanding of the various processes involved in generation of the stored data [27]. It is based upon the techniques that generate the subset of the features and those selected subset features are evaluated. In this work, statistical analysis-based feature selection methods are applied to get a better set of features.

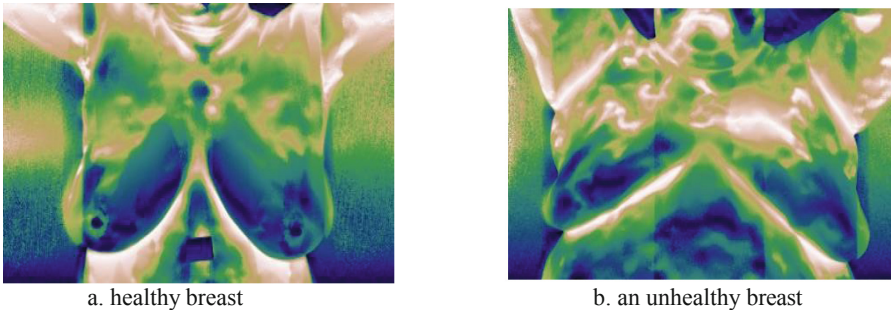


Fig. 2. The breast thermograms

1. Filter Method For Feature Selection

Filter methods find out as to how much the features are correlated or relevant to each other. It is not incorporated with any machine learning algorithm [29]. It ranks the features based upon the descriptive statistical measure and then gives the highest ranking subset of the features. The methods are often univariate and consider the

features independently, or with regard to the dependent variable. Advantages of using filter methods are that they take less computational time as compared to other two methods i.e., wrapper and embedded methods. Also it has a very less chance of being prone to overfitting. In this paper, Pearson's correlation have been used for selecting the features.

- **Pearson's correlation**

Statistical measures are used to find the rank of the feature. It finds the relation between independent and dependent features [30].

2. Wrapper Method For Feature Selection

Wrapper method is a greedy search approach by finding all possible combination of features on the basis of evaluation criterion [29]. A downside of this method is that it is highly computationally expensive while trying all the combinations of the features and overfitting problem might occur because of training model in algorithm with various combination of features. This method evaluates on particular machine learning algorithms to find the optimal features. In this paper, Recursive Feature Elimination is used for selecting the features.

- **Recursive Feature Elimination (RFE)**

It is the backward selection of the predictor that recursively eliminates the features until the specified number of features reached. It calculates the score for building the model on each predictor and according to the score, value of the features are getting eliminated [31].

3. Embedded Method For Feature Selection

It measures the feature subset usefulness by combining both the above methods i.e. Filter method and Wrapper Method [29]. It fixes the features during the model building process. This method helps to reduce overfitting by penalizing the coefficients of a model being too large. Regularization method is the most commonly used embedded method. In this paper, Lasso method is used for selecting the features.

- **LASSO**

LASSO implies absolute shrinkage and selection operator. It is a supervised technique. LASSO is basically a shrinkage and selection method which minimizes the sum of squared errors. It will be penalizing the coefficients of the linear regression [32].

3.6 Classification Models Used

The classification algorithms help in developing are the predictive models in order to determine the class of unknown sample/subject. The main rule behind the classification of algorithms is the analysis of the features that helps in differentiating the individual classes from the training dataset.

In this work, two classification models viz. Random Forest and Decision tree are used for classification among the unhealthy and healthy breasts.

1. Decision Tree

Decision tree is used for building the classification model in the form of tree structure. Decision tree is having the flowchart like tree structure [33]. The decision tree algorithm breaks down the dataset into smaller and smaller subset, while at the same time it is incrementally being developed. The instances are classified by the starting at the root node and the final result is obtained in the last node that is leaf node (see Fig. 3). The node in decision tree represent the test for the value of the attribute and branch represent the result of the node and its connectivity to the next or leaf node. The training of the model should neither be under-fitted nor over-fitted. In this work, depth of three levels is used with gini index.

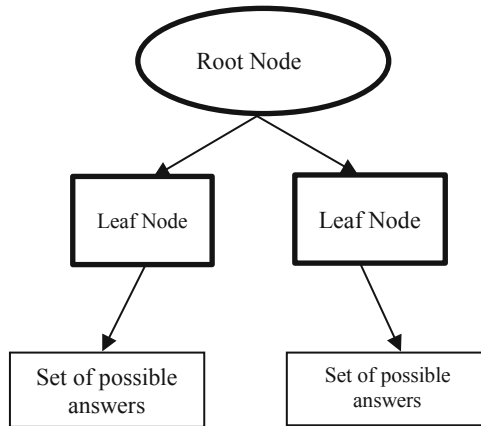


Fig. 3. A simple decision tree

2. Random Forest

Random forest method is a classification algorithm which is based on the set of unpruned decision trees [34]. Each tree classifies the data on the basis of a subset of attributes (see Fig. 4). Classifier's prediction is obtained considering as a majority of vote of the trees. The parameters of the algorithm include amount of trees and size of the subset of attributes each tree is given. The attributes correlation and error rate of individual trees helps in minimizing the error rate of the algorithm. Every subset of decision tree is applied for training the dataset. In this work, 100 trees are applied with a maximum depth of three levels.

4 Results Analysis

4.1 Performance Parameters

A representation technique known as confusion matrix helps to assess the performance of classification models. It depicts the samples classified as correctly and incorrectly

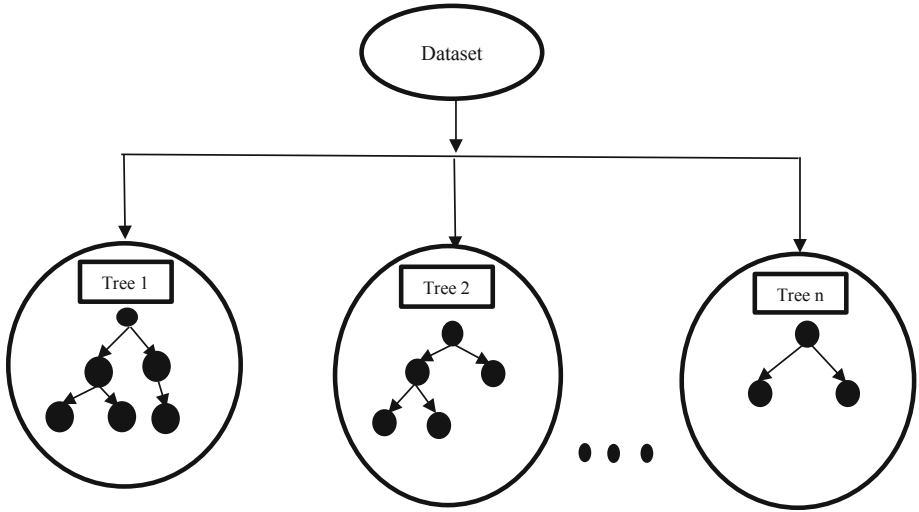


Fig. 4. Random forest example

w.r.t the test data for actual result. The four elements denoted as TP, FP, FN and FP are the element of the matrix which count the quantity of a binary classifier for all the four results. The values are calculated for the actual versus the predicted values (see Fig. 5). The different parameters such as accuracy, specificity, sensitivity, precision, F1-score and MCC (Mathews Correlation Coefficient) are calculated by these four elements to measure the performance of classification models [26].

		Actual	
		No	Yes
Predicted	No	True Negative	False Positive
	Yes	False Negative	True Positive

Fig. 5. Confusion matrix

True positive (TP): Number of malignant tumors predicted as unhealthy

False positive (FP): Number of benign tumors predicted as unhealthy

True negative (TN): Number of benign tumors predicted as healthy

False negative (FN): Number of malignant tumors predicted as healthy.

4.2 Analysis

In this work, dimensionality reduction technique is applied using autoencoder followed by statistical analysis based feature selection methods for obtaining the best subset of features.

While applying dimensionality reduction technique, the autoencoder with 4 layers is being considered. The non-linear pixel intensities are obtained for the breast thermograms. In this work, by applying autoencoders, a reduced feature set from twenty seven features to eight features that are best amongst the rest is obtained. These eight features are statistically analyzed using the filter, wrapper and embedded methods individually. Feature selection methods help in removing the unwanted and redundant data. Hence, they help to identify more of the important features. They help to further enhance the classification accuracy with minimum error rate. The subset selection of features helps in getting the important information from the dataset.

In filter method, Pearson's correlation method is applied which selects five features from the obtained eight set of features from the autoencoders. Hence, it removes the duplicate features from the set by determining the dependency of two variables. In Wrapper method, Recursive elimination method is applied for selecting the features. It removes the weakest features that are correlated by cross validating. It finds the optimal number of features from different feature subsets and hence, the best scoring features are selected. It selects five best features from the subset of eight features. In embedded method, Lasso method is applied to increase the interpretability of the features by eliminating the irrelevant features from the set for the response variable. It helps in increasing the model accuracy. Here, the Lasso method has selected the best five features from the subset of eight features. All the tests are performed individually.

After the selection of features by applying pearson's correlation, RFE and Lasso method, the Decision Tree classifier is applied as shown in Table 1 which gives the performance metrics for the three statistical test performed. Among the three methods applied, Recursive Feature Elimination method gives the better accuracy among the three tests performed. It has selected the best of five features. Other parameters such as specificity, sensitivity, precision, f1-score and MCC are also calculated.

Table 1. The performance parameters using Decision Tree classifier

	Accuracy	Sensitivity	Specificity	F1-score	MCC	Precision
Pearson's correlation	69.19	75.23	58.16	75.93	33.15	76.64
Recursive feature elimination	73.71	81.59	58.09	80.50	40.25	79.43
LASSO	73.45	79.29	61.95	79.91	40.93	80.54

The classification is further carried out using the Random Forest with the different performance metrics viz. sensitivity, specificity precision, F1-score and MCC as shown in Table 2 with the three tests performed. It is applied on the features selected by pearson's correlation, RFE and Lasso method individually. Among the three tests

performed, Recursive Feature Elimination method gives the best performance among all the techniques applied by selecting five features.

Table 2. The performance parameters using Random Forest classifiers

	Accuracy	Sensitivity	Specificity	F1-score	MCC	Precision
Pearson’s correlation	74.81	85.84	49.01	82.69	37.05	79.76
Recursive feature elimination	81.63	92.65	59.73	87.03	57.18	82.05
LASSO	79.41	89.26	60.09	85.17	52.33	81.43

The comparison for the different accuracies obtained using the three statistical tests viz. Pearson’s correlation, Recursive Feature Elimination and Lasso methods by obtaining the five features sub set are been displayed in the graph (see Fig. 6). Among them Random Forest classifier gives the better performance when compared with Decision Tree.

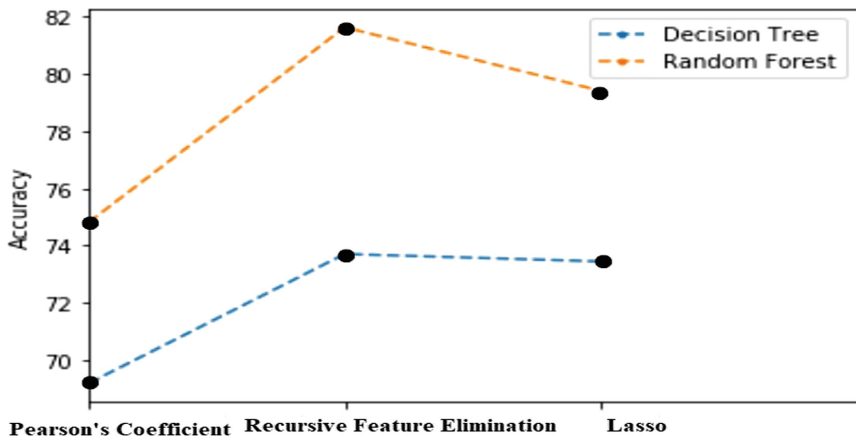


Fig. 6. The comparison of accuracies between Random Forest and Decision Tree

Application of feature selection along with feature classification algorithms based on thermography dataset is a novel application for detection of tumour in thick vascular breast instead of following the conventional mammography dataset.

5 Conclusion

Appropriate selection of features helps to boost the accuracy levels and efficiency of classifiers methods. In this study, dimensionality reduction technique has been applied in which autoencoder methodology is used. They reduce the feature set from twenty seven features to eight features. Further, the three statistical tests are applied viz. Pearson’s

correlation, Recursive Feature Elimination method and Lasso in order to obtain the best set of features which are used for classification among the healthy and unhealthy breast by applying decision tree and random forest. The individual tests select five best features from the subset of eight features selected by applying autoencoder.

Among all the tests performed for different classifiers, Random forest with Recursive feature elimination method yields the best accuracy, to the tune of 81.63% with a subset of five features. Wrapper based method is advantageous for giving better performances since they use the target classifier for the feature selection algorithm.

It is hereby proposed to extend the study by making feature analysis based on ranking of features which will interpret the most important attributes for the detection of breast thermograms.

References

1. Daisy, P.A., Nagappa, A.N., Udupa, N., Mathew, N., Carla, B.: Effectiveness of a planned teaching program on improving the knowledge on warning signs, risk factors and early detection methods. *Indian J. Pharm. Pharmacol.* **2**(1), 6–9 (2015)
2. Tan, T.Z., Quek, C., Ng, G.S., Ng, E.Y.K.: A novel cognitive interpretation of breast cancer thermography with complementary learning fuzzy neural memory structure. *Expert Syst. Appl.* **33**(3), 652–666 (2007)
3. Etehadtavakol, M., Ng, E.Y.K.: Breast thermography as a potential non-contact method in the early detection of cancer: a review. *J. Mech. Med. Biol.* **13**(2), 1330001 (2013)
4. Yao, X., et al.: A comparison of mammography, ultrasonography, and far-infrared thermography with pathological results in screening and early diagnosis of breast cancer. *Asian Biomed.* **8**(1), 11–19 (2014)
5. Neal, C.H., Flynt, K.A., Jeffries, D.O., Helvie, M.A.: Breast imaging outcomes following abnormal thermography. *Acad. Radiol.* **25**(3), 273–278 (2018)
6. Acharya, U.R., Ng, E.Y.K., Sree, S.V., Chua, C.K., Chattopadhyay, S.: Higher order spectra analysis of breast thermograms for the automated identification of breast cancer. *Expert Syst.* **31**(1), 37–47 (2014)
7. Borchardt, T.B., Resmini, R., Conci, A.: Thermal feature analysis to aid on breast disease diagnosis. In: *Proceedings of 21st Brazilian Congress of Mechanical Engineering Natal, RN, Brazil, COBEM, BCM*, pp. 24–28 (2011)
8. Nicandro, C.R., et al.: Evaluation of the diagnostic power of thermography in breast cancer using Bayesian network classifiers. *Comput. Math. Methods Med.* **2013**(5), 1–10 (2013)
9. Pramanik, S., Bhattacharjee, D., Nasipuri, M.: Wavelet based thermogram analysis for breast cancer detection. In: *International Symposium on Advanced Computing and Communication (ISACC) Silchar, India*, pp. 205–212 (2015)
10. Francis, S.V., Sasikala, M., Saranya, S.: Detection of breast abnormality from thermograms using curvelet transform based feature extraction. *J. Med. Syst.* **38**(4), 2–9 (2014)
11. EtehadTavakol, M., Chandran, V., Ng, E.Y.K., Kafieh, R.: Breast cancer detection from thermal images using bispectral invariant features. *Int. J. Therm. Sci.* **69**, 21–36 (2013)
12. Qi, H., Kuruganti, P.T., Snyder, W.E.: Detecting breast cancer from thermal infrared images by asymmetry analysis. In: *Nicholas, A., Diakides, M., Bronzino, J.D. (eds.) Medical Infrared Imaging Principles and Practice, The Biomedical Engineering Handbook, 3rd edn.*, pp. 11.1–11.13. Taylor and Francis (2007)
13. Sathish, D., Kamath, S., Prasad, K., Kadavigere, R., Martis, R.J.: Asymmetry analysis of breast thermograms using automated segmentation and texture features. *Signal Image Video Process.* **11**(4), 745–752 (2017)

14. Singh, D., Singh, A.K.: Role of image thermography in early breast cancer detection-past, present and future. *Comput. Methods Program. Biomed.* **183**, 105074 (2020)
15. Madhu, H., Kakileti, S.T., Venkataramani, K., Jabbireddy, S.: Extraction of medically interpretable features for classification of malignancy in breast thermography. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1062–1065. IEEE (2016)
16. Raghavendra, U., Acharya, U.R., Ng, E.Y.K., Tan, J.-H., Gudigar, A.: An integrated index for breast cancer identification using histogram of oriented gradient and kernel locality preserving projection features extracted from thermograms. *Quant. InfraRed Thermogr. J.* **13**(2), 195–209 (2016)
17. Mishra, V., Rath, S.K.: Detection of breast cancer tumours based on feature reduction and classification of thermograms. *Quant. InfraRed Thermogr. J.* 1–14 (2020). <https://doi.org/10.1080/17686733.2020.1768497>
18. Madhavi, V., Thomas, C.B.: Multi-view breast thermogram analysis by fusing texture features. *Quant. InfraRed Thermogr. J.* **16**(1), 111–128 (2019)
19. Gogoi, U.R., Bhowmik, M.K., Ghosh, A.K., Bhattacharjee, D., Majumdar, G.: Discriminative feature selection for breast abnormality detection and accurate classification of thermograms. In: 2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC), pp. 39–44. IEEE (2017)
20. Borchardt, T.: Thermographic image analysis for the change of classification in breast. Univerisadade Federal Fluminense, Brasil (2013)
21. Jeyanathan, J.S., Shenbagavalli, A., Venkatraman, B., et al.: Analysis of transform-based features on lateral view breast thermograms. *Circ. Syst. Signal Process.* **38**(12), 5734–5754 (2019)
22. URL for thermogram repository. <http://visual.ic.uff.br/en/proeng/thiagoelias/>
23. Silva, L.F., Sadde, D.C.M., Sequeiros, G.O., et al.: A new database for breast research with infrared image. *J. Med. Imaging Health Inform.* **4**(1), 92–100 (2014)
24. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
25. Mohanty, A.K., Beberta, S., Lenka, S.K.: Classifying benign and malignant mass using GLCM and GLRLM based texture features from mammogram. *Int. J. Eng. Res. Appl.* **1**(3), 687–693 (2011)
26. Tang, X.: Texture information in run-length matrices. *IEEE Trans Image Proces.* **7**(11), 1602–1609 (1998)
27. Raghavendra, U., Gudigar, A., Rao, T.N., Ciaccio, E.J., Ng, E.Y.K., Acharya, U.R.: Computer aided diagnosis for the identification of breast cancer using thermogram images: a comprehensive review. *Infrared Phys. Technol.* **102**, 103041 (2019)
28. Wang, Y., Yao, H., Zhao, S.: Auto-encoder based dimensionality reduction. *Neurocomputing.* **184**, 232–242 (2016)
29. Inza, I., Larranaga, P., et al.: Filter versus wrapper gene selection approaches in DNA microarray domains. *Artif. Intell. Med.* **31**(2), 91–103 (2004)
30. Blessie, E.C., Karthikeyan, E.: Sigmis: a feature selection algorithm using correlation based method. *J. Algorithms Comput. Technol.* **6**(3), 385–394 (2012)
31. Zeng, X., Chen, Y.-W., Tao, C., van Alphen, D.: Feature selection using recursive feature elimination for handwritten digit recognition. In: 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1205–1208. IEEE (2009)
32. Fonti, V., Belitser, E.: Feature selection using lasso. *VU Amsterdam Res. Pap. Bus. Anal.* **30**, 1–25 (2017)

33. Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., Khovanova, N.: Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. *Biomed. Signal Process. Control* **52**, 456–462 (2019)
34. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)



Distributed Computing Solution for Steganography Using Visual Cryptography and Genetic Algorithm

Harsh Maru^(✉), P. Pranav, Yashwanth Miryala, and Bhawana Rudra

National Institute of Technology Karnataka, Surathkal, India
bhawanarudra@nitk.edu.in

Abstract. Photo steganography is a rising research field for improvising secure data hiding and transmission over the network. The base concept of the proposed system which is Least Significant Bit (LSB) based Steganography along with Visual Cryptography using Genetic Algorithm is implemented over a network of distributed systems connected to each other via LAN. The original message is converted into cipher text and then hidden into the Least Significant Bit – LSB of the original image. Visual Cryptography and Genetic Algorithm are implemented for enhancing the security. Genetic Algorithm is basically used to modify the pixel location of steganography image and thus making the detection of the message complex. To encrypt the visual information, Visual Cryptography is used. To achieve it, we break the image into two shares based on a given threshold. The complete workload of the above proposed systems is divided among Distributed Systems with the help MPI, NFS and SSH and the work is done in parallel among the machines in the distributed network. The purpose of the proposed method is to improve the speed of execution of the enhanced secure algorithm to make the proposed system viable to larger data and a wider spectrum of applications.

Keywords: Distributed computing · Steganography · Secure data hiding · NFS · MPI

1 Introduction

Simply put, steganography is the embedding of secret data into an uncontroversial medium or any kind of unsusceptible medium. Steganography is basically applied by taking a message-a binary file and some kind of envelope or cover that could be an audio file (sound) or any image file, and combining both to get a “stego-object.” One of the most common steganalysis, the RS review, is capable of detecting hidden messages by statistically analyzing the pixel values. RS steganalysis algorithm uses the natural and singular classes as parameters for making the pixel correlation approximate. The main drawback is that using the conventional LSB replacing steganography, the method ends the alteration

in the proportion in singular and normal groups that indicates that steganography was performed on the image, as the picture now known contains some secret data. Thus it is highly useful and efficient to make the Steganography challenging to detect, thus making it in turn much more secure. For image security, steganography and visual cryptography are considered as different topics. Although there exists extensive and in-depth research based on combining the approaches as mentioned earlier: Steganography and Visual Cryptography. But unfortunately, the results found are not as satisfactory as proposed given the potential of the RS algorithm, which is stated above. It can be said that fundamentally, one could have a secret image which contains hidden data broken up into various encrypted shares. These encrypted shares, when are reassembled together for the decryption to redesign the original image, there is existence of a possibility that expose of the image which was shared which may still consists of classified data. Therefore such kinds of proposed systems can not persist without the ownership of an appropriate features in the visual cryptography procedure. The base for this is the fact that if the encoding method or even reassembly method changes the data that is present in the image, then the system may change the information encrypted, which makes the whole system susceptible to the extraction of the encrypted data from the compromised image. The process of exposure of the confidential message even certain uncertain media. There have been various attacks reported on LSB. However, although the combination of Steganography along with Visual Cryptography using Genetic Algorithm may make the image data hiding much more secure the fact that the whole process is time-consuming makes it an unattractive choice for transfer of confidential data over networks. But if the whole process of Steganography and Visual Cryptography along with Genetic Algorithm for data hiding in images is done in parallel not only it makes the entire process faster but by making the whole process faster in turn it makes the whole system a viable choice and making it applicable to various other implementations. Larger data and images can also be used. MPI – Message Passing Interface is implemented here for the parallelization of the complete code, but to increase the performance, the whole process is executed on a distributed network of machines. Parallelization of the whole system on a single machine definitely depicts the increase in performance and speed. Still, if the entire process is done on a network of machines, the performance of the whole algorithm increases drastically. Network File System is used for the transfer and assembly of the data between the machines in the distributed system. The proposed system for the working of the distributed computing is that the machines that will be connected in the distributed computing environment can connected over any network for our proposed system the distributed computing nodes were connected over LAN – Local Area Network. Out the machines connected to the network for the distributed computing, one machine is chosen to be the leader which does not do any computation but this machine is used to send out instructions to the machines in the distributed computing environment and do the finally assembly of the hidden message. The leader uses SSH-Secure Shell (SSH). The other machines are connected over the LAN network, the leader and

the communication relating to the transfer of data and the encrypted shares is done with the implementation of the NFS–Network File System.

2 Motivation and Related Work

The main aim of this paper is to improve the speed of execution of the above described method of Steganography using Visual Cryptography and Genetic Algorithm so that this method can have more applications and support larger amount data and images. [1] Describes a novel approach to Steganography with Visual Cryptography that uses Genetic Algorithm to make the detection of Steganography difficult thus making the whole process much more secure than previous methods. However the proposed method in [1] was time-consuming and therefore could only be applied on smaller image sizes and smaller data to get results. For bigger images and data proposed method in [1] took alot of time thus reducing the applications of the proposed method. Ghascmi al. [8] suggested a novel steganography scheme based on a transformation of the integer wavelet and Genetic algorithm. [9] compresses the secret message and encrypts it with the receiver's public key along with the stego key and uses an embedding algorithm to embed both messages into one carrier.

[10] technique was suggested of well-known k-n hidden sharing with random number using a variable length key with share division on color images. [11] describes how ASCII-based encryption can be implemented and how usage of gray code for encrypted message conversion and image embedding can protect the message, making it difficult for cryptanalysts to function.

3 Proposed System

The proposed system is basically making the whole earlier mentioned Steganography with Visual Cryptography along with Genetic Algorithm distributed that is to say that the entire computation for the confidential data hiding in the image by the implementation of the above-given concept can be performed on in parallel over a distributed system of computing nodes connected over a network, for our performance measures the computing nodes of the distributed computing environment that is used are connected over a network or LAN. However, this does not put any kind of restrictions or limit on the system might as well be connected over network, and the computing nodes of the distributed computing environment may be placed anywhere. The whole distributed environment is shown in Fig. 1. As depicted, there exists a leader that does not do any kind of computations. Still, it sends out instructions, and this leader node is the user interface via which the user can enter the image and the confidential data to be hidden inside the image. Thus the user has the concept that the whole proposed system is performed on this node, therefore, the entire network of the distributed computing environment remains invisible to the user. The part of the system that is related to secure data hiding in the image is described below;

basically, there exist two modules Steganography using Genetic Algorithm and Visual Cryptography. The concept of this model is depicted in Fig. 1.

The input image, which is chosen by the user, is accepted as the cover, which is then used to hide the confidential data that is also entered by the user. The method that is used for embedding the secret message is LSB – Least Significant Bit of the cover image. The implementation of the Genetic Algorithm is done to modify the pixel values of the steg-image to keep their statistic characters.

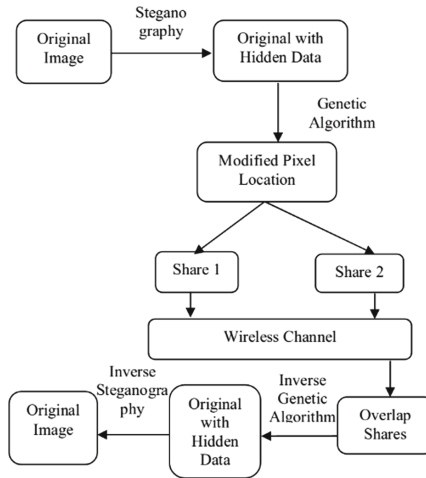


Fig. 1. Steganography and Visual Cryptography along with Genetic Algorithm [1]

The user selects their confidential information in terms of plain text to be hidden inside the LSB of the cover image. Visual Cryptography enables the value of the pixel of the steganography image to keep their intrinsic character that is to say that there are not made enough changes in the images that it may look susceptible that some data is hidden inside of it as it looks different from the original image. Genetic Algorithm is used to make it more difficult to identify where an image has even tampered with secret data or not. The above-explained method is however, slow when large images are used or large plain text that is to be hidden inside the image is given by the user. Therefore the whole above stated system is done on a distributed environment in parallel, thus improving its performance, speed, and widening its applications. The distributed system model for the Steganography and Visual Cryptography, along with the Genetic Algorithm, is depicted in Fig. 2 (Figs. 3, 4).

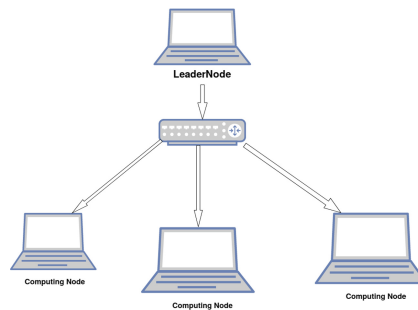
The computing nodes are connected over a network. There is one node to chosen as the leader node which does not execute any computation, the job of the leader node is mainly to distribute the task among the computational nodes of the distributed environment and the distribution of files, and the re-assembly of the final image and the hidden message is done by leader the node. To perform

Algorithm: Steganography**Input:** Cover Image**Output:** Stego Image

- Step 1: Read the cover image.
- Step 2: Find out the pixel values of cover image.
- Step 3: Read the secret data character wise.
- Step 4: Convert each character into its equivalent ASCII code.
- Step 5: ASCII code is converted into binary values.
- Step 6: Enter the secret key.
- Step 7: Secret data is converted into cipher data.
- Step 8: The stream of 8-bits (cipher data) are embedded into LSB of each pixel of the cover image.
- Step 9: To apply Genetic Algorithm in the stego image the pixel location should be modified.

Fig. 2. Steganography and Visual Cryptography along with Genetic Algorithm [1]**Algorithm:** Visual Cryptography**Input:** Stego-Image**Output:** Encrypted Shares

- Step 1: Read Stego-Image generated.
- Step 2: The stego image is broken into three layers namely split-1, split-2, split-3 these three files are containing the hidden data and to get the hidden data these three files have to be reconstructed perfectly.
- Step 3: The re-assembled picture and the extracted data will be gained again.

Fig. 3. Steganography and Visual Cryptography along with Genetic Algorithm [1]**Fig. 4.** Distributed environment.

the above-stated functions, the leader node must be able send instructions to the computing nodes to start multiple processes, collect the results from numerous processes, share the user input with the computing nodes connected in the distributed computing environment. This is made possible with the use of SSH – Secure Shell. So the challenge of Management of processes in compute nodes is overcome by the use of SSH – Secure Shell. The computing nodes connected in the distributed computing environment must be able to have real-time communication between processes running on different computing nodes. The computing

nodes must be able to share process data and status amongst themselves. This is completed via the implementation of MPI. The whole system is done in parallel, that is to say, that the entire user input data including images and the confidential message that is to hidden must be broken into parts and each computing node must execute its part in parallel with other nodes of the distributed computing environment. Thus for the transfer of data among the computing nodes in the distributed computing environment NFS: Network File System is used.

Thus now, the leader node is able to break full implementation of the algorithm into smaller pieces that can be distributed over to the computing nodes in the computational environment and are computing it in parallel. There are two phases of the module that is done on the distributed environment in parallel execution. The first phase is the parallelization of the data hiding in the image which was earlier done in a serial manner in Steganography and Visual Cryptography along Genetic Algorithm, since it was done serially there was a lot of scope for improvement of time taken by making the whole execution in parallel by breaking up the message as well as the image into smaller pieces and sharing these more minor information to the computing nodes in the distributed computing environment. Thus each computing node that receives this information thinks that this small piece of the image of the original user given image is the whole image and the smaller part of the message is the entire message that is to hide inside this images thus all computing nodes think that they are executing the whole earlier proposed system. Still, in fact, they are only achieving a small part of the complete problem in parallel with other nodes in the distributed computing environment. The second phase is the parallelization of the secret data extraction from the image which was earlier done in a serial manner in Steganography and Visual Cryptography along Genetic Algorithm, since it was done serially there was a lot of scope for improvement (same concept that is applied for the parallelization of the secret data extraction) of time taken by making the whole execution in parallel by breaking up the image into smaller pieces and sharing these more minor information to the computing nodes in the distributed computing environment.

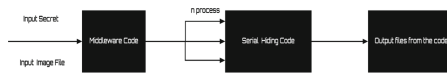


Fig. 5. Phase 1: distributed data hiding.



Fig. 6. Phase 1: distributed data extraction.

4 Performance

The main aim of this paper is to improve the overall performance of the algorithm. The Steganography and Visual Cryptography method although made the overall model much more secure by making it difficult for RS Steganalysis method to find out whether a given image has been tampered or not. But the whole algorithm performed well for small images and a limited amount of plain text that is to be hidden inside image. The algorithm did not perform well for large images and plain text that is the text give by the user to be hidden inside the image not only did the whole algorithm performed slow but the performance continuously decreased with increase in the sizes of the images given and increase in the length of the plain text entered by the user to be hidden inside the image. By running the above algorithm in Distributed Computing Environment a drastic increase in performance was observed. Lightweight images are images that were observed to be not that computationally intensive both for data hiding as well as data extraction. The Fig. 5, depicts the graph when our proposed system of the carrying out Steganography and Visual Cryptography along with Genetic Algorithm was implemented in a Distributed Computing System Connected via LAN.

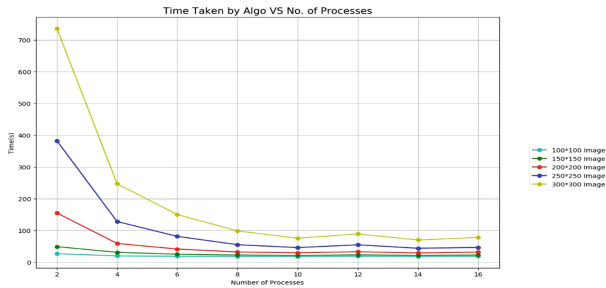


Fig. 7. Time improvisation for light-weight images.

The main comparison is the amount of total time taken when the whole system was divided into N number of processes to be executed among the computing nodes of the distributing environment. As observed from Fig. 5 we can observe that as number of processes that were to be executed in parallel over the distributed computing environment the amount of the total time continuously decreasing thus thereby indicating the distributed environment is definitely helping in the improvement of performance by improving the speed of the whole algorithm. Since we are doing the complete system in parallel, the calculation of speed - up is very important. The main comparison to be done next is the comparison of speed-up versus the number of processes deployed on the Distributed Computing Environment. As observed from the below given Fig. 6 we can observe that as the number of processes that were to be executed in parallel

over the distributed computing environment the speed up that was calculated increased each thus proving that parallelizing indeed was the force behind the increase in speed of the whole algorithm when done in parallel over a distributed environment (Fig. 8).

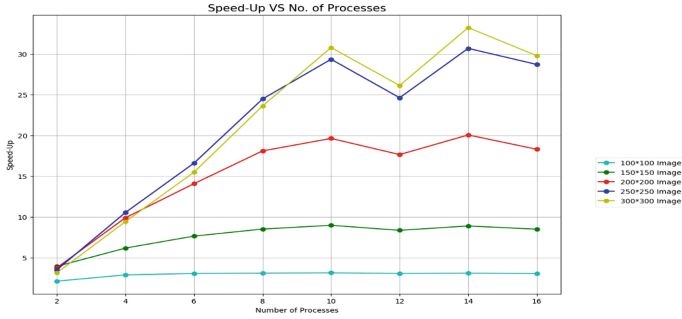


Fig. 8. Speed up for light-weight images.

However, although there is an observed improvement in the speed as well as time taken using Distributed Computing but the main aim was to make this whole system viable for larger images and input data. Thus the below given Fig. 7 shows the variation of Speedup vs No. of Processes for Heavy-Weight Images. Heavy-Weight images are images that are much more computationally intensive and the size of these images are greater than of 300*300 we have used 700*700 pixels image as the largest Heavy Weight Image (Fig. 9).

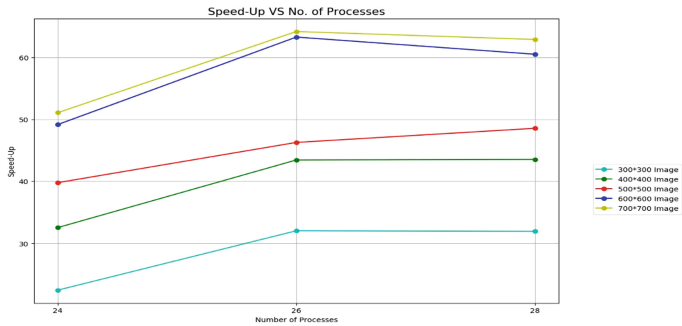


Fig. 9. Speed up for heavy-weight images

From the above stated image we can observe that a very high speed-up is obtained there by confirming the aim of our project. Because the high speed up observed the proposed method of Steganography and Visual Cryptography along with Genetic Algorithm run on a Distributed Computing Environment makes its must faster and improves its performance in all areas of measurement.

5 Future Improvements

Our current implementation and creation of the Distributed Computing environment only consisted and was tested on Laptops. But however traditionally this model would not need any improvement given the increase in speed up and time and any other kind of machine used as a computing node would most likely be a personal computer. But recent trends in the Mobile Industry has given rise to production of highly powerful mobiles with out great performance and compute power which in some cases can also be compared with the performance of personal computers. Thus since Mobiles are much more ubiquitous and since they have good computing power they could also be included in the Distributed Computing Environment. Mobiles fall short on performance in most cases compared to Personal Computers but this can be made up with the fact that mobiles are much more easier to find and the computing power of mobiles is not completely used by the user in almost all cases. Thus we would try to improve by Distributed Computing Environment by inclusion of Mobile Phones in the Distributed Computing Environment as computing nodes.

6 Conclusions

We conclude that when normal image encryption is implemented using steganographic and visual cryptographic technique, decryption of the encoded secret message makes the job of the investigators impossible. The steganographic safety features are highly optimized with the use of genetic algorithm. The system proposed is highly robust against RS attack and is optimally used for both grayscale and colored performance in visual hidden shares, making it highly compatible with real-time applications. Future work mainly would be to focus on to improve the algorithm using the neural visual cryptography network, so that the system can produce highly secure undetectable secret shares using selected training data that can be automatically generated and disposed of after the task has been completed. This method could make the safest and the most secure steganographic and visual cryptographic scheme. In this paper we have successfully parallelized in distributed computing environment a highly time consuming steganography algorithm using genetic algorithm. We have been successful in exploiting the paralyzable nature of the distributed environment and executing it. Thus with improve in the speed of execution this proposed method can be applied to both larger image files and larger text data that is to be hidden.

References

1. Prema, G., Natarajan, S.: Steganography using genetic algorithm along with visual cryptography for wireless network application. In: 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, pp. 727–730 (2013)

2. Manoharan, S.: An empirical analysis of RS steganalysis. In: Proceedings of the Third International Conference on Internet Monitoring and Protection. IEEE Computer Society, Washington (2008)
3. Fridrich, J., Goljan, M., Du, R.: Reliable detection of LSB steganography in colour and grayscale images. In: Proceedings of ACM Workshop on Multimedia and Security, Ottawa, 5 October 2001, pp. 27–30 (2001)
4. Chandramouli, R., Memon, N.: Analysis of LSB based image steganography techniques. IEEE (2001)
5. Fridrich, J., Goljan, M., Hogeia, D.: Steganalysis of JPEG images: breaking the F5 algorithm. In: Proceedings of the ACM Workshop on Multimedia and Security 2002 (2002)
6. Rana, R., Singh, D.: Steganography-concealing messages in images using LSB replacement technique with pre-determined random pixel and segmentation of image. *Int. J. Comput. Sci. Commun.* **1**(2), 113–116 (2010)
7. Yadollahpour, A., Naimi, H.M.: Attack on LSB steganography in colour and grayscale images using autocorrelation coefficients. *Eur. J. Sci. Res.* **31**(2), 172–183 (2009). ISSN 1450-216X
8. Ghasemi, E., Shanbehzadeh, J., ZahirAzami, B.: A steganography method based on integer wavelet transform and genetic algorithm. In: International Conference on Communications and Signal Processing (ICCSP), pp. 42–45 (2011)
9. Umamaheswari, M., Sivasubramanian, S., Pandiarajan, S.: Analysis of different steganographic algorithms for secured data hiding. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **10**(8), 154–160 (2010)
10. Kandar, S., Maiti, A.: Variable length key based visual cryptography scheme for color image using random number. *Int. J. Comput. Appl.* **19**(4), 139–145 (2011)
11. Bairagi, A.K.: ASCII based even-odd cryptography with gray code and image steganography: a dimension in data security **01**(02). Manuscript Code: 110112. ISSN 2078-5828 (Print), ISSN 2218-5224 (Online)



Wearable Device Design for Cattle Behavior Classification Using IoT and Machine Learning

Fatema Ahmed^(✉), Bholanath Roy, and Saritha Khetawat

Department of Computer Science, MANIT, Bhopal, India

Abstract. The term Internet of Things refers to the huge network of interconnected smart devices (sensors, actuators, RFID tags and readers) that can share and communicate information without any human intervention. One of the diverse applications of IoT is in the field of dairy farming for cattle behavior classification, automated heat detection (time period when the cows are sexually receptive) and calving time prediction. In this paper, first the work proposed by various researchers for cattle behavior classification using supervised and unsupervised machine learning techniques has been summarized. One of the limitations found in previous work was that the researchers used LM35 sensor for cow's body temperature measurement which read environmental temperature readings thus giving inaccurate results. Second limitation was quick discharge of wearable device due to high power consumption modules like Wi-Fi module. The proposed model discussed in this paper overcomes these limitations. The sensor system is divided into two parts: wearable transmitter module and receiver module. The wearable collar transmitter module consisted of only sensors and wireless transceiver module. It uses contactless infrared temperature sensor (Tm006) instead of LM35. The high-power consumption modules were placed in the receiver module which had continuous power supply. Hence, the power requirements were minimized and the battery life was increased in the wearable from few hours to 4 days.

Keywords: Internet of things (IoT) · Automated heat detection · Machine learning · Classification

1 Introduction

The goal of every dairy farmer is to improve the efficiency and productivity of their livestock animal. Nowadays with the help of IoT based devices, the farmers can get the information of their livestock animals located at remote locations sitting at one centralized location. The data collected from each individual animal gives a detailed insight of the animal's health, its location, reproductive cycle, feeding and grazing patterns and movement patterns. The connected sensors in the wearable device allow the dairy farmers to monitor the vital parameters like heart rate, blood pressure, temperature, digestion. From the wearable devices, the data is streamed directly to the cloud allowing the farmers in identifying and addressing issues like illnesses, feeding problems before they significantly pose a threat to animal's health.

The dairy farmers in many countries have to bare huge losses because of animal deaths due to illnesses. According to USDA, US dairy farmers lose nearly about 2.4 million dollars because of animal illnesses leading to death [13]. The connected sensors in the wearable devices monitor vital body parameters for notifying the farmers immediately on the first sign of any illness.

Nowadays, IoT based solutions are being used for prediction of calving time in cows. During the calving time, the dairy farmer has to be continuously vigilant. Careful monitoring of the calving process helps in minimizing the consequences of dystocia among cows [12]. IoT based devices are being used for sending an alert to the farmers as soon as the cow goes into labor.

Wearable devices are also being used for detecting heat (estrus period when the cow is sexually receptive). Typically, the standing heat period in cows lies for about 12–18 h but for some cows it can vary between 6–24 h [3]. So accurate detection of heat within this short period has become a major challenge. Nowadays the dairy farmers use artificial insemination injections for impregnating cows, so detection of heat period is crucial for obtaining successful outcomes. Calving interval (CI) is the time period between two successful calvings of a cow. Increase in calving interval means reduction in milk production. The main aim of a dairy farmer is to reduce the calving interval of the dairy cows. Ideally the average calving interval should be between 12–13.5 months [3]. Dairy farmers try to achieve this calving interval by using artificial insemination technique using automated heat detection. The cows should be inseminated 6–24 h prior to ovulation [2].

For cattle behavior classification, a wearable device is fitted across the neck or leg of the cow, consisting of a microcontroller, triple axis accelerometer for measuring the acceleration in all the three axes and a temperature sensor for measuring the core body temperature. The data from these sensors is collected at a regular interval. Both supervised and unsupervised machine learning approaches have been proposed by researchers for behavior classification.

The remaining sections in the paper are as follows: Sect. 2 provides an overview of the application areas of IoT in livestock management, Sect. 3 consists of literature review, Sect. 4 gives an overview of proposed wearable device design and its deployment and data collection, Sect. 5 gives insight of feature extraction and machine learning, Sect. 6 discusses the model training and results obtained and finally Sect. 7 concludes the paper.

2 IoT Application Areas in Livestock Management

There are various areas in livestock management where IoT is being used. Some of these applications are discussed below [13].

2.1 Health Monitoring

Connected sensors in the wearable allow the real time monitoring of health parameters like digestion, blood pressure, temperature, respiratory rate and other vital features for allowing the farmer to be alerted at first sign of any illness in the livestock animal.

2.2 Monitoring Reproductive Cycles and Calving

IoT is nowadays being used for monitoring cow's reproductive cycle. Such as, to monitor when a cow goes into heat and when they are ready for giving birth. The heat period of cows can varied from 8 h to 12 h which can be detected using IoT based wearable devices.

2.3 Fine Tuning Feeding

The wearable devices can be used by the farmers to monitor specific behavioral patterns of herd like how frequently they feed (in-order to avoid food wastage and over feed).

2.4 Maximizing Milking

Sensors used in the wearable can be used for detecting when the cow needs to be milked, for tracking animal's milking speed and for analyzing the amount and quality of milk.

2.5 Tracking Location

The wearable devices can be also used for tracking the location of livestock animal especially in farms spread over multiple acres. Location tracking allows farmers in immediately locating the sick animal or the animal that has gone into heat so that the farmer can separate that animal for providing treatment.

3 Literature Survey

Jorge A. Vazquez Diosdado et al. [1] proposed a simple decision tree algorithm that was less computationally expensive and that could be easily implemented in the bio telemetry device. The authors used a triple axis accelerometer in their wearable device for their data collection and tried to classify the behavior of cows into three categories (standing, lying and feeding). They laid down a comparison of their proposed algorithm with computationally expensive algorithms like SVM and Hidden markov model. The two characteristics features that were used in their proposed algorithm were SCAY which measured the static component of acceleration in the y-axis and VeDBA which was used for measuring overall dynamic body acceleration. The mean of the VeDBA values for a particular window size was compared by the authors with their chosen threshold A (0.0413 g), if the mean was found to be higher, the activity was classified as high activity (feeding). If the mean was found to be lower, then the mean of SCAY values was compared by the authors with their chosen threshold B (-0.055 g). If the calculated mean for the window size was higher than the threshold, then the activity was classified as standing otherwise the activity was classified as lying.

Ritaban Dutta et al. [4] proposed a multi classifier pattern recognition system that was based on supervised machine learning algorithms. The wearable collars used by the authors consisted of GPS antennae, receiver, triple axis accelerometer and 3-axis magneto resistive sensor. Their proposed system was applied to classify five common cattle behavior classes, namely Grazing, Ruminating, Resting, Walking, and Scratching. The multi-classifier pattern recognition system proposed by the authors consisted of two stages. In the first stage, a hybrid unsupervised clustering framework was used which consisted of Fuzzy C means, self-organizing map network and principle component analysis algorithms. This hybrid unsupervised framework was used for grouping the large dataset datapoints into separable data clusters. In the second stage, an ensemble classifier consisting of Bagging, AdaBoost and Random Subspace method along with other supervised classification methods like LDA classifier, Naïve Bayes classifier, K-nearest neighbor classifier and adaptive neuro fuzzy inference classifier was used.

Ciira wa Maina [9] developed a methodology to detect daily activity pattern of dairy cattle and tried to correlate the daily activity pattern with heat event and illness detection in dairy cattle. The authors used k-nearest neighbor classifier and classified the cattle behavior into the following categories (eating at trough, eating grass, walking and standing). The authors partitioned the continuously gathered data into blocks of 64 samples each and calculated mean, standard deviation, one lag correlation and cross correlation between the three axes. The continuously gathered data collected by the authors was randomly divided into two parts (70% was used for training and the remaining 30% was used for testing). K-nearest neighbor classifier was used by the authors for classification.

Rahman et al. [5] devised an experiment by trying three different positions of the accelerometer sensor on the animal's body (neck, ear and head). The authors used two testing approaches (leave out one animal approach and stratified cross validation approach). In the leave out one animal approach, the authors trained the models from the data collected from a set of animals and the models were tested on the animals that were not used in the training process. In the stratified cross validation approach, mixed data from a set of animals were collected and the models were trained on a part of that data and models were tested on the data that were not used in the training process. The authors observed that the data collected from the head (halter) position provided better accuracy. They also observed that stratified cross validation approach performed better than leave out one animal approach.

Mohit Taneja et al. [6] proposed a fog computing assisted application system for animal behavior analysis and health monitoring. The authors devised this system to overcome the large communication delays which is a major disadvantage of cloud computing. They performed data analytics and preprocessing at the network edge itself and hence reduced the data sent to the cloud. They classified the gathered data into the following categories: 1. Latency insensitive data (data not requiring immediate analysis), 2. Latency sensitive data (data of high critical value and requiring immediate analysis) and 3. Latency tolerant data (data that is generally time insensitive but becomes sensitive under certain circumstances). In their proposed system, the data collected from the sensors was sent to the fog node. If the data was latency sensitive, then alert or response was generated otherwise the aggregated and processed data was sent to the cloud for historical storage and analysis.

A. Apinan et al. [7] proposed a simple technique to classify cattle behavior into standing, walking, grazing and lying by using magnitude and variance of accelerometer output signal. They proposed a simple decision tree algorithm having low resource requirements as it requires less mathematical operations. In the first step, they found the average of all the three axes C_x , C_y , C_z . They compared these averages with the reference voltage levels (R_{rx} , R_{ry} , R_{rz}). If $C_x > R_{rx}$ and $C_y > R_{ry}$ and $C_z > R_{rz}$ then the activity was classified as lying otherwise the activity was classified as standing or walking-grazing. In the second step, standing and walking-grazing activity were classified using variance of Y axis.

Nariyasu Watanabe et al. [11] proposed a new automatic classification system for eating, ruminating and resting activities of cattle using a three-axis accelerometer which measured the under-jaw acceleration at 1 s interval. The authors calculated 12 variables for one-minute window including mean, standard deviation and inverse coefficient of variance ($\frac{mean}{\sqrt{variance}}$) for x, y, z axis and their resultant. Quadratic Discriminant Analysis was used by authors to classify eating, ruminating and resting activities. The authors divided the dataset into a ratio of 2:1 and repeatedly carried out this division thousand times using bootstrap method. Wiki's λ statistic was used by the authors to find whether the combination of acceleration variables was effective for discriminant function analysis or not.

Daniel Smith et al. [10] proposed a feature selection method using greedy algorithm for selecting a set of features from a wide set of features (statistical features, spectral features and information theory coding-based features), that were most effective in discriminating between six different cattle behaviors. The authors used three machine learning classifiers: - support vector machine, decision tree and linear discriminant analysis. A total of 14 features were extracted from 5 s non overlapping time series that included (Acc_x , Acc_y , Acc_z , Acc_m , Pressure) where $Acc_m = \sqrt{Acc_x^2 + Acc_y^2 + Acc_z^2}$. The authors validated the performance of each classifier using F score metric by performing leave one cow out cross validation that is observation of six cows was used for training the model and the observation from seventh scow was used for testing the model. This process was carried our seven times. The authors found out that second order moments standard deviation and sum of absolute values of acc_x and acc_z were the most effective discriminant features selected. The comparison of existing work done has been laid out in Table 1.

Table 1. Comparison of existing work

Paper	J.A.V. Diosdado et al. [1]	R. Dutta et al. [4]	A. Rahman et al [5]
Features used	Static component of acceleration in y-axis (SCAY) and Vectoral dynamic body acceleration (VeDBA) $\sqrt{ax^2 + ay^2 + az^2}$	For each block mean, standard deviation, difference between maximum and minimum sample, inverse coefficient of variance were calculated for each axis (X, Y, Z) corresponding to total 12 features	Acc_m was calculated as $\sqrt{ax^2 + ay^2 + az^2}$ For the acc_m window of 200 samples, mean, standard deviation, skewness and kurtosis were calculated. Average of X, Y, Z axis and variance of Y axis were also used
Machine learning technique used	Decision tree algorithm	K nearest neighbor	Random forest (collection of decision trees)
Components used in wearable collar	Omnisense 500 series tags containing accelerometer	GPS receiver chip, active GPS antennae, Atmel ATmega microcontroller, transceiver module (915 MHz), 4 alkaline D-cell batteries connected in series, Micro SD card (4 GB), micro-electromechanical chip containing a triple axis accelerometer and a 3-axis magneto-resistive sensor	Piezoelectric micro electromechanical system (MEMS) chip containing a triple axes accelerometer and a 3-axis magneto-resistive sensor, and micro SD card module
Results	Lying (Sensitivity 74.0, Precision 96), Standing (Sensitivity 82, Precision 47.01), Feeding (Sensitivity 95.65, Precision 92.03)	K nearest gave classification accuracy of 93%, 97% sensitivity and 90% F1 score	Grazing 0.809, Standing 0.874, Ruminating 0.913

4 Proposed Wearable Device Design and Data Collection

The problem of power backup of wearable devices was one of the major problems faced in previous works. The Wi-Fi module and SD card module are power draining and consume a lot of power. So, the wearable devices needed to be recharged at short intervals of 6 h using lithium battery (3.7 V, 2.8 A). In order to solve this problem the sensor system is divided into two parts: transmitter module and receiver module (shown in Fig. 1). The transmitter module consists of temperature sensor, acceleration sensor and Nrf24l01 transceiver module. All above sensors consume less power. The

transceiver module is used for transmitting sensor data through wireless communication. The receiver module consists of Nrf24101 transceiver module, SD card module and Wi-Fi module. The transceiver module receives the data transmitted by transmitter module and then stores the data on SD card module. The received data is also sent to the cloud through Wi-Fi. However both SD card module and Wi-Fi module consume high power. So the receiver module is connected with the main power supply. The block diagram of our proposed system has been shown in Fig. 1 respectively.

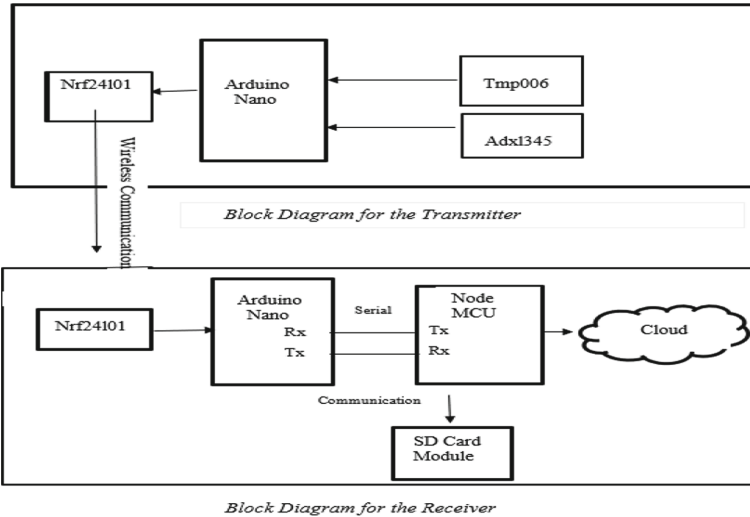


Fig. 1. Block diagram of proposed system

4.1 Wearable Collar Transmitter

The wearable collar (shown in Fig. 2) consists of the following components: a. Arduino Nano microcontroller, b. Adxl345 sensor for measuring static as well as dynamic acceleration in all the three axes, c. TMP006 sensor (contactless infrared temperature sensor) for measuring core body temperature, d. NRF24L01 transceiver module used for wireless communication and e. Lithium ion (3.7 V, 2.8 A) battery is used for supplying power to the wearable device.

The Arduino Nano microcontroller is programmed using Arduino IDE to continuously collect the acceleration as well as temperature data from the sensors at 10 s interval and wirelessly transmit this data to the receiver module.



Fig. 2. Wearable collar transmitter

4.2 Wearable Collar Receiver Module

The receiver module, as shown in Fig. 3 consists of the following components: a. Arduino Nano Microcontroller, b. Node MCU Microcontroller, c. NRF24L01 transceiver module with antenna used for wireless communication, d. RTC (Real Time Clock Module), e. Micro SD card Module with 32 Gb SD Card.

The Arduino Nano microcontroller is programmed to receive the data using NRF24L01 transceiver module, sent by the NRF24L01 module in the transmitter. The data received by Arduino Nano is sent to Node MCU microcontroller using serial communication. The Node MCU microcontroller is programmed using Arduino IDE to store the received data on the SD card as well as it is connected with Wi-Fi and continuously sends the received data to ThingSpeak IDE. The data is continuously sent at an interval of 20 s to ThingSpeak IDE.

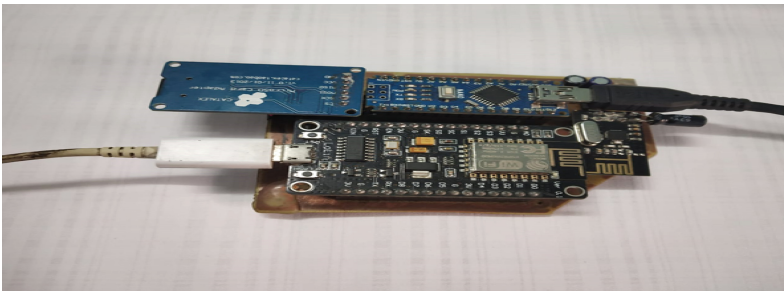


Fig. 3. Receiver module

4.3 Data Collection

The wearable device was mounted on the cow's neck (as shown in Fig. 4) in such a way that:

- a. X axis measured the left and right movement of the cow.

- b. Y axis measured the forward and backward motion of the cow.
- c. Z axis measured the sidewise rotation of the neck.



Fig. 4. Positioning of wearable device on the cow's neck

A ThingSpeak channel named Cow Collar was created having four fields X axis, Y axis, Z axis and Temperature. These four fields were continuously updated at an interval of 10 s for 60 days.

5 Feature Extraction and Machine Learning

From the acceleration X, Y and Z series two additional series were obtained:

- a. SMV (signal magnitude vector) = $\sqrt{ax^2 + ay^2 + az^2}$ and
- b. SMA (signal magnitude area) = $|accx| + |accy| + |accz|$

Signal magnitude area was used for distinguishing between motion and resting periods of cow. Signal magnitude vector was used for determining the intensity of motion in cows. For each series that is X, Y, Z, SMA and SMV, a window of 15 samples was taken. For each series minimum, maximum and average were taken out for the window size, thus constituting a total of 15 features and one feature was used for representing the core body temperature. Figure 5 and Fig. 6 shows the screenshot of dataset obtained.

Max x	Max y	Max Z	Max Abs	Max Sqr	Min X	Min Y	Min Z	Min Abs
40	-8	5	53	41.097445	34	-8	-2	44
39	-1	7	53	40.632499	32	-11	3	43
23	16	17	53	31.416556	23	13	13	52
24	7	19	50	31.400637	24	7	19	50
-14	3	21	38	25.41653	-14	3	21	38
-17	17	21	55	31.921779	-17	17	21	55
-22	2	24	48	32.619013	-22	2	24	48
-21	7	23	51	32.388269	-23	6	22	51
-20	15	20	54	31.464265	-21	11	18	50
-24	3	21	48	32.031235	-24	3	21	48
-17	10	28	55	34.249088	-17	10	28	55
-20	10	23	53	32.07803	-20	10	23	53
-20	16	25	61	35.79106	-20	16	25	61
-21	6	22	49	31	-21	6	22	49
-24	5	20	49	31.638584	-24	5	20	49
-21	6	23	49	31.543621	-21	5	21	48
-20	6	23	49	31.064449	-20	6	23	49
-21	5	23	49	31.890437	-23	2	22	47
-17	10	24	51	31.717503	-21	6	23	50
32	-2	6	41	32.695565	30	-3	2	34
32	-1	6	39	32.572995	32	-1	6	39

Fig. 5. Screenshot of the unlabeled dataset

Min Sqr	Avg X	Avg Y	Avg Z	Avg Abs	Avg Sqr	Temp
34.985711	37	-8	1.5	48.5	38.041578	28.52
32.939338	35.666667	-5.666667	5.3333333	46.666667	36.753237	30.33
30.88689	23	14.5	14.75	52.25	31.019307	33.3
31.400637	24	7	19	50	31.400637	33.66
25.41653	-14	3	21	38	25.41653	33.12
31.921779	-17	17	21	55	31.921779	33.5
32.619013	-22	2	24	48	32.619013	32.81
31.921779	-22	6.5	22.5	51	32.155024	32.86
29.698485	-20.333333	13	19	52.333333	30.763838	32.63
32.031235	-24	3	21	48	32.031235	32.83
34.249088	-17	10	28	55	34.249088	32.76
32.07803	-20	10	23	53	32.07803	33.12
35.79106	-20	16	25	61	35.79106	33.06
31	-21	6	22	49	31	32.36
31.638584	-24	5	20	49	31.638584	32.08
30.298515	-21	5.5	22	48.5	30.921068	32.02
31.064449	-20	6	23	49	31.064449	31.85
31.543621	-22	3.5	22.5	48	31.717029	31.55
31.064449	-19	8	23.5	50.5	31.390976	31.74
30.133038	31	-2.333333	4.6666667	38	31.489063	32.6
32.572995	32	-1	6	39	32.572995	32.66

Fig. 6. Screenshot of the unlabeled dataset (contd)

6 Model Training and Result Analysis

(a) Unsupervised K-Means Clustering for Labeling Data

The data that was collected from sensors was unlabeled data. Labeling of data was not possible as it required continuous manual observation for observing the cow's behavior. So, in the first stage K means clustering was used for clustering the data points into 4 clusters (0, 1, 2, 3). When the clustered data was matched with manual observations then the four clusters represented the following activities: - 0 represented standing, 1 represented walking, 2 represented grazing and 3 represented lying.

The centroids obtained after K- means clustering are shown in Fig. 7.

	0	1	2	3	4	5	6	7	8
0	2.23965	-4.11844	-27.4872	44.7213	32.3475	-0.999537	-9.04256	-29.684	40.0955
1	2.38945	-3.60385	31.1198	39.9648	32.1348	1.3459	-4.44472	30.536	34.3475
2	26.0282	1.75267	-8.65997	46.7073	31.4264	24.528	-0.39611	-10.0697	45.1549
3	-5.5476	2.81937	27.5207	46.6493	32.7525	-11.1546	-3.23434	22.9731	41.5017
	9	10	11	12	13	14	15		
	30.4839	0.618147	-6.63326	-28.6222	42.3941	31.4088	31.3791		
	35.6774	28.0534	2.10211	-7.61921	37.7939	35.4869	31.9828		
	30.7518	25.1743	0.527012	-9.17029	44.0295	33.8205	32.5068		
	30.7078	-8.44182	-0.377136	25.5808	43.9168	31.7449	31.3362		

Fig. 7. Snapshot of centroids obtained after k- means clustering

(b) Supervised K Nearest Neighbor

After K means clustering was applied for labeling of data, then the labeled data was used for training K nearest neighbor for cattle behavior classification. The dataset was splitted into a ratio of 80–20. 80% of data was used for training the model and remaining 20% was used for testing the model. The value of k was varied for checking accuracy and other performance measures.

For different values of k, the performance of k- nearest classifier was evaluated. The precision, recall, F1-Score values for different activities (standing, walking, grazing, lying), taking k = 5, k = 6, k = 7 and k = 10 are shown in Tables 2, 3, 4, 5 respectively.

It was observed that the k-nearest classifier performed best for the value of k = 5 with precision values of 0.99 for standing, 0.98 for walking, 0.99 for grazing, 0.98 for lying and recall values of 1.00 for standing, 0.98 for walking, 0.99 for grazing and 0.99 for lying.

Table 2. Performance of k- nearest classifier for k = 5

Activities	Performance measures			
	Precision	Recall	F1 – Score	Support
0 (Standing)	0.99	1.00	0.99	824
1 (Walking)	0.98	0.97	0.98	276
2 (Grazing)	0.99	0.98	0.99	622
3 (Lying)	0.98	0.99	0.99	245

Table 3. Performance of k- nearest classifier for k = 6

Activities	Performance measures			
	Precision	Recall	F1 – Score	Support
0 (Standing)	0.99	1.00	0.99	878
1 (Walking)	0.98	0.98	0.98	255
2 (Grazing)	0.99	0.99	0.99	583
3 (Lying)	0.98	0.99	0.99	251

Table 4. Performance of k- nearest classifier for k = 7

Activities	Performance measures			
	Precision	Recall	F1 – Score	Support
0 (Standing)	0.98	0.99	0.99	864
1 (Walking)	0.97	0.91	0.94	247
2 (Grazing)	0.97	0.98	0.98	638
3 (Lying)	0.97	0.98	0.97	218

Table 5. Performance of k- nearest classifier for k = 10

Activities	Performance measures			
	Precision	Recall	F1 – Score	Support
0 (Standing)	0.99	0.99	0.99	872
1 (Walking)	0.98	0.94	0.96	227
2 (Grazing)	0.98	0.99	0.99	621
3 (Lying)	0.98	0.97	0.97	247

7 Conclusion

Internet usage has been increasing enormously throughout the world. With the introduction of IPV6 supporting 128-bit address space allowing billions of devices to be connected to the internet, IoT is becoming more popular. One of the many applications of IoT is in the field of dairy farming. This paper discussed the applications of IoT in dairy farming and the work proposed by various researchers for cattle behavior classification. One of the limitations found in previous work was that the researchers used LM35 sensor for cow's body temperature measurement [8] which read environmental temperature readings thus giving inaccurate results. Second limitation that was found out was quick discharge of wearable device due to high power consumption modules like Wi-Fi module [5]. These limitations were overcome in our proposed design.

In our proposed wearable design contactless infrared temperature sensor (Ttmp006) was used instead of LM35. The high-power consumption modules were placed in the receiver module which had continuous power supply. The wearable collar transmitter module consisted of only sensors and wireless transceiver module. Hence, the power requirements were minimized and the battery life was increased in the wearable from few hours to 4 days.

References

1. Diosdado, J.A.V., et al.: Classification of behavior in housed dairy cows using an accelerometer based activity monitoring system. *Anim. Biotelemetry* **3**(1), 3–15 (2015)
2. Shahriar, M.S., et al.: Detecting heat events in dairy cows using accelerometers and unsupervised learning. *Comput. Electron. Agric.* **128**, 20–26 (2016)
3. Zin, T.T., Kai, H., Sumi, K., Kobayashi, I., Hama, H.: Estrus detection for dairy cow using a laser range sensor. In: Third International Conference on Computing Measurement Control and Sensor Network (CMCSN), Matsue, pp. 162–165 (2016)
4. Dutta, R., Smith, D., Rawnsley, R., Bishop-Hurley, G., Hills, J.: Cattle behaviour classification using 3-axis collar sensor and multi-classifier pattern recognition. In: *SENSORS*, IEEE, Valencia, pp. 1272–1275 (2014)
5. Rahman, A., Smith, D.V., Little, B., Ingham, A.B., Greenwood, P.L., Bishop-Hurley, G.J.: Cattle behavior classification from collar, halter, and ear tag sensors. *Inf. Process. Agric.* **5**(1), 124–133 (2018)
6. Taneja, M., Byabazaire, J., Davy, A., Olariu, C.: Fog assisted application support for animal behavior analysis and health monitoring in dairy farming. In: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, pp. 819–824 (2018)
7. Apinan, A., Rattanawong, T., Kuankid, S.: Classification of the cattle behaviors by using magnitude and variance of accelerometer signal. *Agric. Eng. Int.: CIGR J.* **17**(4), 397–402 (2015)
8. Gokul, V., Tadepalli, S.: Implementation of smart infrastructure and non-invasive wearable for real time tracking and early identification of diseases in cattle farming using IoT. In: International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), Palladam, pp. 469–476 (2017)
9. Maina, C.: IoT at the grassroots—Exploring the use of sensors for livestock monitoring. In: 2017 IST-Africa Week Conference (IST-Africa), Windhoek, pp. 1–8 (2017)

10. Smith, D., Little, B., Greenwood, P.I., Valencia, P., Rahman, A., Ingham, A., Bishop-Hurley, G., Shahriar, S., Hellicar, A.: A study of sensor derived features in cattle behavior classification models. In: IEEE SENSORS, Busan, pp. 1–4 (2015)
11. Watanabe, N., Sakanoue, S., Kawamura, K., Kozakai, T.: Development of an automatic classification system for eating, ruminating and resting behavior of cattle using an accelerometer. *Grassland Sci.* **54**, 231–237 (2008)
12. Speroni, M., Malacarne, M., Federico, R., Franceschi, P.: Increasing of posture changes as indicator of imminent calving in dairy cows. *Agriculture* **8**, 182 (2018)
13. Using IoT to Increase Efficiency, Productivity for Livestock. <https://www.precisionag.com/systemmanagement/using-iot-to-increase-efficiency-productivity-for-livestock/>



Fuzzy Edge Image Steganography Using Hybrid LSB Method

Debina Laishram^{1(✉)}, Themrichon Tuithung¹, and Tayenjam Jeneetaa²

¹ National Institute of Technology, Chumukeidma, Dimapur 797103, Nagaland, India

² Manipur Technical University, Imphal 795004, Manipur, India

Abstract. The identification of correct edge location is an important step in edge based image steganography. The edge pixels in an image are characterized by sudden sharp change in intensity; changes made at these locations are inconspicuous to the human eye making it optimal for modification. In this paper, use of Fuzzy inference system to identify the correct edges have been extended to perform edge image steganography, that hides secret message using a hybrid LSB technique. The aim of the proposed hybrid LSB method is to achieve high payload embedding that ensures statistical resemblance between the cover and the generated stego image. It embeds two message bits in the seventh and eighth edge bit using LSB-Matching (LSBM) and LSB-Replacement (LSBR). The performance of our method shows substantial improvement in terms of security and image quality compared to existing edge based image steganography approaches. The experimental analysis shows that the stego image achieves high PSNR of 61 dB and SSIM of 0.99 with minimum level of distortion at high embedding rate observed by gradually increasing % pixels modifications. The method is validated to be secure against RS analysis and Pixel-wise Histogram steganalysis, tested with 2000 grayscale images from BOSSBase 1.0 database.

Keywords: Edge image steganography · Fuzzy inference system · Least significant bit-matching · Least significant bit-replacement · Steganalysis

1 Introduction

Steganography is described as the art and science of covert information communication in a completely undetectable manner using digital medium such as images, videos, texts and audio [1]. Every steganographic system should aim to achieve high embedding capacity, imperceptibility and robustness against visual and statistical attacks [1, 2]. In digital image steganography, covert communication is achieved by hiding secret message in parts of the cover image so that the resultant stego image is visually and statistically similar, if not exactly to cover. On the contrary, steganalysis refers to the technique that are designed to break the steganographic system by distinguishing between the cover and stego images.

Visual imperceptibility and statistical resemblance of cover and stego images is dependent on the pixels chosen for embedding [1]. Edge pixels in the image make better choice for steganography because it is often characterized as noise and is difficult to model due to sudden sharp change in visual and statistical properties [1]. The smooth region are statistically weak under attacks because even minor change can cause severe visual distortions. Since edges contribute to only a fraction of the overall image content, it may be insufficient and a challenge to hide high payload resulting in loss of information [3].

In this paper, we use the fuzzy inference system (FIS) edge detector from Matlab and propose an embedding method which is an amalgamation of LSBM and LSBR to produce a hybrid LSB which hides two message bits in the edge pixel. We compare and test the application of this scheme on grayscale images from BOSSBase 1.0 database. Our method shows considerable improvement over other edge based approaches in terms of payload capacity without compromising on the visual and statistical similarity of the cover and generated stego image. The rest of the paper is arranged as follows. Section 2 recurs and discusses relevant edge steganography schemes. Section 3 demonstrates the proposed data embedding and extraction methods. Section 4 presents experimental results and discussions on stego image quality, security against steganalytic attacks. Finally Sect. 5 concludes and iterates the future scope of the work.

2 Relevant Approaches

Spatial domain Least Significant Bit (LSB) based embedding are proven to achieve high payload data hiding. The Least Significant Bit-Replacement (LSBR) [4] also known as Least Significant Bit-Substitution (LSBS) and Least Significant Bit-Matching (LSBM) [5] are very similar in their approach. In both, the sequence of pixels to modify is determined by a pseudorandom number generator (PRNG) where the message bits are randomly hidden. In LSBR, the least significant bits of the pixel are simply overwritten with the message bits. Even at low embedding payload it causes structural asymmetry that are easily detectable by steganalytic algorithms like RS analysis, histogram analysis and the general structural steganalysis [6]. In LSBM, whenever the secret message bit does not match the LSB of the cover $+1$ or -1 is randomly added after embedding the message bits. To extract the secret message, the same sequence of pixels have to be traced. It was observed that it changed the center of mass (COM) of histogram characteristic function (HCF) of the stego image [7]. Improvised steganalytic algorithms based on HCF-COM called calibrated HCF-COM and calibrated adjacency HCF-COM was proposed [5, 8]. A variation of LSBM called Least Significant Bit Matching Revisited (LSBMR) [9] hides the secret message bits into pixel-pairs. Given a pair of adjacent cover pixels (x_i, x_{i+1}) the corresponding stego pixels (y_i, y_{i+1}) is obtained such that $LSB(m_i)$ and $LSB(\frac{y_i}{2} + y_{i+1}) = m_{i+1}$ using LSBM and LSBR where m_i and m_{i+1} are the two message bits. Pixel value differencing (PVD) [10, 11] based embedding works by dividing the image pixels into non-overlapping blocks. The smoother regions

have small difference and edge/textured regions have larger difference. Using the sequence determined by PRNG and depending on the difference different amount of data embedding is performed.

Edge pixels are characterized as noise. It is preferred choice for embedding because it can withstand modifications with passable statistical and no visible artifacts. Edge based steganography techniques rely on the correct identification of the edges using traditional edge detectors like Sobel, Prewitt, Robert, Laplacian and Canny. Recent advances include edge detection using XOR-coding [12], median filter with fuzzy technique [13]. An adaptive edge embedding using LSBMR called AE-LSBMR was proposed by Luo et al. [14] to preserve the statistical and visual correlation of neighboring pixels. It works by embedding into the sharper edges according to threshold determined by the size of the payload and edge gradients. The hybrid LSB [15] used a seven bit secondary image to determine the highly textured region using complexity threshold then randomly selects a cover pixel and embeds two message bits using LSBM and LSBR in the first and second bit plane. The use of fuzzy inference system for edge identification by Sanjeev et al. [16] defines the horizontal and vertical edge image gradients than LSBR is used to modify the two LSB of the edge pixel with two message bits. Performance of the detector is determined by the standard deviation values of the membership function. High value will decrease the intensity of the detected edge pixels and vice versa.

3 The Proposed Scheme

The flow diagram of our proposed method is shown in Fig. 1. As a preliminary the edge image is generated for the cover image using FIS edge detector [17]. Then depending on the message payload two bits is modified in the edge pixel using the hybrid LSB embedding. In the extraction phase, it obtains the secret message according to the payload by extracting two bits from the edges. The details of data embedding and extraction algorithm is as follows.

3.1 The Embedding Procedure

The Preliminary Edge Detection. We adopt the FIS edge identification technique in Matlab [17] to identify the edges correctly and precisely using a set of fuzzy condition based on intensity changes to identify the sharp and weak edges. The method relies on image gradients along the x and y axis to locate the edges. The gradients are then, used to find breaks in uniform area and is measured by taking the divergence of the neighboring pixels.

$$\begin{aligned} H(i, j) &= X(i, j + 1) - X(i, j - 1) \\ V(i, j) &= X(i + 1, j) - X(i - 1, j) \end{aligned} \quad (1)$$

where (i, j) denotes the i^{th} row and j^{th} column of the image X. Then to determine the intensity of each pixel provide zero mean Gaussian membership function. The

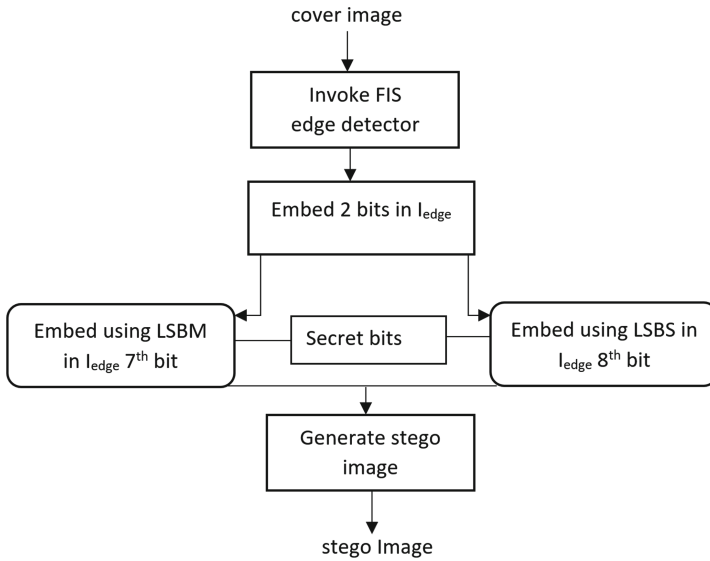


Fig. 1. Proposed method flowchart.

sample grayscale cover images a, b and c and the corresponding edge images is shown in Fig. 2.

The Hybrid LSB Embedding. We start by initializing the PRNG by a seed which is shared between the sender and receiver. The fuzzy edge image of the cover is denoted by I_{edge} . The hybrid LSBM-LSBR embeds two message bits at a time in each edge pixel depending on the payload. The I_{edge} value will be 1 for edge and 0 for non-edge pixel. Figure 3 shows the location for 50% and 100% edge pixel changed and Fig. 4 shows the generated stego images for the three sample cover with varying modification %. Post the embedding procedure, it can be observed that no visual artifacts are introduced and the resemblance between the cover and stego is high. The embedding algorithm pseudocode is as follows:

```

    set PRNG
    [row,col] ← size(cover)
    n ← 0
    stego ← cover
    for each pixel cover(x,y) do
      if I_ledge(x,y)=1 then
        if cover(x,y) mod 2 ≠ message(n) then
          Embed in the 7th bit of the cover pixel
          stego(x,y) ← cover(x,y) ± 1
        end if
      end if
    end for
  
```

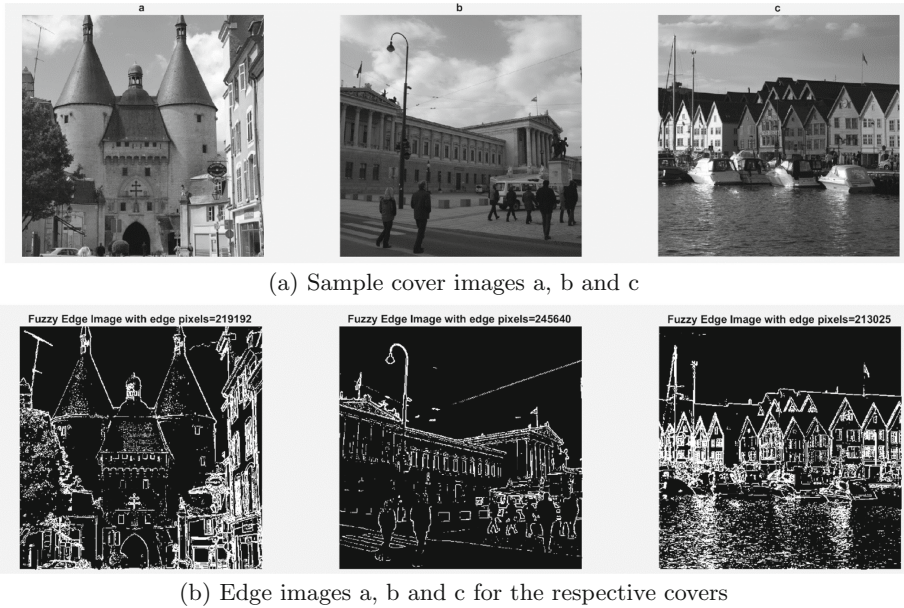


Fig. 2. Sample cover images and the corresponding edge images generated using FIS detector.

```

n ← n+1
  Embed in the 8th bit of cover pixel
  stego(x,y) ← message(n)
n ← n+1
end if
end for

```

3.2 The Extraction Procedure

The preliminary stage is the same where stego image is taken as the input to generate the edge image using FIS, which is denoted by S_{edge} . From S_{edge} , depending on the payload, the two LSB of each edge pixel is extracted using the shared PRNG seed to retrieve the secret message. The extraction pseudocode is as follows:

```

[row,col] ← size(stego)
n ← 0
for each pixel stego(x,y)
  if  $S_{edge}(x,y) == 1$  then
    twobits = stego(x,y) mod 4
    message(n:n+1) ← dec2bin(twobits)
    n ← n+2
  end if
end for

```

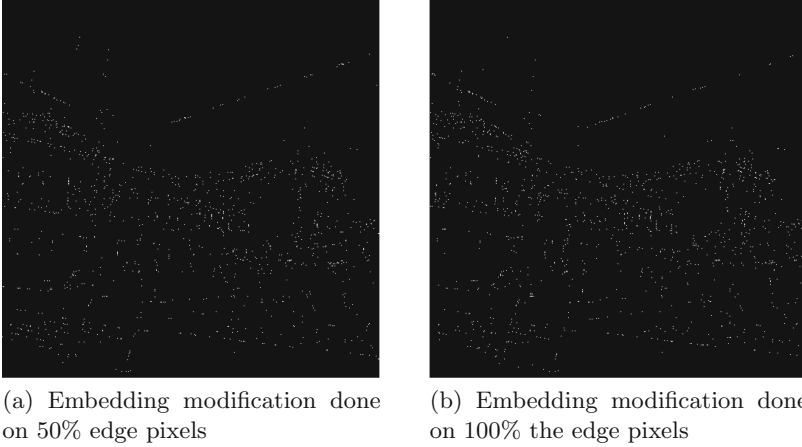


Fig. 3. Difference between stego and cover image (b.pgm) post embedding. The black and white pixels denote the changed locations.

4 Experimental Results

The proposed method is implemented using MATLAB 2018b using images from BOSSbase 1.01 database consisting of 10,000 512×512 grayscale images [18]. The evaluation criteria of any steganographic scheme is based on three steganographic principal as benchmarks: The embedding capacity (payload), imperceptibility (quality of stego image) and robustness (security) under steganalytic attacks. The performance of the method is validated against visual and statistical attack.

Visual Attack. The image quality metrics to measure the visual and structural similarity [19] of cover and stego along with the embedding time is illustrated in Fig. 5. Analysis is done with embedding rate of 10%, 30% and 50% using the three sample cover images and compared with Luo et al. AE-LSB [14] and Sanjeev et al. Fuzzy Edge [16]. We notice that the proposed algorithm outperforms the other approaches and does not leave any obvious visual artifacts with PSNR values in the range 51–62 db when edge pixels are modified by increasing % at minimal embedding time. The SSIM indicate that the stego and cover image is imperceptible by HVS and almost identical with value very close to 1. Although AE-LSB performs better than fuzzy edge with values very similar to our method, it deteriorates significantly when the modification percentage is more than 50%. Thus, for high embedding payload AE-LSB is not suitable.

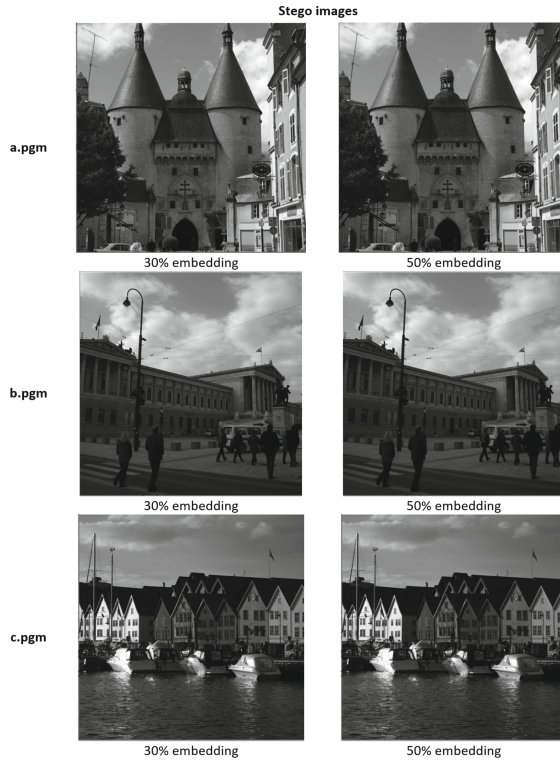


Fig. 4. Stego images post embedding.

Statistical Attack. The statistical resemblance and security of the proposed method is validated against RS steganalysis at increasing payload reaching 100% modification and pixel-wise histogram difference analysis since LSB embedding is sensitive to histogram changes. Ideally, a secure steganographic system should satisfy the relation $R_m \cong R_{-m} > S_m \cong S_{-m}$. The proposed method show ample security and RS steganalysis is ineffective in detecting the stego images as $R_m(R_{-m})$ and $S_m(S_{-m})$ is almost overlapping with large difference between $R_m(S_m)$ and S_{-m} and S_{-m} as shown in Fig. 6 at 100% embedding rate. Pixelwise histogram difference of image b at 50% modifications is illustrated in Fig. 7 (a–f). AE-LSB in Fig. 7a shows slight deviation but probability normal in Fig. 7b is identical. Fuzzy edge in Fig. 7c and 7d is severely distorted. Both the methods show vulnerability and introduce step effects in stego histograms. However, the proposed method histogram difference proves to be more robust and remains undetectable at 50% (on average with other methods) embedding rate but at 100% modification (2.0 bpp), the pixel-wise histogram analysis generates slight distortion.

Cover Image	Algorithm	Embedding rate	PSNR	SSIM	KL Divergence	Average Difference	Normalized cross-correlation	Embedding time/s	
a	AE-LSBMR [17]	10%	59.7896	0.99987	0.0001190	0.0004798	0.99999	0.484167	
		30%	54.5854	0.99894	0.0004207	0.0004043	0.99997	0.509668	
		50%	51.6332	0.99689	0.0013465	0.069687	0.99994	0.568408	
	Fuzzy Edge [19]	10%	55.9146	0.99978	0.0080911	0.0003891	0.99998	1.037153	
		30%	52.9771	0.99961	0.0002113	0.0010147	0.99996	0.878509	
		50%	50.9692	0.99561	0.0003346	0.0001831	0.99994	0.782130	
	Proposed*	10%	61.9583	0.99953	0.0007013	0.0004234	0.99999	0.043925	
		30%	57.1547	0.99869	0.0014535	0.0005989	0.99998	0.098618	
		50%	55.0049	0.99810	0.0015245	0.0014992	0.99997	0.163280	
	b	AE-LSBMR [17]	10%	58.9598	0.99962	0.0001824	0.0003700	0.99999	0.492894
			30%	53.9855	0.99597	0.0010419	0.0892910	0.99995	0.553051
			50%	51.3567	0.99537	0.0124778	0.1062000	0.99995	0.605930
Fuzzy Edge [19]		10%	57.0933	0.99968	0.0001676	0.0002288	0.99999	1.100701	
		30%	53.6819	0.99967	0.0001475	0.0001297	0.99999	0.839515	
		50%	51.7106	0.99627	0.0002465	0.0069687	0.99994	0.668408	
Proposed*		10%	61.0151	0.99931	0.0007315	0.0001716	0.99999	0.039475	
		30%	55.6736	0.99781	0.0014201	0.0003204	0.99998	0.111725	
		50%	53.8224	0.99695	0.0017519	0.0013161	0.99997	0.216715	
c		AE-LSBMR [17]	10%	59.9275	0.99989	0.0001719	0.0001792	0.99999	0.536660
			30%	54.9976	0.99930	0.0004436	0.0362946	0.99997	0.517553
			50%	52.0517	0.99712	0.0011969	0.0791269	0.99994	0.488433
	Fuzzy Edge [19]	10%	56.1040	0.99986	0.0001642	0.0009994	0.99997	0.874720	
		30%	52.3474	0.99952	0.0003541	0.0003280	0.99994	0.930843	
		50%	52.3916	0.99952	0.0003577	0.0012817	0.99993	0.869904	
	Proposed*	10%	62.0505	0.99946	0.0001551	0.0003891	0.99999	0.034800	
		30%	57.4244	0.99836	0.0010975	0.0006294	0.99998	0.101436	
		50%	55.1135	0.99758	0.0047229	0.0014648	0.99997	0.160204	

Fig. 5. Comparison of visual image quality of the proposed method at increasing embedding modification with [14] and [16].

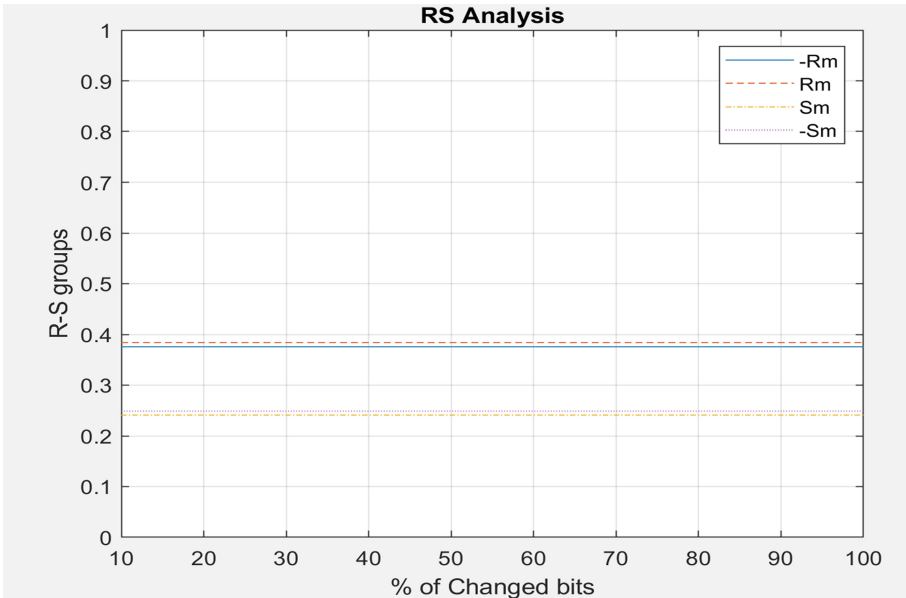


Fig. 6. RS diagram of the proposed method using cover image a. x-axis is the embedding rate and y-axis is relative change of regular and singular groups taken using mask = [0 1 1 0].

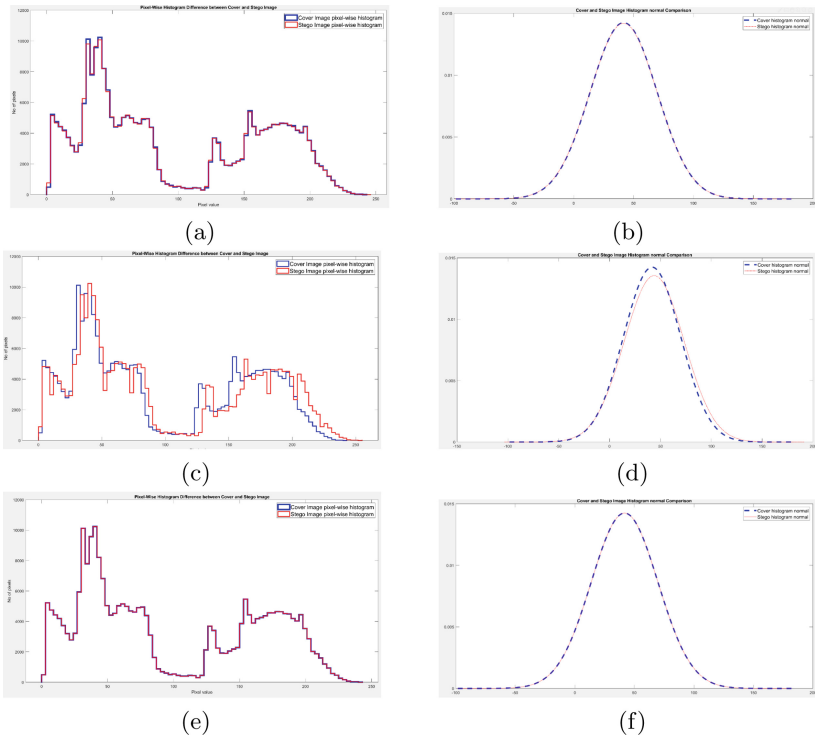


Fig. 7. Histogram analysis between stego and cover image (b.pgm) post 50% embedding procedure.

5 Conclusion and Future Work

The proposed method is a spatial image steganography algorithm based on edges identified by fuzzy inference system that takes advantage of the human visual system being less sensitive to changes in sharp intensity values. The fuzzy hybrid LSBM-LSBR is able to recognize the correct edge pixels, achieves high embedding rate by modifying the seventh and eighth bits as a pair of embedding unit, preserves the statistical and visual properties of cover images. The visual and statistical quality of the hybrid LSBM-LSBR embedding is evaluated on 2000 grayscale images from BOSSbase 1.0; the experimental results shows that the proposed method is imperceptible, offers high payload and is secure from different steganalytic attacks compared to other LSB based edge approaches. The future work will aim to remove the statistical quality deterioration at 100% modification by incorporating the higher order statistical features of the image in the design of the membership function in FIS and using a blockwise approach to reduce the complexity of the embedding procedure.

References

1. Laishram, D., Tuithung, T.: A survey on digital image steganography: current trends and challenges. *SSRN Electron. J.* (2018). <https://doi.org/10.2139/ssrn.3171494>
2. Subhedar, M., Mankar, V.: Current status and key issues in image steganography: a survey. *Comput. Sci. Rev.* 13–14 (2014). <https://doi.org/10.1016/j.cosrev.2014.09.001>
3. Islam, S., Modi, M., Gupta, P.: Edge-based image steganography. *EURASIP J. Inf. Secur.* 8 (2014). <https://doi.org/10.1186/1687-417X-2014-8>
4. Chan, C.-K., Cheng, L.M.: Hiding data in images by simple LSB substitution. *Pattern Recogn.* 37, 469–474 (2004). <https://doi.org/10.1016/j.patcog.2003.08.007>
5. Ker, A.: Steganalysis of LSB matching in grayscale images. *IEEE Sig. Process. Lett.* 12, 441–444 (2005). <https://doi.org/10.1109/LSP.2005.847889>
6. Li, B., He, J., Huang, J., Shi, Y.Q.: A survey on image steganography and steganalysis. *J. Inf. Hiding Multimed. Sig. Process.* 2, 142–172 (2011)
7. Jung, K.-H.: Comparative histogram analysis of LSB-based image steganography. *WSEAS Trans. Syst. Control* 13, 103–112 (2018)
8. Li, X., Zeng, T., Yang, B.: A further study on steganalysis of LSB matching by calibration, pp. 2072–2075 (2008). <https://doi.org/10.1109/ICIP.2008.4712194>
9. Mielikainen, J.: LSB matching revisited. *IEEE Sig. Process. Lett.* 13, 285–287 (2006). <https://doi.org/10.1109/LSP.2006.870357>
10. Tseng, H.-W., Leng, H.-S.: A steganographic method based on pixel-value differencing and the perfect square number. *J. Appl. Math.* (2013). <https://doi.org/10.1155/2013/189706>
11. Chang, K.-C., Chang, C.-P., Huang, P., Tu, T.-M.: A novel image steganographic method using tri-way pixel-value differencing. *J. Multimed.* 3 (2008). <https://doi.org/10.4304/jmm.3.2.37-44>
12. Al-Dmour, H., Al-Ani, A.: A steganography embedding method based on edge identification and XOR coding. *Expert Syst. Appl.* 46 (2015). <https://doi.org/10.1016/j.eswa.2015.10.024>
13. Dhargupta, S., Chakraborty, A., Ghosal, S., Saha, S., Sarkar, R.: Fuzzy edge detection based steganography using modified Gaussian distribution. *Multimed. Tools Appl.* <https://doi.org/10.1007/s11042-018-7123-x>
14. Luo, W., Huang, F., Huang, J.: Edge adaptive image steganography based on LSB matching revisited. *IEEE Trans. Inf. Forensics Secur.* 5, 201–214 (2010). <https://doi.org/10.1109/TIFS.2010.2041812>
15. Hiary, H., Sabri, K.E., Mohammed, M., Al-Dhamari, A.: A hybrid steganography system based on LSB matching and replacement. *Int. J. Adv. Comput. Sci. Appl.* 7 (2016)
16. Kumar, S., Singh, A., Kumar, M.: Information hiding with adaptive steganography based on Novel Fuzzy edge identification. *Defense Technol.* 15, 162–169 (2019)
17. <http://in.mathworks.com/help/fuzzy/fuzzy-logic-image-processing.html>
18. <http://dde.download/ImageDatabase.binghamton.edu>
19. <http://in.mathworks.com/matlabcentral/fileexchange/image-quality-measures>



Universum Based Nonparallel Kernelized Extreme Learning Machine for Binary Classification

Akansha Mangal^(✉) and Sanyam Shukla

Maulana Azad National Institute of Technology, Bhopal, India

Abstract. Extreme Learning Machine (ELM) is proved to be fast, cost-effective and efficient for solving the binary classification problems. On the other hand, learning with the universum samples is also considered useful as these samples are incorporated as the prior information with the classifier. In our proposed work, we design the novel Universum based nonparallel kernelized Extreme Learning Machine (UNPKELM) which is a variant of Extreme Learning Machine. The two ELMs are learned and trained with the two nonparallel hyperplanes and hence make a robust classifier. This work also presents the mathematical formulation of UNPKELM and equations are derived to determine the output weights. Our proposed classifier UNPKELM is evaluated using 10 benchmark datasets and its performance (G-Mean) is compared to KELM, WKELM, CCR-KELM which verifies its effectiveness.

Keywords: Extreme Learning Machine · Universum · Classification · Nonparallel hyperplane

1 Introduction: Universum Based Kernelized Extreme Learning Machine with Non-parallel Hyperplanes

Extreme Learning Machine [4, 5] is one of the types of single hidden layer feed forward network that is used for solving classification problems. It can be used for the binary as well as multiclass classification problems. A Number of variants [9, 10] of ELM has been proposed to improve the classification accuracy. In the following Sect. 1.1, ELM and its variants are summarized and universum learning is discussed in Sect. 1.2. The rest of the sections describe the proposed work in detail.

1.1 ELM and Its Variants

Extreme Learning Machine: It is an efficient and efficacious learning algorithm used for binary and multiclass classification problems. Suppose the training dataset is $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ where x_i is the i th input sample and y_i is its corresponding target class. N is the total number of samples in the training dataset.

ELM introduces the weights lie between the input and the hidden layer which are assigned randomly. Once assigned, these weights will not change during the training phase. Moore- Penrose (MP) generalized pseudoinverse is used to compute the weights that lie between the hidden and the output layer. ELM uses various activation functions but the commonly used activation function is the Sigmoidal activation function. With the use of kernel functions like linear kernel, gaussian kernel etc., data is mapped from input space to feature space, hence generalization performance gets better in this approach than Sigmoidal based ELM.

Weighted Extreme Learning Machine: Weighted Extreme Learning Machine [16] uses weights of the training instances amid the training phase to cope up with the class imbalance problem. Two weighting schemes have been proposed for WELM. In the first scheme, $1/t_k$ weight is assigned to the minority as well as majority training samples where k is the total number of samples in k th class. In the second scheme, $0.618/t_k$ weight is assigned to the majority class samples and $1/t_k$ weight is assigned to the minority class samples. It is experimentally proven that WELM performs better than the traditional Sigmoidal based ELM and kernelized ELM for most of the datasets.

Class-Specific Cost Regulation Extreme Learning Machine: CCRELM [14] employs class specific regulation cost to handle the class imbalance problem. The two regularization parameters, C^+ and C^- assign to the positive minority and negative majority samples respectively. Kernel based CCR-ELM is also proposed and obtained the more satisfactory performance.

1.2 Universum Learning

The abstract idea of universum samples was introduced [13] to find better decision boundary. Universum samples are the samples that do not belong to any of the target classes. These universum points incorporate prior information to the classifier. Adding the universum samples has been proved to be useful in supervised learning. For example, ‘3’ vs ‘6’ classification problem for handwritten digit recognition dataset, the other digits 0, 1, 2, 4, 5, 7, 8 and 9 can be taken as universum samples. These universum samples do not belong to the target classes. The learning by universum samples can be deemed as learning by contradiction. Various classifiers [8, 11, 15] have incorporated the universum samples to achieve better performance. In this paper, Universum samples are created by using the Imbalanced Modified Universum (IMU) method proposed by [12].

$$x_u^* = \left(\frac{N_n}{N_n + N_p} \right) * x_i + \left(\frac{N_p}{N_n + N_p} \right) * x_j \quad (1)$$

Here, N_p and N_n represent the total number of positive and negative samples respectively. x_i and x_j represents the negative and positive class sample respectively.

1.3 Proposed Work

Traditional ELM learns with one hyperplane to separate different classes in the feature space. Here, we propose a novel Universum based nonparallel Extreme learning Machine (UNPKELM) by incorporating universum samples with the training dataset. In our proposed work, two ELMs are trained with two nonparallel hyperplanes by tuning the regularization parameter. This work includes the following points:-

1. Universum samples are created by using Eq. 1.
2. Two regularization parameters C_1 and C_2 are tuned to train the ELMs with the nonparallel hyperplanes.

Mathematical Formulation of UNPKELM: The mathematical formulation of UNPKELM is given below:

$$\begin{aligned} \text{Minimize: } & \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \sum_{i=1}^N \xi_i^2 + \frac{1}{2} C_u \sum_{j=1}^U \psi_j^2 \\ \text{Subject to: } & h(x_i)\beta = t_i - \xi_i, \quad i = 1, 2, \dots, N \\ & h(x_u^*)\beta = t_u - \psi_u, \quad u = 1, 2, \dots, U \end{aligned} \quad (2)$$

Here, U and N are the number of universum and training input samples. $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ and its output by $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$ where n and m are the input and output neurons respectively. $X_u = x_1^*, x_2^* \dots x_U^*$ are the universum samples. β is the output weight vector lies between the hidden and the output layer. $\|\xi\|^2$ and $\|\psi\|^2$ are the least square errors for the target class samples and the universum samples. $h(x_i)$ is the output of the hidden layer of the i th sample. ψ and ξ are the error vectors. By introducing the Lagrangian multipliers to the above equation, the dual problem is obtained

$$L = \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \sum_{i=1}^N \xi_i^2 + \frac{1}{2} C_u \sum_{u=1}^U \psi_u^2 - \sum_{i=1}^N \alpha_i (h(x_i)\beta - t_i + \xi_i) - \sum_{u=1}^U \eta_u (h(x_u^*)\beta + \psi_u)$$

The Karush-Kuhn-Kucher (KKT) sufficient and necessary optimality conditions for β, ξ, ψ, α and η are given by

$$\frac{\delta L}{\delta \beta} = \beta - \sum_{i=1}^N \alpha_i (h(x_i)) - \sum_{u=1}^U \eta_u (h(x_u^*)) = 0$$

$$\beta = \mathbf{H}^T \alpha + \mathbf{U}^T \eta \quad (3)$$

$$\frac{\delta L}{\delta \xi_i} = C \xi_i - \alpha_i = 0, \quad i = 1, 2, \dots, N \quad (4)$$

$$\frac{\delta L}{\delta \psi_u} = C_u \psi_u - \eta_u = 0, \quad u = 1, 2, \dots, U \quad (5)$$

$$\frac{\delta L}{\delta \alpha_i} = h(x_i) \beta - t_i + \xi_i = 0 \quad (6)$$

$$\frac{\delta L}{\delta \eta_i} = h(x_u^*) \beta - t_u + \psi_u = 0 \quad (7)$$

By combining Eqs. (3), (4), (5), (6) and (7):

$$\mathbf{T} - \mathbf{H}[\mathbf{H}^T \quad \mathbf{U}^T] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\eta} \end{bmatrix} = \frac{\boldsymbol{\alpha}}{C} \quad (8)$$

$$\mathbf{T}_u - \mathbf{U}[\mathbf{H}^T \quad \mathbf{U}^T] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\eta} \end{bmatrix} = \frac{\boldsymbol{\eta}}{C_u} \quad (9)$$

Then substituting Eq. (8) and (9) into Eq.(3), the solution can be determined as follows:

$$\boldsymbol{\beta} = [\mathbf{H}^T \quad \mathbf{U}^T] \left(\begin{bmatrix} \frac{I_{N \times N}}{C} & 0 \\ 0 & \frac{I_{U \times U}}{C_u} \end{bmatrix} + \begin{bmatrix} \mathbf{H} \\ \mathbf{U} \end{bmatrix} [\mathbf{H}^T \quad \mathbf{U}^T] \right)^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{T}_u \end{bmatrix} \quad (10)$$

For ELM 1, the output weight vector between the hidden and the output layer i.e. $\boldsymbol{\beta}_1$ is given as follows:

$$\boldsymbol{\beta}_1 = [\mathbf{H}^T \quad \mathbf{U}^T] \left(\begin{bmatrix} \frac{I_{N \times N}}{C_1} & 0 \\ 0 & \frac{I_{U \times U}}{C_u} \end{bmatrix} + \begin{bmatrix} \mathbf{H} \\ \mathbf{U} \end{bmatrix} [\mathbf{H}^T \quad \mathbf{U}^T] \right)^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{T}_u \end{bmatrix} \quad (11)$$

Similarly, $\boldsymbol{\beta}_2$ is given as follows:

$$\boldsymbol{\beta}_2 = [\mathbf{H}^T \quad \mathbf{U}^T] \left(\begin{bmatrix} \frac{I_{N \times N}}{C_2} & 0 \\ 0 & \frac{I_{U \times U}}{C_u} \end{bmatrix} + \begin{bmatrix} \mathbf{H} \\ \mathbf{U} \end{bmatrix} [\mathbf{H}^T \quad \mathbf{U}^T] \right)^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{T}_u \end{bmatrix} \quad (12)$$

UNPKELM employs Gaussian kernel function $K(\cdot, \cdot)$ which maps data from input space to feature space. Given a new sample \mathbf{x} , the class label can be predicted with two ELMs for UNPKELM by using the Eq. (10) is given as follows:

$$\mathbf{f}_1(\mathbf{x}) = \text{sign} \begin{bmatrix} K(x, q_1) \\ \vdots \\ K(x, q_{N+U}) \end{bmatrix}^T \left(\begin{bmatrix} \frac{I_{N \times N}}{C_1} & 0 \\ 0 & \frac{I_{U \times U}}{C_u} \end{bmatrix} + \begin{bmatrix} \mathbf{K}(\mathbf{x}_N, \mathbf{x}_N) & \mathbf{K}(\mathbf{x}_N, \mathbf{y}_U) \\ \mathbf{K}(\mathbf{y}_U, \mathbf{x}_N) & \mathbf{K}(\mathbf{y}_U, \mathbf{y}_U) \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{T}_u \end{bmatrix} \quad (13)$$

$$\mathbf{f}_2(\mathbf{x}) = \text{sign} \begin{bmatrix} K(x, q_1) \\ \vdots \\ K(x, q_{N+U}) \end{bmatrix}^T \left(\begin{bmatrix} \frac{I_{N \times N}}{C_2} & 0 \\ 0 & \frac{I_{U \times U}}{C_u} \end{bmatrix} + \begin{bmatrix} \mathbf{K}(\mathbf{x}_N, \mathbf{x}_N) & \mathbf{K}(\mathbf{x}_N, \mathbf{y}_U) \\ \mathbf{K}(\mathbf{y}_U, \mathbf{x}_N) & \mathbf{K}(\mathbf{y}_U, \mathbf{y}_U) \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{T}_u \end{bmatrix} \quad (14)$$

Here, $Q \in [X \ X_u]^T$ and $q_i \in Q$ where $i = 1, 2, \dots, N + U$.
So, $f(\mathbf{x}) = \text{arg} : \max(f_1(x), (f_2(x)))$ with respect to similarity.

Notes and Comments. The target class (t_u) is considered as 0 for the universum samples for the classification problems (Fig. 1).

1.4 Proposed Algorithm of UNPKELM

Algorithm UNPKELM

Input: The training dataset: $\{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, 2, \dots, n\}$,

Output: UNPKELM Model for Classification problem

1. **Procedure** (Universum based nonparallel KELM)
 2. Universum samples (UNS) are generated by using Eq. 1
 3. Generate the new training dataset (Q) by adding the Universum Samples (UNS) to the training dataset (X).
 4. Initialize the kernel parameter σ and the regularization parameters C_1, C_2
 5. Generate the kernel matrix $\mathbf{K}(\mathbf{Q}, \mathbf{Q})$ by using Q dataset.
 6. Compute the weights between the hidden and the output layer β_1 & β_2 by using following Eq. (11) and (12)
 7. Compute $f(x)$ by taking $arg : \max$ of $f_1(x)$ and $f_2(x)$ by using Eq. (13) and (14)
 8. **return** $f(x)$
 9. **end procedure**
-

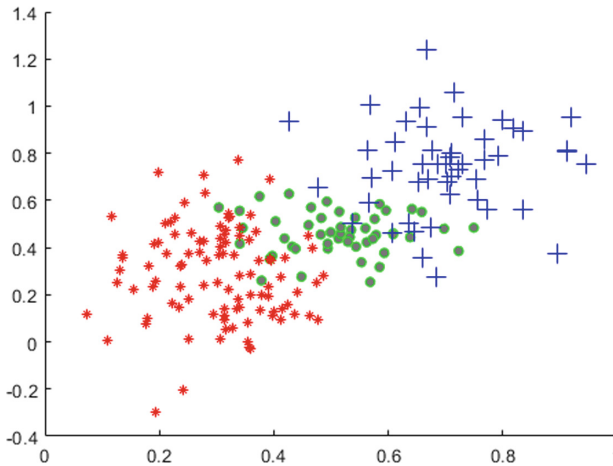


Fig. 1. Illustration of the UNPKELM. The blue + represents the positive point, red * represents the negative point and green o represents the universum point.

1.5 Computational Cost Analysis

In UNPKELM, the output weight i.e. β can be computed by using Eq. (10) with the kernel function. Here, N is the number of training vectors and C is the number of corresponding class labels. D represents the number of input neurons of same dimensionality as a input vector and C represents the number of output neurons which is equal to the number of classes whereas U represents the number of universum samples. The computational complexities for evaluating the terms $K(\mathbf{x}_N, \mathbf{x}_N)$, $K(\mathbf{x}_N, \mathbf{y}_U)$, $K(\mathbf{y}_U, \mathbf{x}_N)$ and $K(\mathbf{y}_U, \mathbf{y}_U)$ have order of $O(DN^2)$, $O(NDU)$, $O(NDU)$ and $O(DU^2)$ respectively. The Computational complexity to determine the Kernel matrix is equal to $D(N + U)^3$. The computational complexities for evaluating the terms \mathbf{T} and \mathbf{T}_u have order of $O(CN)$ and $O(CU)$ respectively. The computational cost of the output weight matrix, β given in [7] and [6] is given as follows.

$$\begin{aligned} & O(DN^2 + DU^2 + 2NDU + D(N + U)^3 + (N + U)^3 + C(N + U) + C(N + U)^2) \\ \Rightarrow & O(D(N + U)^2 + D(N + U)^3 + (N + U)^3 + C(N + U) + C(N + U)^2) \\ \Rightarrow & O((D + C)(N + U)^2 + (D + 1)(N + U)^3 + C(N + U)) \end{aligned} \quad (15)$$

1.6 Experimental Setup

The performance of our proposed work is evaluated on 10 benchmark datasets downloaded from the UCI Machine Learning Repository [3] and KEEL Data repository [1,2]. These datasets are accessible in five cross-validation format at the KEEL dataset repository. Gaussian kernel function is used for the data mapping from input space to the feature space. For the first ELM, the regularization parameter C_1 is equal to C_u and for the second ELM, the regularization parameter C_2 is equal to C_u . The width of kernel parameter (σ), regularization parameter (C_1 & C_2) are assigned by conducting a grid search on $\{2^{-18}, 2^{-16}, \dots, 2^{18}, 2^{20}\}$ and $\{2^{-18}, 2^{-16}, \dots, 2^{18}, 2^{50}\}$ respectively. Number of Universum samples are generated in the input space and taken as half of the number of training input samples.

The performance classifier in terms of geometric mean has been reported. The geometric mean (G-mean) is the root of the product of classwise true positive rate (TPR) i.e. specificity.

1.7 Result Analysis

The experimental results are computed for KELM, WKELM, CCR-KELM and UNPKELM for the ten datasets. It is worth noticing from Table 1 that UNPKELM outperforms KELM, WKELM and CCR-KELM for most of the datasets. Box plot is represented in Fig. 2 for different classifiers. A t-test is also used to determine the significant difference between UNPKELM and the other classifiers. The results of t-test in terms of (h,p) value are given below:-

1. KELM vs UNPKELM - 1, 0.0062
2. WKELM vs UNPKELM - 1, 0.0498
3. CCR-KELM vs UNPKELM - 1, 0.0108

Table 1. Performance evaluation for datasets in terms of G-mean. (Best result on each dataset is highlighted as bold)

Dataset	KELM	WKELM	CCR-KELM	UNPKELM
glass0	85.35	85.65	85.35	86.53
glass1	77.48	80.35	76.60	81.05
glass5	93.16	96.51	89.71	97.51
glass6	92.60	94.04	93.77	95.97
ecoli2	94.31	94.09	92.90	94.95
iris0	100	100	100	100
haberman	57.23	66.26	59.71	66.21
shuttle-C2vsC4	94.14	100	100	100
ecoli01vs5	88.36	91.10	88.36	95.48
vowel	100	100	100	100

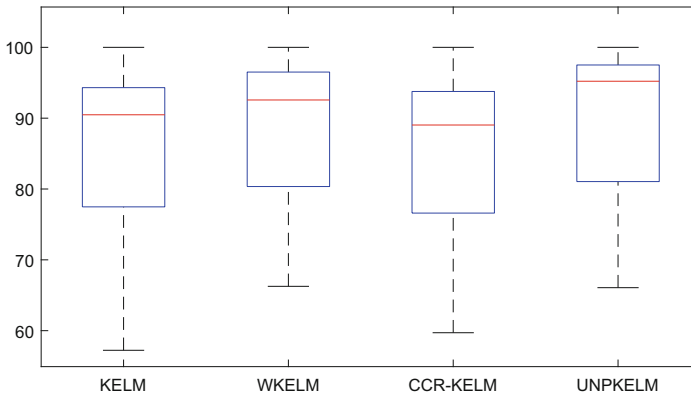


Fig. 2. Box-plot representation of different classifiers

1.8 Conclusion and Future Work


KELM, WELM and CCR-ELM train with parallel hyperplane. In this paper, we design a novel classifier UNPKELM, which incorporates universum samples. Further, the two ELMs are trained with the non-parallel hyperplanes which is found to perform better. It also uses universum samples to find a better decision boundary leading to better performance. Universum samples generation being the crucial and decisive criteria can enhance the performance of the classifier. So, future work includes more exhaustive evaluation of the proposed classifier with other possible techniques for universum generation.

References

1. Alcalá-Fdez, J., et al.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Multiple-Valued Logic Soft Comput.* **17**, 255–287 (2010)
2. Alcalá-Fdez, J., Sánchez, L., García, S., Del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas Santos, V., Fernández, J.C., Herrera, F.: Keel: A software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.* **13**, 307–318 (2009)
3. Dua, D., Graff, C.: UCI machine learning repository (2017)
4. Huang, G., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **42**(2), 513–529 (2012)
5. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1), 489–501 (2006)
6. Iosifidis, A., Gabbouj, M.: On the kernel extreme learning machine speedup. *Pattern Recogn. Lett.* **68**(Part 1), 205–210 (2015)
7. Iosifidis, A., Tefas, A., Pitas, I.: On the kernel extreme learning machine classifier. *Pattern Recogn. Lett.* **54**(Supplement C), 11–17 (2015)
8. Qi, Z., Tian, Y., Shi, Y.: A nonparallel support vector machine for a classification problem with universum learning. *J. Comput. Appl. Math.* **263**, 288–298 (2014)
9. Raghuvanshi, B.S., Shukla, S.: Class-specific extreme learning machine for handling binary class imbalance problem. *Neural Netw.* **105**, 206–217 (2018)
10. Raghuvanshi, B.S., Shukla, S.: Class-specific kernelized extreme learning machine for binary class imbalance learning. *Appl. Soft Comput.* **73**, 1026–1038 (2018)
11. Sinz, F., Chapelle, O., Agarwal, A., Schölkopf, B.: An analysis of inference with the universum. In: *Advances in Neural Information Processing Systems 20: Proceedings of the 2007 Conference*, vol. 20, pp. 1369–1376 (2008)
12. Wang, Z., Hong, S., Yao, L., Li, D., Du, W., Zhang, J.: Multiple universum empirical kernel learning. *Eng. Appl. Artif. Intell.* **89**, 103461 (2020)
13. Weston, J., Collobert, R., Sinz, F., Bottou, L., Vapnik, V.: Inference with the universum, pp. 1009–1016 (2006)
14. Xiao, W., Zhang, J., Li, Y., Zhang, S., Yang, W.: Class-specific cost regulation extreme learning machine for imbalanced classification. *Neurocomputing* **261**, 70–82 (2017). *Advances in Extreme Learning Machines (ELM 2015)*
15. Xu, Y., Chen, M., Li, G.: Least squares twin support vector machine with universum data for classification. *Int. J. Syst. Sci.* **47**(15), 3637–3645 (2016)
16. Zong, W., Huang, G.B., Chen, Y.: Weighted extreme learning machine for imbalance learning. *Neurocomputing* **101**, 229–242 (2013)



A New Image Watermarking Scheme Based on Block Conversion and DWT-SVD Approach

Ranjeet Kumar Singh 

Department of Computer Science and Engineering, DIT University, Dehradun, India

Abstract. The authors propose a digital image watermarking mechanism to improve data security, robustness, and capacity by combining singular value decomposition with the discrete wavelet transformation. In the proposed approach, an image is decomposed horizontally as well as vertically into their low and high sub-frequency bands, respectively by using DWT. A horizontal low-frequency sub-band is selected and converted into an $M \times M$ block size. After that, the watermark images are embedded into each block using the SVD approach. After the completion of this process, where the watermarked images were subjected to various kinds of attacks such as cropping, rotation, sharpness, noise, contrast adjustment, filtering etc., the inserted watermark image was recovered from the cover image. The quality of recovered watermarks was compared based on correlation coefficients. The proposed algorithm's robustness was found to be comparatively better than the previously existing methods. Tables show the results.

Keywords: DWT · SVD · Correlation coefficient · PSNR

1 Introduction

In the era of the rapid growth of network technology, digital data can be shared with no loss of data quality, low cost and expected fast delivery. Data authentication and security have become a challenging issue because of the users' inadequate awareness of the ownership of intellectual property [1]. Several solutions are available for data protection, such as steganography, cryptography, etc. One of the better solutions to address such issues is watermarking digital content. Digital image watermarking is a means to protect digital image-based information from unlawful copy and manipulation. Digital image watermarking techniques today are a lot more robust, sophisticated, and advanced. By employing an appropriate watermarking scheme, it is possible to recover the information embedded in the original image or detect its occurrence in the multimedia element for different use including, access control, copyright protection, and monitoring digital broadcasts [2]. Both the watermarking system and the steganography are learned under the information hiding approach, i.e., both are used to hide secret data into the original data. However, there is an understated difference between the two. Steganography takes care of the undisclosed information. In case the existence of undisclosed data is open, steganography fails. But in watermarking techniques, the existence of undisclosed data may be identified [3]. In watermarking, three terms are normally used, viz. watermark

image, cover image, and watermarked image. A digital watermark image is a signal which is inserted into the original image, also known as the cover content, that can be reproduced in the upcoming stage for the proof of identity, digital document ownership and authentication. Based on the nature of the cover content, digital watermarking can be categorized in different forms such as image, audio, and video watermarking. In the sections to follow, discussion on the works based mostly on the block-based image watermarking approach, the proposed algorithm and the experimental result and analysis are given.

2 Related Works

In [4], the authors proposed a digital data watermarking technique that was constructed on logistic and singular value decomposition (SVD). In the first stage, the original image was distributed into 8×8 blocks size, with the grayscale watermark image being scrambled using logistic mapping. After that, the authors applied SVD on each block to obtain the U, S and VT and chose the singular value components for embedding the watermark image. In [5], a combined approach of SVD-based and DWT-based watermarking scheme was explained. In this mechanism, the authors modified the more orthodox SVD to develop a shuffled SVD (SSVD). The original image was divided into four sub-bands, i.e., vertically low, and high-frequency bands and horizontally low and high-frequency bands (LL, HL, LH, and HH) by applying DWT. The LL frequency band is further divided into $N \times N$ blocks size. The principal components of the watermark are embedded in blocks of the LL frequency band using SSVD. Farzad Golshan et al. [6] used the particle swarm optimization and genetic programming as a hybrid technique to create a tradeoff between imperceptibility and robustness. The authors used intelligent methods in the DCT_DWT_SVD domain. At first, all the cover images are transformed into the DCT domain. A section of the DCT matrix is taken to be divided into four sub bands using discrete wavelet transform. Subsequently, the S component of every sub band is formed by the S component of the image to insert the image watermark.

Sun et al. [7] proposed a method where 8×8 block size was considered. To begin with, the host image was split up into blocks, and SVD was applied on each block to recover the S component. By applying certain mechanisms on the S component, a binary watermark was inserted in the image, and by using the reverse procedure, the extraction was done. Similarly, Liu et al. [8] came up with a method that was created using DCT and SVD techniques. The host image was disintegrated into independent blocks with a size of 8×8 , and DCT was applied on each block. It was followed by SVD to regenerate the watermarked image. These methods require more computation as they consider the whole image for block division.

The proposed approach in this study advances these methods using DWT and application of SVD in LL-band, which involves less computation. Rao et al. [9] made use of SVD, DWT, DCT and Particle Swarm Optimization (PSO) to propose a hybrid watermarking mechanism. It is desirable to find the optimum scaling factor for having a transparent, robust, and imperceptible watermark scheme. PSO is commonly used for this purpose.

Chung rong et al. [10] proposed a method that uses 8×8 blocks for watermarking. Before the block division, another method was applied to the host image called Moore-model cellular automata scrambling. The image was scrambled to obtain more security for the watermarking procedure. DCT was applied on the independent 8×8 blocks to get the low-frequency coefficients, followed by SVD application to embed the watermark into these blocks. The embedding of the blocks was done with a frequency α and the same frequency is used for the extraction purpose.

Liu et al. [4], in their study, illustrated a method using a grayscale watermark image and a host image. Blocks of size 8×8 were used for the watermarking purpose. Scrambling of the watermark was done before block decomposition. SVD is applied to each block of watermark and host image to obtain the watermarked image, and through the reverse process recovery of watermarking is done. This method applied to grayscale images only. The method proposed in this paper overrides this method by making use of DWT and color images.

Chin-Chin Lai [11] introduced a block-based scheme of watermarking by using the characteristics of human visuals. In this method, the host image was divided into non-overlapping 8×8 blocks. Some blocks with special properties were selected for the embedding process. The blocks with the least sum of edge and visual entropies were selected. The selection of blocks for the embedding process depends on the watermark image size. The size of the watermarked image and the number of blocks selected for embedding are directly proportional. DCT was applied on these select blocks, followed by the SVD technique, and after some examination on the U matrix, the binary watermark was embedded. Extraction was done using a similar method. The disadvantage of this method is that it only uses a binary watermark image for embedding.

Chang et al. [12] explained a method where a grayscale image was the host image, and a binary image sufficed as the watermark image. The host image was broken down into several independent blocks of the same size and SVD was applied. From the available S blocks, few were selected according to some computations and the U blocks related to those S blocks were selected for the embedding process. Embedding was done and extraction was done through predictions. Abdallah et al. [13] introduced a watermarking method that split every sub-band of the DWT component into different blocks. On each block, a fast Hadamard transform, followed by SVD, was performed. By calculating the result, the embedding of the watermark was done. The extraction procedure was implied in the embedding procedure.

Xing et al. [14] explained a method that divides the RGB components of the image into several blocks of size equal to a few powers of two. Before decomposing the image into RGB components, Arnold transform was applied to obtain robustness against attacks. The blocks were then subjected to SVD and S component was taken for the embedding process, and the extraction is done through the reverse of the embedding process.

Golea et al. [15] introduced a technique where they divided all the three RGB components into blocks of the same size and applied SVD on every block. After that, they modified the S components by embedding the watermark in the same way and creating the watermarked image. Similarly, the watermark was extracted from the watermarked image. Due to the decomposition of the image into the RGB components, this approach

is not suitable for some malicious attacks like rescaling, blurring and winner, median filter etc. But the proposed approach in this paper is reasonably better for these types of malicious attacks.

Mathew [16] introduced a block-based watermarking scheme that employed SVD as the main technique. This method made use of U and V components of SVD. The method divided the host image into different independent blocks, followed by an SVD calculation on each block to get U and V components. Using certain calculations embedding was done on the first columns of U and V components. The extraction was like an embedding procedure. Kansal et al. [17] proposed a method using DWT, DCT and SVD on blocks of images. The blocks were created from the LL sub-band of the host image obtained from DWT. These blocks were then subjected to SVD and watermarking was done on the S component.

The proposed approach is programmed to be a less complex approach.

3 Proposed Method

Embedding Algorithm

The watermark embedding process is divided into ten steps. As a prerequisite for the embedding process, the cover object (original image) and the watermark must be of the same dimension. The algorithm is pronounced as follows:

Step 1: Selection of the original image.

Step 2: Decomposition of the original image into four sub-bands using DWT:

$$LL1, LH1, HL1, HH1 = \text{dwt2}(\text{original image})$$

Step 3: Choosing the LL1 subband and dividing it into 8×8 blocks.

Step 4: Breaking down every 8×8 size blocks into its RGB component.

Step 5: Applying SVD to all the RGB components of the original image block.

$$[uh1 \ sh1 \ vh1] = \text{svd}(\text{original image block})$$

Step 6: Taking the QR Code as a watermark image.

Step 7: Repeating steps 2–4 on the watermark image.

Step 8: Applying SVD to all the RGB components of the QR code watermark image.

$$uw1 \ sw1 \ vw1 = \text{svd}(\text{watermark blocks})$$

Step 9: Embedding the SVs of the RGB components of every block of the watermark into their corresponding blocks of the original image with a factor α .

$$sh = sh1 + \alpha \times sw1$$

Step 10: Recreating the new blocks by using the new sh and u and v components of each block of the RGB component of the original image.

Step 11: Combining the RGB components of each block.

Step 12: Combining all the blocks of the original image.

Step 13: Applying inverse DWT to the modified sub-band and created the watermarked image.

Extraction Procedure

Step 1: Selecting the watermarked image.

Step 2: Decomposing the watermark image into four sub-bands using DWT.

$$LL1, LH1, HL1, HH1 = \text{dwt2}(\text{watermarked image})$$

Step 3: Choosing the LL1 subband and dividing it into 8×8 blocks.

Step 4: Converting all the 8×8 size blocks into their RGB components.

Step 5: Applying SVD to the RGB components of the watermarked image.

Step 6: Taking a QR Code as the watermark image.

Step 7: Repeating steps 2–4 for the watermark image.

Step 8: Applying SVD to all the RGB components of the QR code image.

$$uw1 \ sw1 \ vw1 = \text{svd}(\text{watermark blocks})$$

Step 9: Selecting the original image.

Step 10: Performing the steps 2–4 on the original image.

Step 11: Applying SVD to all the RGB components of the original image.

$$uh \ sh \ vh = \text{svd}(\text{original blocks})$$

Step 12: Extracting the SVs of the RGB component of all the blocks of the watermark from the conforming blocks of the watermarked image with a factor α .

$$sw1 = (sh1 - sh)/\alpha$$

Step 13: Recreating the new block by using the new sw, u and v component of each block of the RGB component of the watermark image.

Step 14: Combining the RGB components of every block.

Step 15: Every block of the watermark image is amalgamated.

Step 16: Reproduced the watermark image by applying inverse DWT to the modified sub-band.

4 Analysis and Result

This new watermarking technique is developed on the MATLAB (R2015a) platform. The results are tabulated below for a comparison between the proposed scheme and the existing ones ([15]). For the experimentation, the standard “Lena” image was taken as the host image and a sample QR code as the watermark. Every image is a 256×256 pixels sized image. The correlation coefficient (CC) values were taken for result

comparison. A correlation coefficient measures the similarity between two images. CC value is measured by using the following equation:

$$CC = \frac{\sum_{i=1}^m \sum_{j=1}^n w(i, j) * w'(i, j)}{\sum_{i=1}^m \sum_{j=1}^n w^2(i, j)} \tag{1}$$

where W & w' are original and extracted images respectively (Figs. 1 and 2).

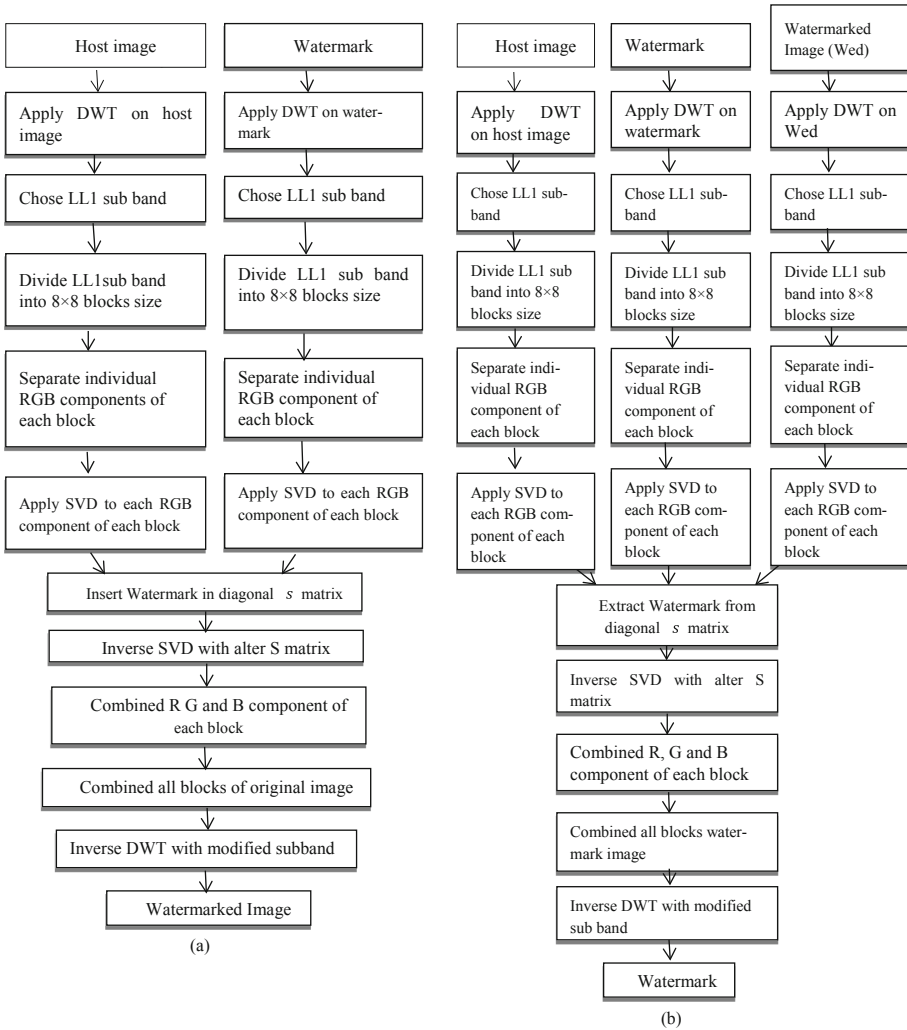


Fig. 1. (a) Watermark embedding procedure (b) Watermark extraction procedure

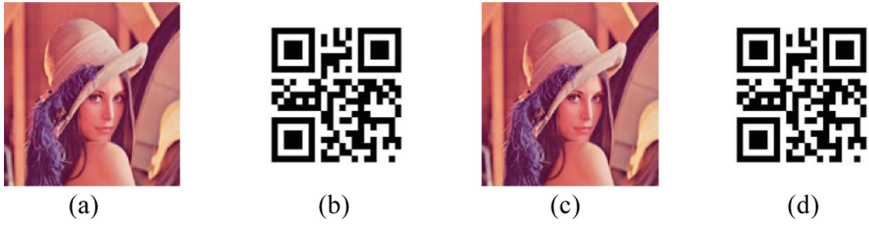


Fig. 2. (a) Lena (b) QR code (c) Watermarked image (d) Recovered watermark

Initially, in the absence of attacks, the results were 1 for the original QR code and the extracted QR code. Results obtained post-exposure to various attacks like rescaling, winner filtering, median filtering, Gaussian filtering, blurring, noise addition, rotation, contrast adjustment, flipping etc., are tabulated in Table 1 for comparison with the results of the already available work.

Table 1. Performance comparison between [12] and proposed method against several attacks for 8×8 block size

Type of attack	Parameters	8×8			
		[15]	Proposed scheme		
Rescaling	0.5	0.5917	0.9781		
	2	0.9537	0.9942		
Winner filter	3×3	0.7131	0.9771		
Median filter	3×3	0.4693	0.9767		
Blurring	Radius = 0.1	0.9994	0.9973		
	Radius = 1.0	0.8469	0.9762		
Salt & pepper	Density = 0.002	0.9929	0.9880		
	Density = 0.008	0.9634	0.9699		
Gaussian noise	$M = 0.0, v = 0.001$	0.9497	0.9802		
	$M = 0.1, v = 0.001$	0.9301	0.8297		
Rotation	Angle = 0.1	0.9732	0.9605		
	Angle = 0.5	0.8139	0.6533		
Contrast adjust	Default parameters	0.9994	0.9967		
Gaussian filter	3×3	0.9994	0.7344		
	5×5	0.9994	0.5759		
Flipping	Vertical	0.5375	0.3862		
	Horizontal	0.2857	0.3314		
	Total	0.4948	0.3283		
Type of attacks		[4]	[11]	[18]	Proposed scheme
Salt and Pepper Noise (0.001)		0.9397	0.7468	0.9354	0.9889
Salt and Pepper Noise (0.002)		0.8861	0.6058	0.9154	0.9880
Gaussian Noise (0.001)		0.6820	0.3545	0.9354	0.9802
Gaussian Noise (0.002)		0.3553	0.1257	0.7772	0.9605
Median Filter (3×3)		0.8349	0.5432	0.8543	0.9767

(A) Filtering: Digital image filtering is normally required to recover the image quality or to repair the digital image, which is degraded by certain noise. The recovered watermarks post-application of the 3×3 winner, median, and Gaussian filter are shown in Fig. 3. It can be observed that after using the filters, images get corrupted and a lot of information is lost. But the recovered watermark is still detectable.



Fig. 3. (a) 3×3 Winner Filter (b) 3×3 Median Filters (c) 3×3 Gaussian Filter (d) 5×5 Gaussian Filters

(B) Addition of noise: Any unwanted signal or function is called as noise. A popular method to test the robustness of a watermark is to add noise to it. Usually, the noise will distort and degrade the quality of the image. The addition of a digital noise will degrade the watermark information and pose problems in the recovery of the watermark image. As shown in Fig. 4, adding Gaussian, salt and pepper and speckle noise with variances 0.004 makes the recovered watermark noisy, but still recognizable.



Fig. 4. (a) Gaussian noise (b) Salt and pepper noise (c) Speckle noise

(C) Rotation: It refers to the rotation of the image following the user-specified angle. The new method proposed in this paper can recover the watermark image after enough rotations. In this case, the watermark was recovered from 0.1 and 0.5 angles. The results are shown in Fig. 5.



Fig. 5. (a) Rotation with 0.1 (b) Rotation with 0.5

(D) General image processing attacks: In the proposed method, the quality of recover watermark for contrast adjustment with default parameter is also tested by blurring with radius 0.1 and 1.0, flipping (vertical, horizontal, total). Figure 6 and Fig. 7 depict the results of flipping and blurring, respectively. The correlation coefficients of all the recovered watermarks after all malicious attacks are shown in Table 1.



Fig. 6. (a) Horizontal flipping (b) Total flipping(c) Vertical Flipping

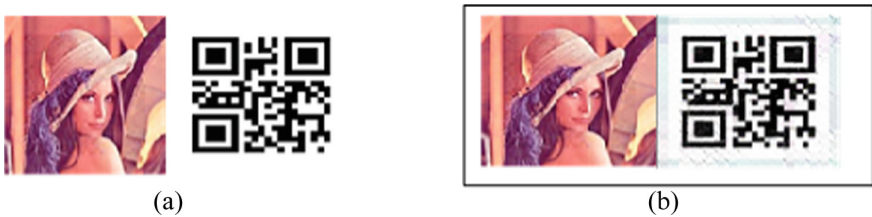


Fig. 7. (a) Blurring with radius 0.1 (b) Blurring with radius 1.0

Table 2 and Table 3 shows the experimental analysis of the impact of a malicious attack on the watermarked image. Table 2 depicts the impact of blurring, flipping, Gaussian noise, and Gaussian filter on the watermarked image. Results noted in Table 2 explain the blurring with 0.1. The recovered watermark (recover QR Code) is decodable but blurring with 1.0 recover watermark is not decodable. Flipping with horizontal vertical and total recover watermark is not decodable. For creating the QR Code and decoding QR Code, the authors used the Zxing QR Code generator and decoder software. Similarly, the impact of the Gaussian filter on the watermarked image recover watermark is not readable. But the impact of Gaussian noise (0.0, 0.001) is readable, and (0.1, 0.001) is not readable. Similarly, Table 3 shows the experimental analysis of the impact of the median filter, winner filter, rescale, rotation and Salt & Pepper noise. Filtering of the watermarked image using median and winner filter, scaling with scale 0.5 and 2 recover watermark is readable. But rotation with scale 0.1 and 0.5 is not readable.

Table 2. Impact of Blurring, Flipping, Gaussian Noise, and Gaussian Filter

	Blurring		Flipping			Gaussian Noise		Gaussian Filter	
	0.1	1.0	Horizontal	Total	Vertical	0.0, 0.001	0.1, 0.001	3×3	5×5
Watermark									
Recover Watermark									
Zxing decoder	YES	YES	NO	NO	NO	YES	NO	NO	NO

Table 3. Impact of median filter, winner filter, rescale, rotation, and Salt & Pepper noise.

	Median	Winner Filter	Rescale		Rotation		Salt & Pepper	
			0.5	2	0.1	0.5	0.002	0.008
Watermark								
Recover Watermark								
Zxing Decoder	YES	YES	YES	YES	NO	NO	YES	NO

5 Conclusion

Through this paper, the authors attempted to introduce a new color digital image watermarking scheme. The sole idea behind this is the appropriate inclusion of watermark using DWT and SVD on a color image, to achieve robust and imperceptible features. This new watermarking approach has a decent efficiency as it can take out the watermark from the watermarked image without any difficulties. Furthermore, experiments suggested that this method is robust and resilient against many conventional attacks such as rescaling, blurring, winner, and median filter, etc. Since the watermarking approach in this study is based on the orientation of the images, it is needed to bring back the position of the watermarked image into its original position so that the embedded watermark may be extracted efficiently. For such attacks, this technique yields a poorer result than [15]. Furthermore, the work is presently in progress to intricate a mechanism that meets security requirements.

References

1. Tao, H., Chongmin, L., Zain, J.M., Abdalla, A.N.: Robust image watermarking theories and techniques: a review. *J. Appl. Res. Technol.* **12**, 122–138 (2014)

2. Al-Asmari, A.K., Salama, A., Ilyyasu, A.M., Al-Qodah, M.A.: A DWT ordering scheme for hiding data in images using pixel value difference. In: Eighth International Conference on Computational Intelligence and Security, pp. 553–557 (2012)
3. Khan, A., Siddiqa, A., Munib, S., Malik, S.A.: A recent survey of reversible watermarking techniques. *Inf. Sci. (NY)* **279**, 251–272 (2014)
4. Liu, J.J., Jiang, H., Liang, G.Y.: A watermarking algorithm for digital image based on logistic and SVD. In: International Conference on Intelligent Computing and Integrated Systems, pp. 140–144. IEEE (2010)
5. Bhuyan, T., Srivastava, V.K., Thakkar, F.: Shuffled SVD based robust and secure digital image watermarking. In: International Conference on Electrical, Electronics, and Optimization Techniques, pp. 1229–1233. IEEE (2016)
6. Golshan, F., Mohammadi, K.: A hybrid intelligent SVD-based digital image watermarking. In: 21st International Conference on Systems Engineering, pp. 137–141. IEEE (2011)
7. Sun, R., Sun, H., Yao, T.: A SVD-and quantization based semi-fragile watermarking technique for image authentication. In: 6th International Conference on Signal Processing, pp. 1592–1595. IEEE (2002)
8. Liu, F., Liu, Y.: A watermarking algorithm for digital image based on DCT and SVD. In: Congress on Image and Signal Processing, pp. 380–383 (2008)
9. Rao, V.S., Shekhawat, R.S., Srivastava, V.: A DWT-DCT-SVD based digital image watermarking scheme using particle swarm optimization. In: IEEE Students' Conference on Electrical, Electronics and Computer Science, pp. 1–4. IEEE (2012)
10. Rong Wei, C., Liu, J.J., Liang, G.Y.: A DCT_SVD domain watermarking algorithm for digital image based on Moore-model cellular automata scrambling. In: International Conference on Intelligent Computing and Integrated Systems, pp. 104–108. IEEE (2010)
11. Lin, C.-C.: An improved SVD-based watermarking scheme using human visual characteristics. *Opt. Commun.* **284**, 938–944 (2011)
12. Chang, C.-C., Tsai, P., Lin, C.-C.: SVD-based digital image watermarking scheme. *Pattern Recogn. Lett.* **26**, 1577–1586 (2005)
13. Abdallah, E.E., Hamza, A.B., Bhattacharya, P.: Improved image watermarking scheme using fast Hadamard and discrete wavelet transforms. *J. Electron. Imaging* **16**, 033020 (2007)
14. Xing, Y., Tan, J.: A color watermarking scheme based on block-SVD and Arnold transformation. In: Second Workshop on Digital Media and its Application in Museum & Heritages, pp. 3–8. IEEE (2007)
15. Golea, N.E.-H., Seghir, R., Benzid, R.: A bind RGB color image watermarking based on singular value decomposition. In: ACS/IEEE International Conference on Computer Systems and Applications, pp. 1–5 (2010)
16. Mathew, K.: SVD based image watermarking scheme. *Int. J. Comput. Appl.* 21–24 (2010)
17. Kansal, M., Singh, G., Kranthi, B.: DWT, DCT and SVD based digital image watermarking. In: International Conference on Computing Sciences, pp. 77–81 (2012)
18. Xueyi, Y., Meng, D., Yunlu, W., Jing, Z.: A Robust DWT-SVD blind watermarking algorithm based on Zernike moments. In: International Conference on Communications Security (2014)
19. Singh, R.K., Shaw, D.K., Alam, M.J.: Experimental studies of LSB watermarking with different noise. *Proc. Comput. Sci.* **54**, 612–620 (2015)
20. Singh, R.K., Shaw, D.: Experimental analysis of impact of noise on various edge detection techniques. In: Proceedings of the World Congress on Engineering (2016)
21. Singh, R.K., Kumar, B., Shaw, D.K., Khan, D.A.: Level by level image compression-encryption algorithm based on quantum chaos map. *J. King Saud Univ. Inf. Sci.* (2018)
22. Singh, R.K., Shaw, D.K., Sahoo, J.: A secure and robust block based DWT-SVD image watermarking approach *J. Inf. Optim. Sci.* **38**, 911–925 (2017)



A Concept of E-Medical Record Storage and Sharing Based on Blockchain Technology

Adla Sanobar  and Shamama Anwar^(✉) 

Department of Computer Science and Engineering, Birla Institute of Technology, Mesra,
Ranchi, Jharkhand, India
shamama@bitmesra.ac.in

Abstract. Medical data contains sensitive information and can have big impact if stolen. The term medical identity theft is coined for such thefts. Medical records are collected from various sources like hospitals, diagnostic labs, physicians, pharmacy and health insurance companies and includes all details of patient including his demographic information, test reports like X – rays, CT scans, MRIs, etc. With the advent of digitization, these medical records are now stored in digital form to make access and sharing easier. However, storing and sharing these data electronically opens the threat of data theft and misuse. Health insurance companies often bear the brunt of fraudulent claims based on stolen medical data. So, the current need is to enable storing and sharing these data with security and make a prohibition on making copies of such data. Hence, considering the importance of healthcare data, the use of blockchain technology can be promising to maintain the security, privacy, immutability, authentication, and reliability of the data to the intended users. The aim of this paper is to investigate the current state of blockchain technology and its diverse applications. The paper also discusses the application of the blockchain technology in the domain of secure medical health record sharing.

Keywords: Blockchain · Electronic medical record · Security · Immutability and authentication

1 Introduction

The concept of blockchain usually gives a sense of a secure financial transaction system specifically of bitcoins. But blockchain is beyond bitcoin, it enables sharing records of a decentralized data storage across many computers without interruption of trusted third party where computers or nodes are divided into a distributed ledger. When records are stored in a blockchain, it cannot be modified. The reason for success of blockchain is due to its features which includes security, reliability, immutability, authentication, non-repudiation, data access, data sharing, decentralized storage, mobility and transparency [1]. Any entity which links to the blockchain is called a node. Some nodes are called full nodes if they are verified by the blockchain. These nodes assemble the transactions into blocks, and they are accountable for determining whether the transactions are valid

or not. So, blockchain is basically a group of interrelated nodes which provide specific features to the infrastructure. At the lowest level of this infrastructure, are the digitally verified transactions between different nodes. These transactions denote an agreement between at least two participants, which may involve the movement of physical or digital assets, task accomplishment etc. At least one participant signs this transaction, and it is disseminated to its neighbors in a distributed network [2, 3]. The various terminology used in context of blockchain are described in Fig. 1.

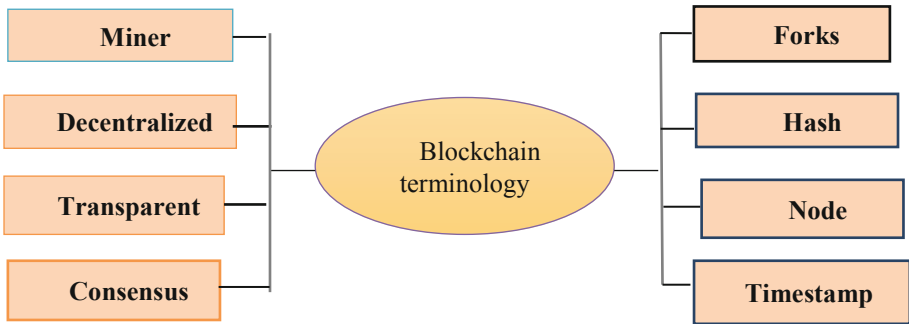


Fig. 1. Terminology of blockchain

As already mentioned, Node is an entity that connects to the blockchain system. A Miner is an entity that verifies a generated transaction and adds it to the blockchain. Fork refers to the problem that arises when the node is used for the different categories of blockchain. A Hash is a one-way function that checks the immutability (integrity) of a transaction or message and Timestamp is a date and time format in the computer system used as an electronic timestamp for the transaction. These components aid to make the blockchain system Distributed and Transparent. Consensus means that the nodes on the network agree on the same state of the blockchain and these algorithms allow the blockchain to be updated maintaining consistency. The two most basic consensus algorithms widely used for node validation of blockchain transactions are [4]:

- (a) **Proof-of-work:** It is the interrelated node for block validation in which each node must compete to add the new block in the blockchain by solving a computationally challenging mathematical puzzle. After solving the puzzle, the result is broadcasted to the network; and later the solution is verified by the remaining nodes on the network.
- (b) **Proof-of-stake:** It is partially similar to the above-mentioned method barring the fact that in this scheme only a randomly selected subset of nodes gets the chance to produce block in the blockchain. There is no competition as a block creator between nodes in the network. The selected node's dominancy on the network is acquired by choosing nodes which has the highest level of investment (monetary) in the system.

However, use of blockchain is not only limited to financial (cryptocurrency) applications [1] (e.g. Bitcoin, Ripple etc.). Recently researchers have explored broad prospects

for blockchain. The applications identified through the literature is summarized in Fig. 2. Further, there are three categories of blockchain in a network’s management environment based on permissions [2]:

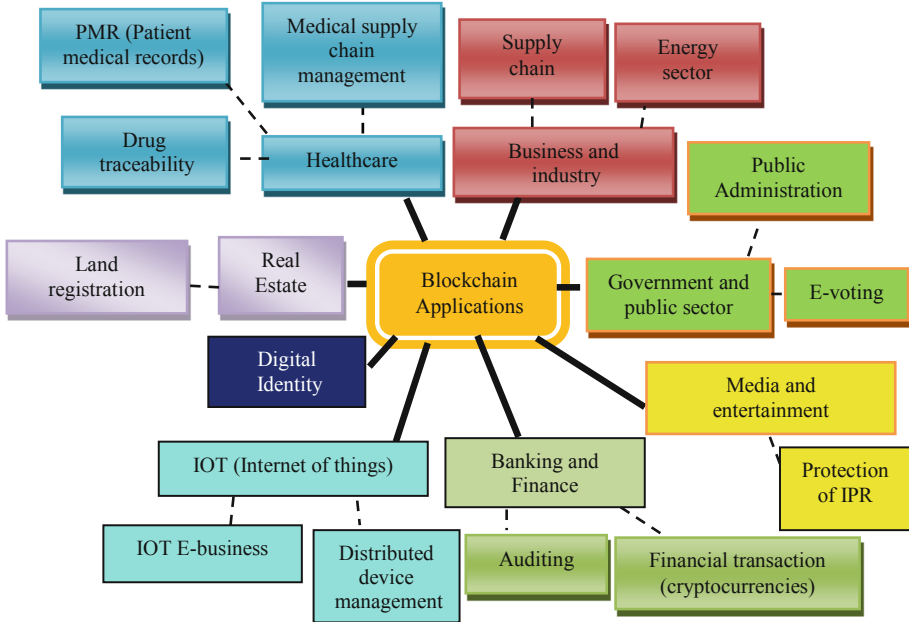


Fig. 2. Blockchain application

- (a) Public blockchains: It is a permission-less blockchain. Everyone in the network can participate for verification of the transaction processes. Example: Bitcoin, Ethereum, Litecoin etc.
- (b) Private blockchains: It is a permission-based blockchain, only limited participant or nodes can participate for verification of the transaction processes. Example: auditing (recording all log files), database management etc.
- (c) Federated blockchains: It is a composite of public and private blockchains. Leader nodes (set of nodes) are appointed to verify the transaction processes as a substitute of a single node. Example: banking and industry sectors. Nowadays, Ethereum has also delivered tools for building federated blockchains.

2 Related Work

According to Pinyapath et al. [5], the blockchain is a database repository that is noncentralized, authentic, and difficult to use for criminal deception purposes. Bitcoin, on the other hand, is a digital currency that uses the application of Blockchain technology to make transactions more secured across the distributed network environment. The basic

terminology of blockchain is very well explained in the paper and it also differentiated between the concept of Blockchain and Bitcoin. A further discussion on the applications of blockchain was included in [2]. The paper presented an organized literature review on blockchain-based applications across multiple areas of research. It also included a categorization of blockchain-based applications across multiple sectors such as supply chain, business and industry, healthcare, IoT E-business, government and public sector, database management etc. with their limitations and established key ideas, flow of research direction and emerging areas for research. The main features of the Blockchain technology was highlighted by Thomas McGhin et al. [1], which explained the five basic characteristics of Blockchain such as authentication, interoperability, immutability, security and privacy and sharing information. These features are the requirement of any healthcare related data sharing and management system. The paper also gives an overview of Blockchain platform for Electronic Clinical Trials and Precision Medicine based systems. Blockchain technology has its own specific susceptibility and issues that needs to be addressed, such as mining impulses, mining attacks, and key management which is also discussed in the paper. Hughes et al. [3], highlighted the potential of blockchain characteristics in business applications. They presented a bitcoin transaction process between the different entities and also described the limitation of blockchain technology such as lack of privacy, high cost, latency and governance.

The earlier onset of the blockchain technology regarding medical domain can be seen in [6]. This paper presented a review of blockchain applications in biomedical domain, the level of state of respective approaches, types of biomedical data examined, detailed discussion on blockchain features and functionalities and the application of blockchain technology frameworks. Further, Meng Shen et al. [7], discussed about retrieval of medical images which contain patient's sensitive and private information with privacy protection based on blockchain technology. They described the archetypal scenarios of medical image retrieval and summarized the corresponding requirements in the layered architecture of blockchain based system design and illustrated the detailed design in each layer. In order to accommodate large-size images with storage-constrained blocks, a set of a carefully picked out feature vector were drafted from each medical image. In the same context, electronic sharing and retrieval of medical imaging data that uses blockchain technology based on distributed storage of patient medical records are discussed in [4]. A framework was developed for image sharing under the medical domains that uses blockchain as a distributed data repository to establish a ledger (database) of radiological studies and permission based blockchain under the patient-defined access permissions. The blockchain framework has shown to eliminate trusted third-party access to protected health related information details which satisfy many criteria of an interoperable and reliable health system. Medical images are not stored on the blockchain. Only the list of the key owners of transaction records are grouped into blocks that are permitted to access each study. When a request is received by the creator (hospital), the creator verifies that the signature is correct, checks that the Unique id stored in the repository which it does by using its own public key. This method confirmed that the hashed UID of imaging data is published earlier for the patient. After verification is done, the block is added in the blockchain where the patient has granted the request access to these images for reviews of the medical history.

Licheng Wang et al. [8] further highlighted the essence of cryptographic primitives or algorithms in blockchain. The paper classified the cryptographic primitives into two classes: primary and optional. The primary class included one-way cryptographic hash function and standard digital signature schemes that are crucial for ensuring the blockchain as immutable and verifiable. While the optional class is mainly referred for enhancing the privacy and anonymity of blockchain based transactions such as ring signature, zero knowledge proof and so on. More insight into permissioned blockchain was given in [9]. The authors presented a blockchain technique that allows access to only authentic nodes/users, and hence, these nodes are called verified nodes. They described in detail about blockchain-based data sharing framework that adequately addresses the access control challenges linked with sensitive data stored in the cloud using tamper-proof and built-in autonomy properties of the blockchain. The system permitted users to request data from the shared pool after their identities and cryptographic keys are verified. The evidence from the system evaluation showed that their scheme is lightweight, expandable, and well organized.

Further insight into application of blockchain in electronic medical records were provided by [10]. The authors used Electronic medical records as a cumulation of personal sensitive data that had to be shared between nodes and developed an EMR (Electronic Medical Record) Knowledge sharing system. On a similar workflow, [11] discussed the problem faced by an Electronic Health Record (EHR) system regarding data security, integrity and database management. In this paper, the authors described how the blockchain technology can be used to transform the EHR systems and could be a solution of these issues. The paper presented a framework for the implementation of blockchain technology in the electronic healthcare records. Moreover, this framework also discussed the scalability problem faced by the blockchain technology in general via use of off-chain storage of the records. This framework provided the EHR system with the benefits of having a scalable, secure and reliable environment of blockchain-based solution.

Based on the insights gained by an extensive literature review, this paper proposes an architecture for securely accessing and sharing medical data between different peers.

3 Proposed Methodology

The current work proposes a blockchain based system to manage electronic medical records. The section on the architecture and workflow below describes the working of the system in details.

3.1 Architecture of Blockchain

Each block in a blockchain is categorized into two main components: a block header (primary component) and transactions (secondary component). Block header maintains the sequencing of blocks and immutability of the blockchain. These block headers include elements such as unique block identifier, hash of the previous and current block, timestamp, nonce and total size of the block. The secondary components of a block in the blockchain is the transaction elements. The transaction elements include types of data,

list of transaction, hashed UID of data and digital signature. The format of block is illustrated in Fig. 3. In short, the summarization of primary and secondary components of block format are: **Unique block identifier** is a unique block name or value of each block in the blockchain, **Previous block's hash** is the hash value of the previous block and chaining is achieved through this hash. **Current block's hash** is a current hash value of the block. **Timestamp** is a date and time format of each block in the transaction process of computer system. **Total block size** is a total size of each block and **Nonce** is a pseudo random number which is used only once in each block to maintain the difficulty level of the consensus algorithm. Secondary components are: **Types of data** which specifies the format of the data that needs to be shared. In this paper the types of data would include the patient details and diagnostic reports (which could be textual, numerical data or images). **Digital signature** is a scheme which depicts the verification of the information held in the block and is meant for authentication process. **Hashed UID** of data is a one-way hash value of data's unique identifier and **List of transaction** contains all the transaction details in list of each block. Each block has the same structure and have different UIDs and transaction lists. A block in this system stores the details of the patient and the transactions and reference to the diagnosis been made for the patient by different participating entities (physician, diagnostic labs, etc.).

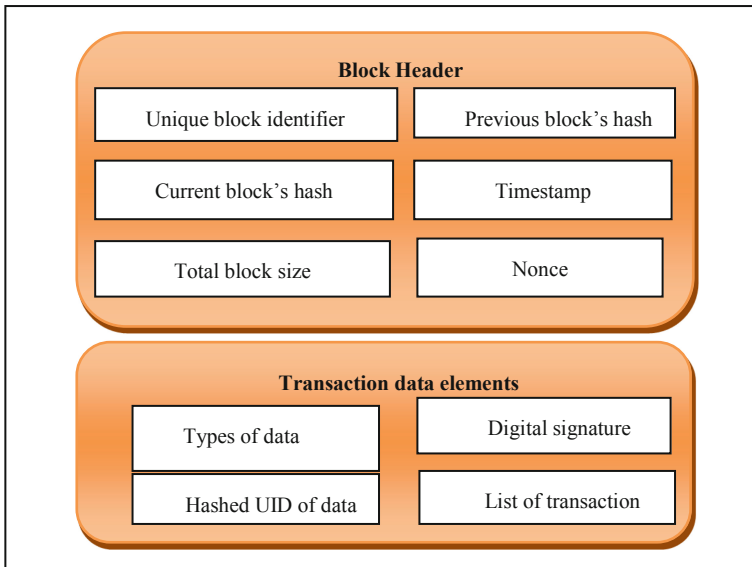


Fig. 3. Block format

3.2 Workflow of Blockchain Transaction

The entire process of the proposed technique is summarized in Fig. 4. The transaction process is based on conventional security protocol notation such as public key, M_e and

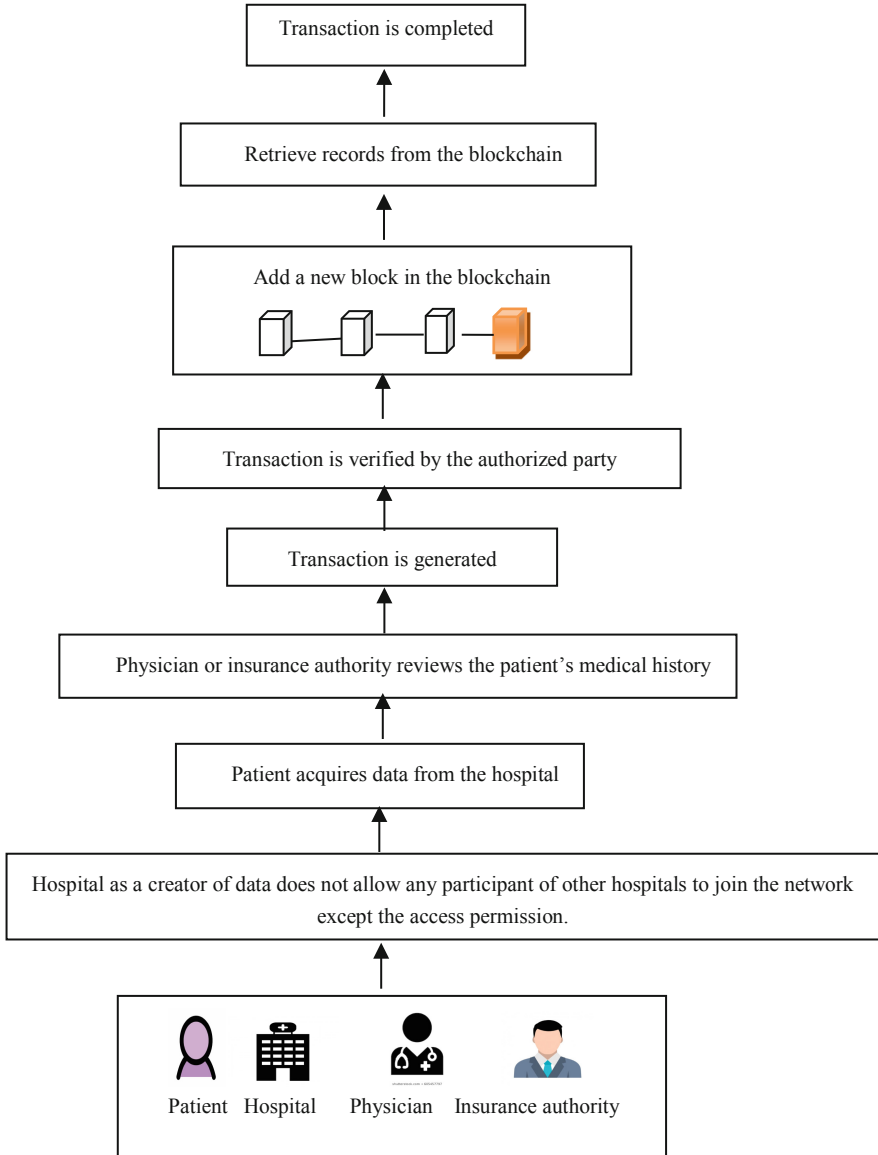


Fig. 4. Flowchart of blockchain transaction process

private key, N_e where e is the entity or participant. The proposed system is a permission-based system; hence all users are verified for authenticity before they become a part of the blockchain. Each entity participates in the transaction process. When a patient P registers at a hospital, a block with his encrypted records is created and added to the blockchain. All records pertaining to the patient is maintained at a repository. Only authorized entities participate in the permissioned blockchain. Hospital does not allow access permission to the external entities other than their own entities of the hospital. Each medical record (test reports, physician diagnostic reports) have a specific *UID* (unique identifier). This UID is converted into one-way hash value using hash algorithm [10] to maintain the integrity of data. This hashed UID is digitally signed by private key, N_e and is later verified by the public key pair, M_e . Authentication is done through digital signature. ECDSA and EdDSA are the two most popular digital signature schemes [8].

After a block is created for a patient it is added to the blockchain structure. Each participant such as patient, hospital, physician or insurance authority are participants in the permission-based blockchain system. For authentication, hospital and patient digitally sign the data (medical reports) using the hospital's and patient's private key, N_H and N_P respectively. This is then stored in a hash form and can later be verified through the hospital's and patient's public key, M_H and M_P respectively. A valid signature for authenticity must be done for each participant before they become the part of the permissioned blockchain. Now, all the transaction records are grouped into blocks by the hospital (owner).

Now, if a patient P requests record at hospital H and wants to share the records with a physician D , then the hospital becomes the creator of the data and the patient becomes the owner of the data. Patient grants access permission to the physician of that hospital where the records are stored. The signing and verifying process between them is also done by the patient's private and public keys, N_P and M_P respectively and physician's private and public keys, N_D and M_D respectively. Finally, physician retrieves the records from the blockchain, and can make diagnosis. Now here, diagnosis becomes a new transaction that requires to be added to the block. Hence, the transaction is again verified by the patient's and physician's private keys and are added to the block representing the patient's details. At this stage, the transaction is said to be complete. Figure 4 represents the flowchart of the workflow.

4 Conclusion

Gaining an insight into the current applications of blockchain technology, the paper aims to present an application for this technology in terms of maintaining and sharing electronic medical records. Due to the threat of medical fraud, this sharing must be secured. Thus, the paper presents an efficient outlook into generating records into blocks representing the patient. The patient's medical records are stored only in one hospital where patient is an owner of the data and hospital is a creator of the data. Both digitally verify the data being uploaded in the block hence making authentication and validity of the data easier to verify. Once a block is created, it can be easily shared by interested and identified entities. Hence, they can review the medical history for their intended use. Since, the technique discussed in this paper is a permissioned blockchain technique, no

unauthorized user has access to the data. This proposed idea is to avoid the tampering of data (integrity) and provides authenticity.

5 Future Work

The future work will focus on encryption technique which can be applied for enhancing patient's medical data security that will be stored in the block with the decentralized manner using blockchain technology across the different city's hospital. Medical data can be text or multimedia. Each hospital will make a block of each patient. Verification process is done through digital signature between hospital and patient. If same patient id is already existing in another hospital, then block will add with the previous block in the blockchain network. Only valid and verified entity in the transaction process could retrieve the data from the hospital database. This type of system would provide a breakthrough in record sharing scenario.

References

1. McGhin, T., Choo, K.-K.R., Liu, C.Z., He, D.: Blockchain in healthcare applications: research challenges and opportunities. *J. Netw. Comput. Appl.* **135**, 62–75 (2019)
2. Casino, F., Dasaklis, T.K., Patsakis, C.: A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telematics Inform.* **36**, 55–81 (2019)
3. Hughes, L., Dwivedi, Y.K., Misra, S.K., Rana, N.P., Raghavan, V., Akella, V.: Blockchain research, practice and policy: applications, benefits, limitations, emerging research themes and research agenda. *Int. J. Inf. Manag.* **49**, 114–129 (2019)
4. Patel, V.: A framework for secure and decentralized sharing of medical imaging data via blockchain consensus. *Health Inform. J.* **25**(4), 1398–1411 (2018)
5. Tasatanattakool, P., Techanpanupreeda, C.: Blockchain: challenges and applications. In: *International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand (2018)
6. Drosatos, G., Kaldoudi, E.: Blockchain applications in the biomedical domain: a scoping review. *Comput. Struct. Biotechnol. J.* **17**, 229–240 (2019)
7. Shen, M., Deng, Y., Zhu, L., Du, X., Guizani, N.: Privacy-preserving image retrieval for medical IoT systems: a blockchain-based approach. *IEEE Netw.* **33**(5), 27–33 (2019)
8. Wang, L., Shen, X., Li, J., Shao, J., Yang, Y.: Cryptographic primitives in blockchains. *J. Netw. Comput. Appl.* **127**, 43–58 (2019)
9. Xia, Q., Sifah, E.B., Smahi, A., Amofa, S., Zhang, X.: BBDS: blockchain-based data sharing for electronic medical records in cloud environments. *Information* **8**(2), 44 (2017)
10. Rao, B.N.K., Rao, B.B.K., Vellingiri, J.: Block chain based implementation of electronic medical health record. *Int. J. Innov. Technol. Exploring Eng.* **8**(8), 715–730 (2019)
11. Shahnaz, A., Qamar, U., Khalid, A.: Using blockchain for electronic health records. *IEEE Access* **7**, 147782–147795 (2019)



Comparative Analysis of 6T, 9T and 10T SRAM Cells Using 18 nm FinFet Technology

Aksh Rastogi^(✉) and V. K. Tomar

Department of Electronics and Communication Engineering, GLA University, Mathura,
Uttar Pradesh, India

{aksh.rastogi_ec17,vinay.tomar}@gla.ac.in

Abstract. Memory design is the most complex part of any electronic devices. In this paper, comparative analysis of static random access memory (SRAM) cells such as conventional 6T, 9T and 10T has been performed and are simulated on cadence virtuoso tool using 18 nm Fin-FET technology. Fin-FET technology is more preferable for low power devices. The SRAM characteristics likewise delay, power and stability are determined out with cadence virtuoso tool. The read and write access time of 10T SRAM cell has been found to be 54.62% and 73% better as compared to Conventional (Conv.) 6T SRAM cell. The read stability and write ability of 9T SRAM cell is 1.21 times and 3.15 times better as of Conv.6T and 10T SRAM cells respectively. The read and write powers in 10T SRAM cell have been observed to be 73% and 31% less as compared to Conv.6T SRAM cell at 0.6 V supply voltage.

Keywords: Delay · Power · RSNM · WSNM

1 Introduction

The growing demand of portable battery operated systems has made power efficient processor. A lot work has been carried out in CMOS technology for designing a robust SRAM cell because of its incredible requirement as a cache memory. It plays a very important role in the processor of portable devices [1]. As the technology node reduces, the leakage power in CMOS devices is more as compared to Fin-FET technology. The Fin-FET device has faster switching speed as of CMOS technology. A Fin-FET is a multi-gate device, a MOSFET built on a substrate where the gate is placed on three or four side of the channel or wrapped around the channel, forming a double gate structure [2]. The power consumption is the main concern in the modern portable devices. Supply voltage scaling is a popular method to minimize the power dissipation [3]. As the supply voltage reduces, the performance of the device is degraded. However, Fin-FET technology is low threshold voltage that works properly in lower supply voltage with higher leakage power dissipation [4]. In 6-T SRAM cell [5], four transistors formed a latch structure and two access transistors are connected between storage node and pre-charge circuit. It was reported that read static noise margin is improved by 4 times and write static noise margin enhancement by 8% as compare with the Conventional 6T at same supply

voltage. Many researchers [6–10] have been reported power efficient SRAM cells with improved performance as explored in subsequent paragraph. Mohammad Ansari et al. [8] proposed 7T SRAM cell with Fin-FET technology. In this topologies, activity factor reduces to 0.5 which minimize the dynamic power dissipation. Paridhi Athe et al. [9] reported that 6T SRAM cell has better delay as compare to 9T SRAM cell. On the other hand, 9T SRAM cell has higher read static noise margin (RSNM) as well write static noise margin (WSNM). Ramya P. et al. [10] reported that that 10T SRAM cell has less power dissipation compared to 6T and 9T SRAM cells.

This paper is organized in the following way. Section 2 presents the schematics of implemented topologies along with read and write operation. Section 3 presents the simulation and analysis of obtained results. Section 4.

2 SRAM Cells Topologies

2.1 6T SRAM CELL

The conventional 6T-cell schematic is shown in Fig. 1. It consists of two cross coupled inverters (PMOS pull-up transistors PM1 and PM2 and NMOS pull-down transistors NM1 and NM2) and two access transistors (NMOS pass-gate transistors NM3 and NM4) [4]. When WL is high, the access transistors are turned on and connect to the bit lines (BL and BLB).

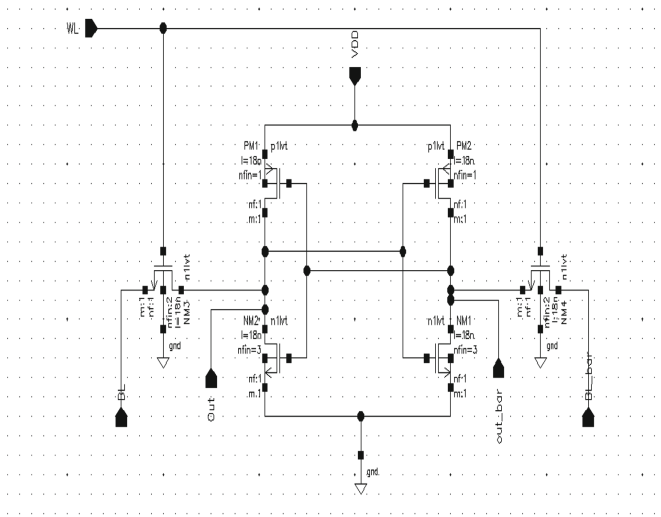


Fig. 1. Schematic of 6T SRAM Cell

During read operation bit-lines (BL and BLB) are set to pre-charge to VDD. WL is set to high to turn on the NM3 and NM4 transistors. During read ‘1’ the BLB gets discharged through NM1 and NM4 transistors and the BL is maintained to logic ‘1’. For a successful read operation, the cell beta ratio which is the ratio of the pull-down transistor

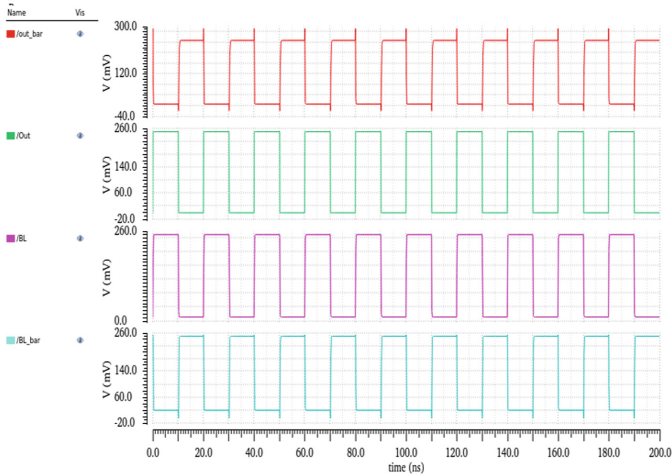


Fig. 2. Read operation of 6T SRAM cell

and the pass-gate transistor, should be greater than 1.6 [11]. During write operation, WL is set to high to turn on the NM3 and NM4 transistors. For write '1' operation, BL is set to high then BLB should be set to low. For write '0' operation BL is set to low and BLB should be set to high. For a successful write operation, the cell gamma ratio, which is the ratio of the pull-up transistor and pass-gate transistor, should be less than one [11].

The Static noise margin is used to measure the stability of SRAM cell. The RSNM (Read Static Noise Margin) is the maximum noise voltage that can be tolerated by the cell without changing the state. It is defined as the largest square box that can be fitted in the butterfly curve of SRAM cell [12]. The RSNM is obtained by plotting the VTC curves of the cross coupled inverters. For Simulation, it is required to add DC noise voltage source at the charging node. The VTC curves are taken out from the output of the inverters.

The WSNM (write static noise margin) is the maximum noise voltage present at bit lines [12]. WSNM is obtained by the both of the VTC curves (RVTC and WVTC). For WVTC, inverted input is applied at the bit lines and by sweeping one of them, the output

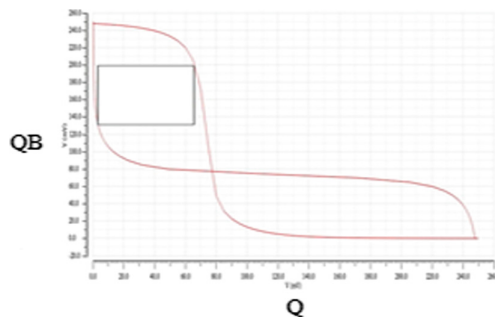


Fig. 3. RSNM of 6T SRAM cell

is taken from outputs of the cross coupled inverters. Figure 4 shows the WSNM of the 6T SRAM cell and it can be calculated using the square fitted in curve.

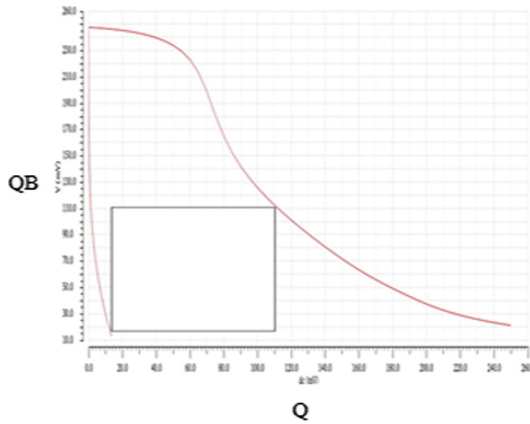


Fig. 4. WSNM of 6T SRAM cell

2.2 9T SRAM CELL

A 9T SRAM cell is demonstrated in Fig. 5. It is essentially a 6T SRAM cell with minimum sized devices. The data is stored in upper memory circuit. The lower sub-circuit of the cell composed of the bit line access transistors (NM5 and NM7) and the read access transistors (NM6) [4]. NM6 has a separate signal RD.

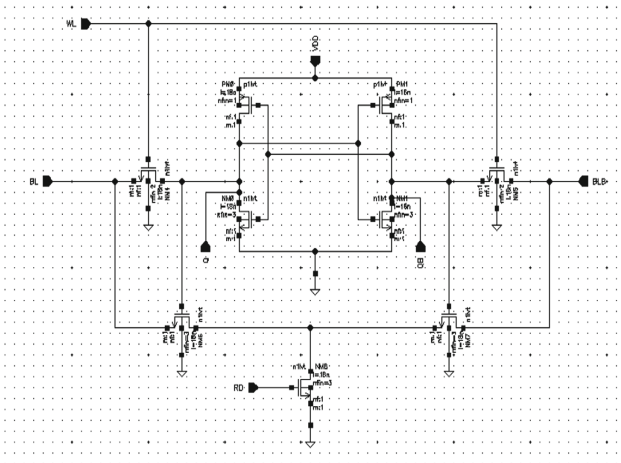


Fig. 5. Schematic of 9T SRAM Cell

During read operation, BL and BLB are connected to pre-charge and RD signal is fixed at 1 which turns on the NM7 transistor. The WL signal is set to low which turns on NM3 transistor and NM4 transistor becomes off [13]. For read operation, apply a low to high pulse at any storage node of the cell. If node Q holds 1 then BL will get discharged through NM5 and NM7 transistors and BLB hold its value. During write operation, WL is set to high which turns on NM3 and NM4 transistors. The RD signal is set to low which turn off NM6 transistor. To write logic '1', BL is set to high and BLB at compliment of BL. For writing logic '0', BL set to low and BLB at the compliment of BL [13]. The RSNM of 9 T SRAM cell is shown in Fig. 6. For RSNM RD is fixed to '1' and WL is kept at logic '0'. The voltage at the storage nodes is swept from 0 to VDD.

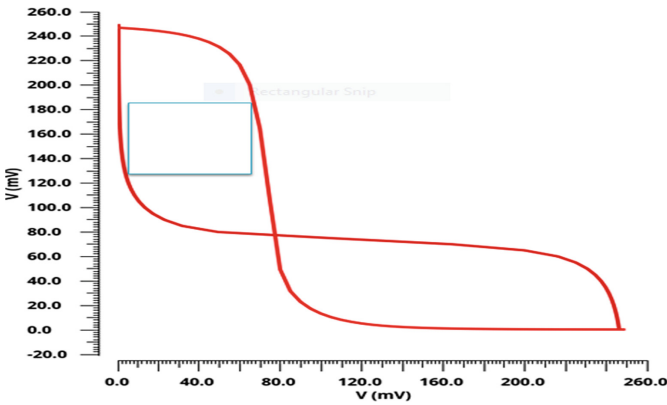


Fig. 6. RSNM of 9T SRAM cell

The WSNM of 9T SRAM cell is shown in Fig. 7. For WSNM, BL is set to logic '1' and its complement on BLB and sweep one of them. WL is fixed to logic '1' and RD kept at logic '0'.

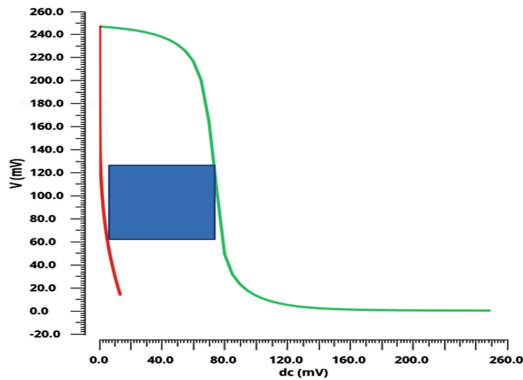


Fig. 7. WSNM of 9 T SRAM CELL

2.3 10T SRAM CELL

10T SRAM cell is demonstrated in Fig. 8. It has a single bit line which means it consumes less power compare to differential bit lines. PM1, PM2, NM1, NM2 make the cross coupled inverters to store the data. The access transistors are controlled by the RWL and WWL signal. NM3 and NM4 are controlled by S1 and S2 respectively. SL1 and SL2 are the sleep transistors which are ON only during read/write operation. When circuit is in standby mode these transistors remain off which decrease the leakage power in the cell [6].

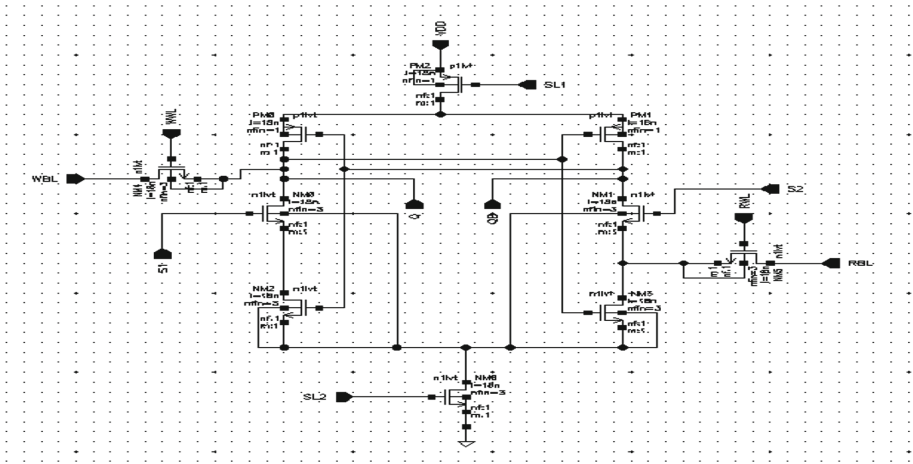


Fig. 8. Schematic of 10T SRAM Cell

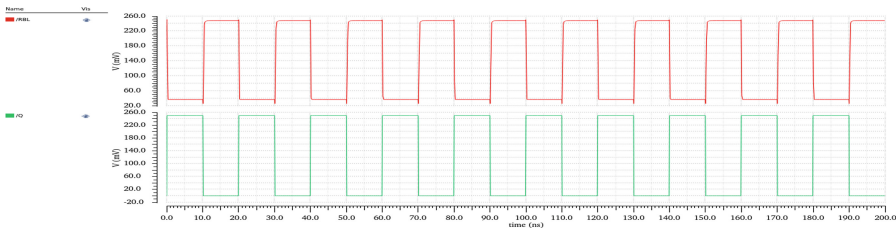


Fig. 9. Read operation of 10T SRAM cell

During the read operation WWL is fixed at logic ‘0’ and RWL is kept at logic ‘1’. SL1 is fixed to logic ‘0’ and SL2 is kept at logic ‘1’ to turn on PM3 and NM7 transistors. S1 and S2 are at logic ‘0’ which turns off NM3 and NM4 transistors. Apply a low to high pulse at the charging node (Q) and output is observed at RBL [6].

During write operation WWL is fixed at logic ‘1’ and RWL is kept at logic ‘0’. SL1 and SL2 are fixed at logic ‘0’ and logic ‘1’ respectively to turn on PM3 and NM7 transistors. Input pulse is provided at the WBL and S1. The inverted pulse is provided at S2. The output can be observed at the charging nodes (Q and QB) of the memory cell.

The noise margin for the read cycle of the 10T Cell is demonstrated in Fig. 10. For RSNM, the charging node is sweep from logic '0' to logic '1'. WWL and RWL are fixed at logic '0' and butterfly curve is observed at node Q and QB. Other signals are same as read operation. The write static noise margin during write cycle of 10T SRAM cell is demonstrated in Fig. 11. For WSNM, WBL is sweep from 0V to VDD. The WWL and RWL signals are kept at logic '1' and logic '0' respectively. SL and SLB are maintained at logic low and logic high respectively.

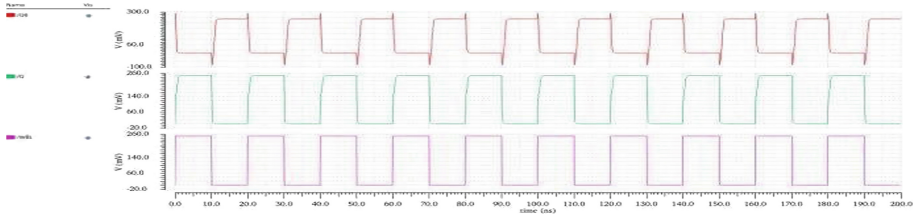


Fig. 10. Write operation of 10T SRAM cell

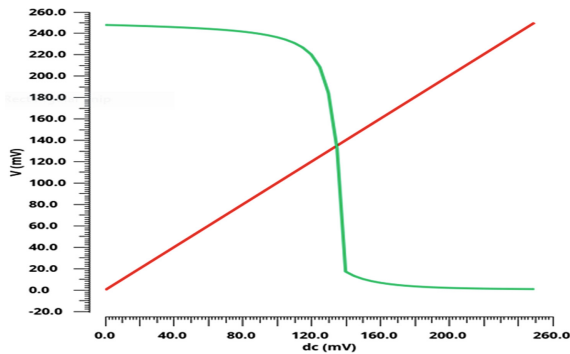


Fig. 11. WSNM of 10T SRAM cell

3 Results and Simulation

3.1 Read and Write Delay

The Read access time for considered SRAM cells topologies (6T, 9T and 10T) with variation in supply voltage is shown in Fig. 12.

It can be seen from the Fig. 12 that 10T SRAM cell has 54.62% improved delay as compared to conventional 6T SRAM cell at 600 mV supply voltage. Further, 9T SRAM cell has 24.38% better delay as compared to conv. 6T SRAM cell. The write delay is demonstrated in Fig. 13 and it is also calculated at various supply voltages. It has been observed from the Fig. 13 that 10T SRAM cell has 73.78% improved write access time as compared to conventional 6T SRAM cell. However, 6T SRAM cell shows an improved write access time as compared to 9T SRAM cell.

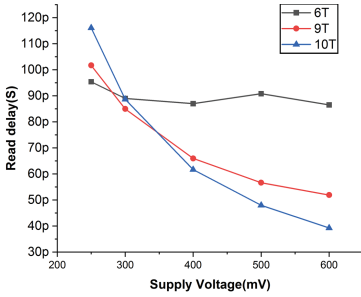


Fig. 12. Read delay vs supply voltage

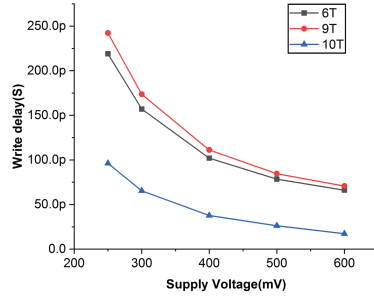


Fig. 13. Write delay vs supply voltage

3.2 Read and Write Power

The Read power consumption is shown in Fig. 14. It clearly shows that 10T SRAM cell has the lowest read power among 6T and 9T SRAM cells. 10T SRAM cell has 73% less power than 6T cell and almost 81% less power than 9T SRAM cell. The write power is shown in Fig. 15. It is evident that power calculated at different voltages. It can be seen from the graph 10T SRAM cell has lowest write power among all the SRAM cells. 10T SRAM cell has 31% less power than and almost 55% less power than 9T SRAM cell.

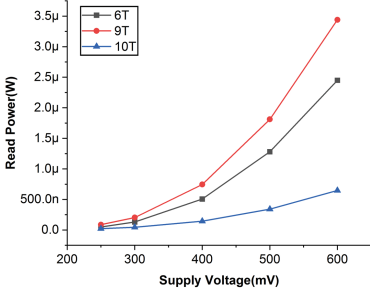


Fig. 14. Read power dissipation vs supply voltage

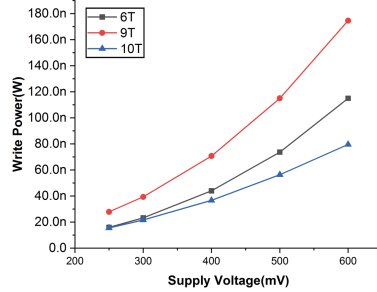


Fig. 15. Write power dissipation vs supply voltage

3.3 Read and Write SNM

The Read Static Noise Margin of all the different SRAM cells is shown in the Fig. 16. The RSNM should be high to stable the SRAM cell. It can be seen from the Fig. 16 that 9T SRAM cell has the highest RSNM i.e. 205 mv among other considered cells. The RSNM values of conv. 6T cell and 10T cell are 170 mv and 65 mv respectively. The WSNM of the different SRAM cell is shown in Fig. 17. It can be seen from the plot that 9T SRAM cell has the better WSNM than conventional 6T and the 10T SRAM cells. 9T SRAM cell has the 47% better WSNM than 6T and 10T SRAM cells.

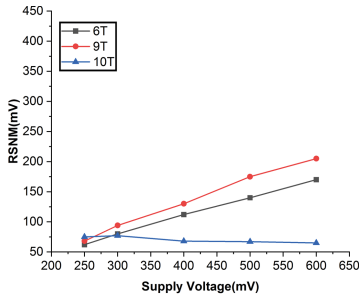


Fig. 16. Read static noise margin vs supply voltage

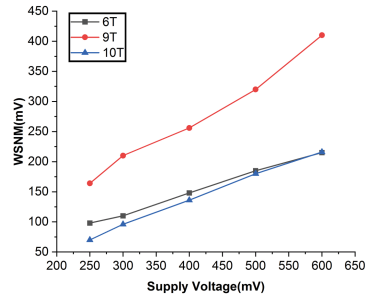


Fig. 17. Write static noise margin vs supply voltage

4 Conclusion

This paper, explore the implementation of various SRAM cells topologies on cadence virtuoso tool using 18 nm Fin-FET technology. It has been observed that 10T SRAM cell provides the improved delay and less power consumption in compare to conventional 6T and 9T SRAM cells. However, 9T SRAM cell provides the better read static noise margin and write static noise margin than conventional 6T SRAM cell. So for power constrained applications 10T SRAM cell can be utilized and for better stability 9T SRAM cell is the preferable choice.

References

1. Jawar, S., Mohanty, S.P., Pradhan, D.K.: Robust SRAM Designs and Analysis. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-1-4614-0818-5>
2. Fossum, J.G., Trivedi, V.P.: Fundamentals of Ultra-Thin-Body MOSFETs and FinFETs-Cambridge University Press (2013)
3. Kumar, H.K., Tomar, V.K.: Stability analysis of Subthreshold 6T SRAM cell at 45 nm for IoT application. *Int. J. Recent Technol. Eng. (IJRTE)* **8**(2) (2019). ISSN 2277–3878
4. Premalatha, C., Sarika, K., Mahesh Kannan, P.: A comparative analysis of 6T, 7T, 8T and 9T SRAM Cells in 90 nm Technology. In: *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (2015)
5. Anand, N., Sinha, A., Roy, C., Islam, A.: Design of a stable read-decoupled 6T sram cell at 16-nm technology node. In: *2015 IEEE International Conference on Computational Intelligence & Communication Technology (CICIT)*, pp. 524–528. IEEE (2015)
6. Banga, H., Agarwal, D.: Single bit-line 10T SRAM cell for low power and high SNM. In: *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, Bhopal, pp. 433–438 (2017)
7. Tomar, V.K., Sachdeva, A.: Implementation and analysis of power reduction techniques in charge transfer sense amplifier for sub 90 nm SRAM. In: *8th International Conference on Computing, Communication and Networking Technologies ICCCNT-2017*, held on 3rd -5th July- 2017, IEEE-40222, IIT Delhi, New Delhi (2017)
8. Ansari, M., Afzali-Kusha, H., Ebrahimi, B., Navabi, Z., Afzali-Kusha, A., Pedram, M.: A near-threshold 7T sram cell with high write and read margins and low write time for sub-20 nm finfet technologies. *Integration VLSI J.* **50**, 91–106 (2015)

9. Athe, P., Dasgupta, S.: A comparative study of 6T, 8T and 9T Decanano SRAM cell. In: 2009 IEEE Symposium on Industrial Electronics and Applications (2009)
10. Ramya, P., Shankavi, K., Shilpa, U., Sushmitha, N.M.: Design and comparison of various low power n-T SRAM cells SSRG. *Int. J. Electron. Commun. Eng.* **4**(5), 20–25 (2017)
11. Joshi, S., Hadia, S.: Design and analysis for low power CMOS SRAM cell in 90 nm technology using cadence tool. *Int. J. Adv. Res. Comput. Commun. Eng.* **2**(4), 1814–1817 (2013)
12. Sachdeva, A., Tomar, V.K.: Design of a stable low power 11-t static random access memory cell. *J. Circ. Syst. Comput.* **29**(13) (2020)
13. Liu, Z., Kursun, V.: Characterization of a novel nine-transistor SRAM cell. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.* **16**(4), 488–492 (2008)



Hybrid Cloud: The Next Generation of EAI

Manish Kumar Dash¹, Supriya Devidutta²(✉), Bhabendu Kumar Mohanta³,
and Debasish Jena³

¹ Indian Institute of Technology, Guwahati, Guwahati, India
manish.dash@iitg.ac.in

² International Institute of Information Technology, Bangalore, Bengaluru, India
supriyadevidutta.ml8@iiitb.net

³ International Institute of Information Technology, Bhubaneswar, Bhubaneswar, India
{C116004,debasish}@iiit-bh.ac.in

Abstract. Today's large enterprises run on a wide variety of independent applications, data sources, and infrastructure which is either disconnected or tied together using point-to-point integration or integration using middleware technologies. With the focus on **Move to Cloud** many new applications are hosted on Public cloud while many legacy applications, for a variety of compliance and security reasons, are still hosted on On-premises data centers or in private cloud across different networks/firewalls. In recent years, **Hybrid cloud** is a promising paradigm in cloud computing space. It provides seamless integration between on-prem, private and public cloud services and applications. There are many advantages to move to a hybrid cloud as compared to the existing middleware network. Migration to hybrid is as simple as 'a lift and shift', provided, the applications are containerized and cloud-ready. The native non-cloud applications can still use the benefit of hybrid cloud by replacing legacy middleware tools and achieving a point-to-point integration with hybrid cloud clusters. In this article, we explore the current middleware landscape with its associated pitfalls and compare it with the Hybrid Cloud approach. Additionally, this article will cover a brief introduction about Redhat OpenShift, a hybrid cloud product from IBM.

Keywords: Hybrid cloud · Enterprise application integration · EAI · RedHat · OpenShift · Service oriented architecture (SOA)

1 Enterprise Application Integration (EAI)

Enterprise Application Integration (EAI) is the integration of systems and applications across an enterprise, which is achieved by an underlying middleware network between applications. Once the middleware network is set up, data and business processes can seamlessly flow between these applications. Most of the large enterprises use multiple network zones and create layered firewalls to protect the network against threats, frauds and exposures. Many industry compliances like PCI (payment and credit card), PI-SPI (Personal information and Sensitive personal information), SOX (Sarbanes-Oxley Act) need to be adhered while building these middleware networks and hosting applications. Thus, the resulting environment is complex where applications are integrated across multiple zones, firewalls and proxies.

1.1 Evolution of the EAI Landscape

Middleware architectures have evolved over the years with the evolution of newer architecture, design principles and technologies. Point-to-point integration, Enterprise Service Bus (ESB) and Service-oriented architecture (SOA) are all part of these evolutions. The middleware networks usually follow a Hub and spoke approach over point-to-point approach.

In the Hub and spoke design, Enterprises use standard middleware products and carry out integrations, messaging, routing and profiling. Some of the well-known product vendors like Tibco, Webmethod, Sterling commerce and Axway provide a wide range of application servers to set up the middleware network. While setting up the network, a common practice is to establish a service bus/hub and integrate each of the application with the hub using **adaptors**. The standard middleware products provide a wide range of adaptors to help establishing a two-way communication between applications and message hub. Some of the common adapters are MQ, HTTP, File system, API, Java adapter, SOA, SAP and Oracle adapters. EAI products are built to integrate applications without worrying about underline technologies, operating systems, and environments (Fig. 1).

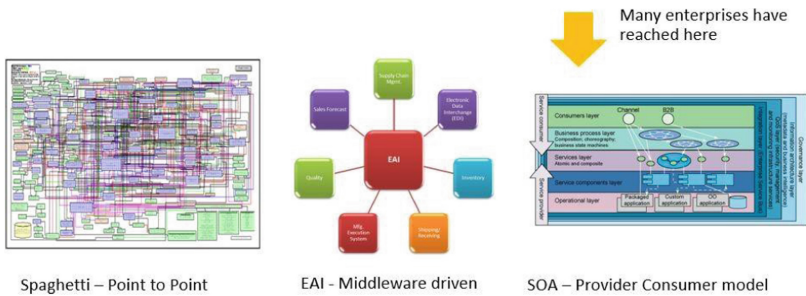


Fig. 1. Evolution of EAI architectures [1]

1.2 Current Challenges

In 2003, a report came out that 70% of all EAI projects fail [2]. Most of these failures are not due to the software or technical difficulties, but due to management issues. Integration Consortium European Chairman, Steve Craggs, has outlined some of the reasons of these pitfalls [3].

Key Reasons

- Constant change: Difficult to manage constant change in the application messaging layout, and customisation to existing implementations.
- Shortage of EAI expertise- EAI architects need to understand multiple environments, tools, technologies to implement the integration networks, which is challenging as technology continues to rapidly evolve.

- **Competing messaging standards:** There is no single universal messaging and networking standards and new standards keep evolving with modernisation of technology space, making it difficult to adopt standard and stable solutions.
- **Difficulty in Building interfaces:** Solutions need to be negotiated with multiple applications and business team to get a common buy-in, which is not easy.
- **Loss of details - Data** which seemed irrelevant a few years back may become a need today, necessitating ever-changing data retention policies.
- **Accountability:** Since so many departments have many conflicting requirements, there should be clear accountability for the end-to-end solution.
- **Lack of centralized coordination of work:** There is a lot of interdependencies across multiple application teams and implementation take a lot of time.
- **Lack of modularity for incremental additions and transformations:** EAIs, once setup, are standard systems that are not flexible to modularise and extensible for future changes.
- **Protectionism:** The applications whose data is being integrated often belong to a different department, country laws and cultural restrictions.

1.3 Future Perspective

With these pitfalls in managing a distributed EAI network, the demand for Cloud adoption has increased. Transformation of this integration landscape is a stepping stone towards enterprise's cloud journey. Traditional enterprise integration approaches throw a patchwork of solutions at this problem and such a setup means one-off tools and technologies accumulated into the technology cluster which is impossible to manage and difficult to scale.

Enterprises are looking into newer architectural patterns for connecting and integrating existing application with cloud-technologies. Modern architectures such as API/Micro-services and event driven architectures are the ideal choices. As enterprises continue to look for better solutions, Hybrid Cloud is an emerging paradigm which promises to alleviate many of the challenges discussed above.

2 Hybrid Cloud

As enterprises continue to move their workloads to the cloud, existing EAI approaches are not enough to keep up with the dynamic nature of cloud. This is evident by a shift in thinking in the EAI landscape. Instead of focusing on standard and stable EAI systems, modern enterprises want to be agile and flexible, with the ability to integrate anything.

Moving to cloud brings a plethora of benefits to enterprises. The cloud-native applications benefit from the loosely coupled dependency on the underlying infrastructure. The ability to dynamically scale based on the workload, the flexibility to evolve individual applications without disturbing the end-to-end solution and the freedom offered by the technology-agnostic nature of API/microservices enable enterprises to deliver modern products that are robust, resource-efficient and easily manageable. Hence, many enterprises today are trying to move majority of their workloads to the cloud.

But converting to an all-in-cloud integrated system is not always the best solution for all enterprises. There may be some legacy solutions that do not have the ability to migrate to cloud. Nor can the enterprises continue with out-of-date legacy systems. This dilemma gave rise to the idea to have a hybrid approach of integrating existing legacy solutions with modern cloud-native products.

Hybrid cloud integration provides a best-of-both-worlds approach, where on-premises legacy systems can integrate and share data with cloud solutions, enabling enterprises to drive business forward and migrate to Cloud while benefitting from the established and reliable legacy solutions. Hybrid cloud abstracts the underlying infrastructure from application development. This enables application teams to have a set of standard EAI policies to integrate data and processes at a low-level where the core policies for data security, privacy and regulatory compliance are embedded, while having complete freedom and flexibility in terms of application design.

The benefits of Hybrid Cloud are getting increasingly acknowledged by industry leaders across domains. “We know that hybrid and multi-cloud approaches represent the future for many of our customers,” says Kip Compton, SVP, Cisco [4]. As per a report by Gartner, 65 percent of large organizations will have realized a hybrid integration platform (HIP) by 2022 to drive their digital transformation [5].

2.1 Related Work

The idea for Hybrid Cloud arose for the need for better EAI solutions and as such there are been prior research done in the field. In [6] the authors propose the concept of dedicated Collaboration points (CPs) to act as points of connection between the platforms to form the hybrid cloud. In [7], the authors propose a Hybrid Cloud manager layer that enforces access and security policies for data to flow between On- Premise and Cloud platforms.

Hybrid Cloud has since permeated into other cloud-related domains. IoT is a natural extension as the IoT devices can be treated as low-bandwidth on-premise systems while the central application is sitting on the public cloud [8, 9]. Media-streaming platforms is another domain where Hybrid cloud can help to integrate the backend media data stores with the client-facing streaming interfaces [10, 11]. Resource-planning in the Hybrid Cloud is a difficult challenge as there needs to be a perfect balance between performing operations on Cloud and delegating tasks to the On-premises back-end [12, 13, 14].

3 Designing a Hybrid Cloud Network

Hybrid cloud is a combination of IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) working together to ensure complete integration between different types of platforms and systems. On-premises systems will seamlessly integrate with cloud based applications enabling businesses to connect their existing custom and packaged solutions and create a single platform for new services.

There are multiple ways to design a Hybrid cloud solution. In our design considerations, we focused on IBM-Redhat’s Hybrid cloud offering called OpenShift. There are many alternate Hybrid cloud products offered by all most all major cloud services providers like Google, Amazon, Microsoft, VMware etc.

3.1 IBM Redhat OpenShift

OpenShift is an IBM-RedHat solution for managing Hybrid cloud applications. OpenShift operates and orchestrates applications deployed on a set of nodes (computers) that may span multiple platforms, availability zones and network firewalls. In case of Hybrid cloud, some of these nodes exist on a public cloud platform, others may exist on a private cloud platform or on-premises datacenter, interconnected at a network-level.

Once the hybrid cloud network is set up, a OpenShift cluster can be deployed on top of this Hybrid Cloud with the flexibility to have its nodes sitting on any of the platforms or network zones. OpenShift defines a robust and secure framework for developing highly available cloud-native applications that can be deployed on the Hybrid Cloud.

OpenShift comes with 3 different types of nodes:

- Master nodes- that control the entire cluster.
- Infrastructure nodes - that help the master nodes monitor and manage the cluster.
- Application nodes- where the actual applications of the cluster will be sitting on

(Fig. 2).

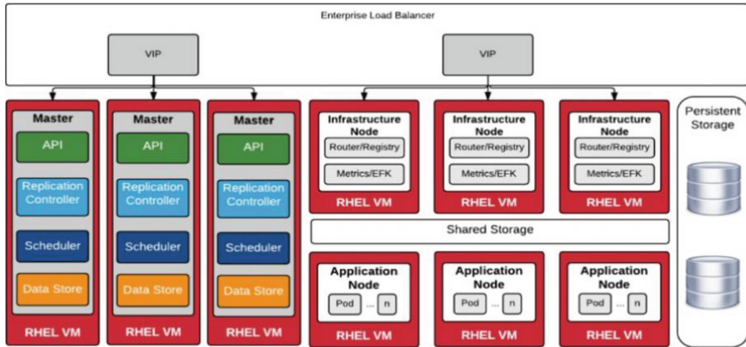


Fig. 2. A typical Openshift cluster setup

Openshift applications follow the standard approach of developing cloud-native applications based on containerisation. Whether an enterprise is developing a monolithic application or a microservices-based solution, it has to be first containerised using Docker before deploying to Openshift. Once the development team has finished building the container(s), OpenShift provide deployment templates for different types of workloads that allow the developers to deploy their Docker containers using minimal configuration. Using these templates OpenShift then deploys the applications as a service onto the cluster, where they enjoy all the benefits of a Hybrid Cloud environment such as auto-scaling, monitoring, load-balancing, auto-recovery and many more (Fig. 3).

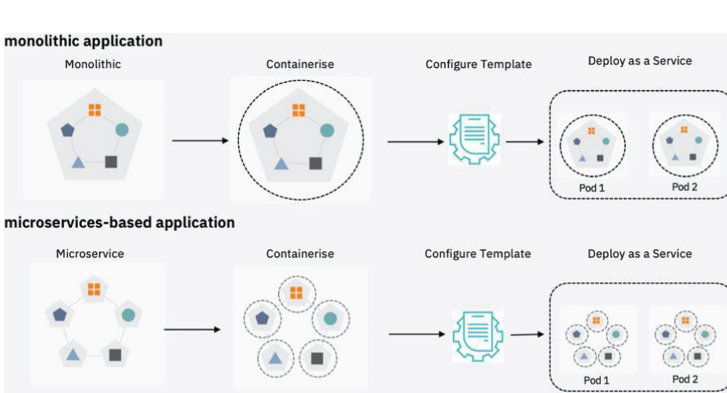


Fig. 3. Developing a cloud-native application for Hybrid Cloud

3.2 Experimental Setup

To compare between the standard EAI techniques and the Hybrid Cloud approach, we simulated both environments on a set of workstations with the same configuration: Intel® Core i7-4765T 2.0 GHz, 16 GB DDR3 RAM, 500 GB 5400 RPM HDD.

- **EAI solution:** We simulated a simple web application that has 3 different components: the database, the backend logic and the front-facing UI. These components were deployed in isolated VMs in separate workstations to simulate isolated platforms. To mimic the EAI middleware connections, we setup Kafka-based messaging queues running on separate VMs running on each workstation, configure in producer-consumer mode. This setup simulates the point-to-point integration approach of EAI.
- **Openshift cluster:** We setup the cluster on multiple CentOS 7.5 VMs running on different workstations to simulate multiple nodes. We further labelled each workstation as a different platform to simulate a Hybrid cloud system. These labelled workstations were inter-connected at a network level but with different firewall rules to simulate the varying compliance and security policies of an enterprise. After the test Hybrid cloud was set up, we used Ansible playbooks to install a Openshift cluster based on the community version of OKD 3.11 [15].

Each node and application can have multiple **labels** associated with it, that may be used to identify the type of functionality it provides or enforce specific policies. As an admin, we were able to define deployment, data retention and data sharing policies based on these labels, encapsulating the core policies for data security, privacy and regulatory compliance of an enterprise. This served as a demonstration of enabling standard EAI policies that are application and technology agnostic. OpenShift takes control of the continuous deployment and hosts the application across clusters, zones and firewalls based on the rule define by the policies. This completely abstract the developer from underlying complexity of networks and integrations, and adopts a “build once and run anywhere” approach.

For legacy On-prem applications, which cannot move to a cloud native container, Hybrid Cloud environment allows to have an application node of the cluster sitting in

the same datacentre as the legacy application. This allows cloud applications to access the non-cloud applications without relying on redundant middleware or re-routing.

4 Key Benefit Over Traditional EAI

Hybrid cloud integration is the next stage in the EAI landscape evolution. Applications will be designed focusing on business value, the underlying infrastructure will encapsulate the technologies, security and complexity and application-to-application communication would become a seamless API integration of micro services. Hybrid network would replace the traditional EAI, SOA technologies and legacy design practices. By alleviating the need to design customized integration solutions for the different systems and relying on the standard Hybrid Cloud integration, the focus can be shifted to factors that drive business value and growth: development agility, IT security, compliance, data security and access controls.

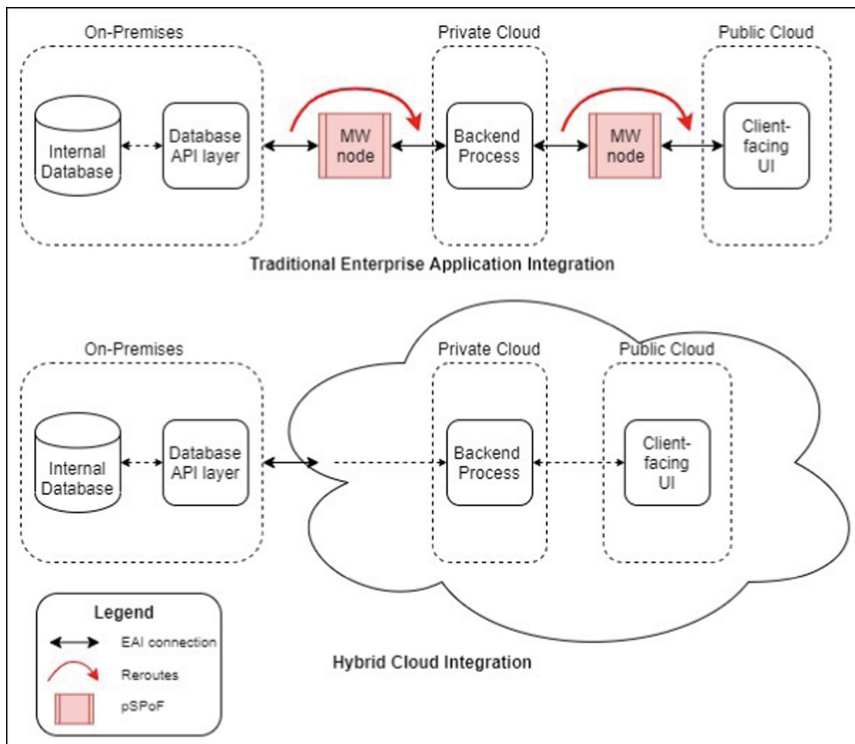


Fig. 4. Traditional EAI approach vs a Hybrid Cloud integration

Table 1. Comparison in performance of Traditional EAI vs Hybrid Cloud integration

Metrics	No. of transactions/sec	Point-to-point EAI	Hybrid cloud
Reroutes (n)	1	4	5
	100	255	170
	10,000	27121	17892
Latencies (ms)	1	29.75	17.65
	10	3025.59	1892.57
	10,000	31757.54	20,205.81

1. Enterprise can become truly cloud ready with a mix of cloud-native, hybrid cloud and pure legacy on-prem application stack. This ensures the enterprises to avoid resource wastage by continuing utilisation of legacy solutions while leveraging the agility and flexibility of cloud to achieve a reduced time to market [16].
2. By removing redundant middleware nodes and directly consuming data by the Hybrid cloud applications, solutions can have minimal reroutes between zones, platforms and systems, enabling lower latencies and more responsiveness. It is evident from Fig. 4 and the results from Table 1, where an example of a typical application transaction is shown.
3. Each middleware node in traditional EAI approaches is a potential Single Point of Failure (pSPoF). By reducing the number of possible pSPoFs, production failures due to integration, networking and messaging errors can be reduced.
4. In traditional EAI approaches, each component of the complete product is managed in different platforms, usually by different teams. This creates an additional complexity in terms of accountability, agreeing to development and messaging standards, and make SRE (Service Reliability Engineering) activities more difficult. By giving a single abstract platform to deploy the entire end-to-end solution with complete transparency will enable development teams to collaborate and SRE teams to have better tracing of the solution.
5. The abstraction provided by Hybrid Cloud allows the developers to focus on the application design and build-test-deploy code from a single interface without worrying about the underlying EAI challenges.
6. Conversely, Hybrid Cloud architects can define EAI solutions and policies for integrating with different platforms. These designs can be standardised as they are a low-level network connectivity, and more complex policies for security and compliance can be abstracted to the application layer.
7. As development teams adopt the cloud and related opensource solutions over legacy tools, enterprises will be able to keep up with the evolving technology, build modern, flexible and scalable applications, and avoid vendor lock-ins.

5 Conclusion

Cloud computing is no longer a new technology. IT teams have been procuring IaaS, PaaS and SaaS offering from all major cloud service provider but the inherent challenge

of migrating some of the enterprise applications running on legacy technology or vendor managed products, with flat license costs have held back many CIOs and CTOs from taking up the massive investment and risk of migrating to cloud. These legacy technologies have been serving businesses for decades. Legacy applications are more focused on business processes within the enterprises while Cloud applications are more focused towards B2C (User and consumer focused). With the help of Hybrid cloud, it is possible to integrate these two separate classes of applications and get the benefit of a single interface to manage end-to-end solution. Hybrid cloud integration is going to be the next big game changer in the enterprise application Integration domain.

References:

1. Joseph, J.: How to modernize the enterprise's integration landscape in the hybrid cloud era. <https://www.ibm.com/blogs/cloud-computing/2019/08/20/modernize-enterprise-integration-landscape-hybrid-cloud/>
2. Trotta, G.: Dancing Around EAI 'Bear Traps'. http://www.ebizq.net/topics/int_sbp/features/3463.html.
3. Enterprise application integration. https://en.wikipedia.org/wiki/Enterprise_application_integration.
4. Hölzle, U., Manor, E.: Introducing Anthos: an entirely new platform for managing applications in today's multi-cloud world. <https://cloud.google.com/blog/topics/hybrid-cloud/new-platform-for-managing-applications-in-todays-multi-cloud-world>
5. van der Meulen, R.: Use a Hybrid Integration Approach to Empower Digital Transformation. <https://www.gartner.com/smarterwithgartner/use-a-hybrid-integration-approach-to-empower-digital-transformation/>
6. Li, Q., Wang, Z., Li, W., Li, J., Wang, C., Du, R.: Applications integration in a hybrid cloud computing environment: Modelling and platform. *Enterprise Inf. Syst.* **7**(3), 237–271 (2013)
7. Breiter, G., Naik, V.K.: A framework for controlling and managing hybrid cloud service integration. In: *IEEE International Conference on Cloud Engineering* (2013)
8. Celesti, A., et al.: An approach for the secure management of hybrid cloud-edge environments. *Future Gen. Comput. Syst.* **90**, 1–19 (2019)
9. Zhang, Y., Zhou, J., Sun, L., Mao, J., Sun, J.: A novel firefly algorithm for scheduling bag-of-tasks applications under budget constraints on hybrid clouds. *IEEE Access* **7**, 151888–151901 (2019)
10. Liu, Y., et al.: Efficient multi-resource scheduling algorithm for hybrid cloud-based large-scale media streaming. *Comput. Electr. Eng.* **75**, 123–134 (2019)
11. Lu, P., Sun, Q., Wu, K., Zhu, Z.: Distributed online hybrid cloud management for profit-driven multimedia cloud computing. *IEEE Trans. Multimedia* **17**(8), 1297–1308 (2015)
12. He, H., et al.: A lightweight secure conjunctive keyword search scheme in hybrid cloud. *Future Gen. Comput. Syst.* **93**, 727–736 (2019)
13. Cao, Y., et al.: Online cost-rejection rate scheduling for resource requests in hybrid clouds. *Parallel Comput.* **81**, 85–103 (2019)
14. Towards operational cost minimization in hybrid clouds for dynamic resource provisioning with delay-aware optimization. *IEEE Trans. Serv. Comput.* **8**(3), 398–409
15. OKD 3.11 Documentation. <https://docs.okd.io/3.11/welcome/index.html>
16. Illsley, R.: Delivering Business Value through Transitioning from Managing VMs to Orchestrating Containers, 02 January 2019. <https://www.ibm.com/downloads/cas/JXY5L6DR>



Comparative Analysis of High Speed, Low Power FIR Multirate Polyphase Filter

Rajendra Rewatkar¹(✉) and S. L. Badjate²

¹ Datta Meghe Institute of Engineering, Technology and Research,
Sawangi (Meghe), Wardha, India

² S.B. Jain Institute of Engineering and Technology, Nagpur, India

Abstract. Authors have proposed the optimization of Multirate Finite Impulse Response (FIR) and Polyphase filters with the Multiple Constant Multiplication (MCM) shifting-adding concept. Proposed logics have reduced Multiplier, adders, and latches in the design. Due to which, it reduces area, power dissipation, and the circuit complexity in the system to a great extent. The primary objective is an optimization using Multirate DSP rather than targeting a single rate Digital Signal Processing (DSP). The proposed system approach can implement for the improvement in the important Parameters area, Power Dissipation, Speed and complexity. An approach provides the design and testing of the FIR filter, Multirate Polyphase filter using multiple constant multiplication and digit-serial architecture. It is an effective way to reduce the adders and subtractors in the Multirate finite impulse response filter. Investigational results are shown the effectiveness of the projected technique and the importance of different architectures. The simulation of essential parameters of the design is analyzed by using the software Active HDL, Synopsis 45 NM, and Xilinx. The Result is verified on the FPGA platform.

Keywords: Register transfer logic · Finite impulse response · Multiple constant multiplication · Field programmable gate array

1 Introduction

Multirate is a discrete-time system with an unequal sampling rate in various parts of the system. It is an efficient method and relevant to many problems. The sampling rate of a sampled signal may change due to many reasons. Multirate filters are interfaced with uninterrupted and sampled signals in a cost reduction part and upgrading of signal quality. A lot of the research efforts in the region of digital electronics have bound for raising the speed of the digital system. In recent times, the constraint of portability and upgrading in battery performance proves that power consumption is the major parameter. The important significant parameters to estimate the eminence of a circuit are area and power dissipation while challenging high-speed. Several proficient algorithms and

architectures made for simplicity. The complexity of DSP systems dominates due to the bit-parallel MCM operation. On the other hand, the main drawback of current approaches is either excessively expensive or not capable. The proposed technique provides an efficient way to develop circuit techniques for high-speed operation on a less supply voltage. It is usually considered that low power circuits are extremely slow and high-speed designs needed very high power utilization [1]. In several realistic applications of DSP, the sampling rate is varied either rising it or falling it by a few amounts. It desires to send different signals at different rates by analogous signal bandwidth. Multirate techniques benefited significantly in the digital audio area in current situations.

1.1 Need of Multirate DSP

Multirate DSP is necessary for a digital system where more number of sampling rate is necessary. The digital auditory system used the different sampling rates for broadcasting, compact disc, and audiotape as 32 kHz, 44.1 kHz, and 48 kHz. Therefore, while moving the recorded music to compact discs (CDs), audio professional wants to make a rate translation. The digital video system requires the sampling rate to joint video signals are 14.317 MHz and 17.735 MHz [1]. These signals are received in video receivers through a sampling rate converter. Multirate DSP is used in a digital transmission system where the data is managing with different rates. Multirate concept is used to decrease the storage space and the broadcast rate in speech processing. FIR Filter is broadly useful in a multi-standard wireless communication system. The FIR filter is used in a wireless communication system for channel equalization, pulse shaping, channelization and matched filtering. FIR filter is usually designed in the transposed form due to its better performance and power efficiency. The multiplier block of digital filter in its invert form is depicted in Fig. 1 and 2:

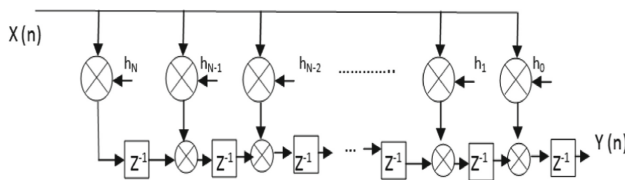


Fig. 1. FIR filter transposed form

Therefore, the filter parameters are multiplied by the input data and implemented under shifting adding concept. Each block contains addition, subtraction and shift operation [2]. Hence, in recent year digit serial systems become more useful. The filter parameters are multiplied by the input data and implemented under shifting adding concept.

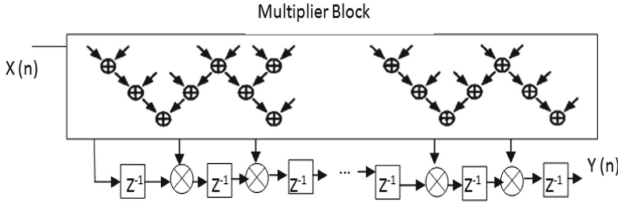


Fig. 2. Invert form with MCM

1.2 Polyphase Decimator

Decimation and interpolation are the two fundamental operations in a Multirate system. The Polyphase arrangement is one more capable filter for decimation like a direct type Decimator. In Interpolation and decimation, the normalize conversion bandwidth relates to the decimation part and the interpolation part. Polyphase is an approach of doing sampling rate translation which leads to very proficient Design. The sampling rate is reduced for well-organized communication, and a sampling rate is increased for the speech restoration. The sampling rate decrease in Polyphase Decimator is depicted in Fig. 3,

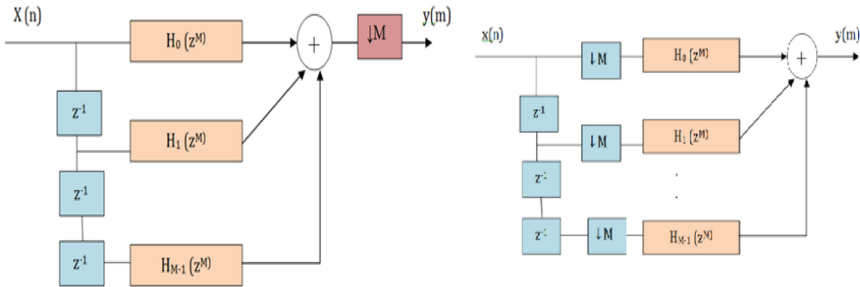


Fig. 3. Polyphase representation of decimator

1.3 Polyphase Interpolator

In interpolation, filter is used to suppress aliasing and eliminate imaging. The interpolator utilize to increase the sampling rate as depicted in the Fig. 4.

The FIR filter efficiency of sampling rate translation is extensively enhanced using the Polyphase design. The filter is used in the interpolation process and a polyphase arrangement to achieve concurrently the interpolation with a specified factor [4]. The normalize transition bandwidth inversely relates to the interpolation factor L in interpolation Multirate filters. The order of an interpolation filter changes as L and perfectly implemented with the FIR filter.

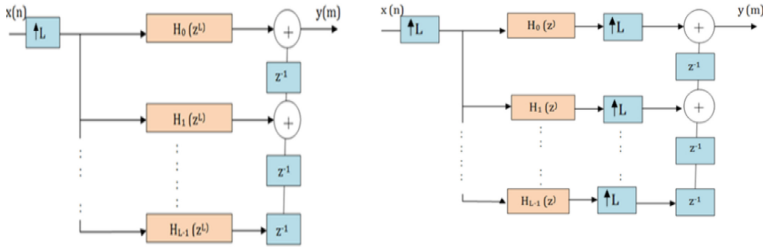


Fig. 4. Block diagram representation of Polyphase Interpolator

The FIR filter efficiency of sampling rate translation is extensively enhanced using the Polyphase design. The filter is used in the interpolation process and a polyphase arrangement to achieve concurrently the interpolation with a specified factor [4]. The normalize transition bandwidth inversely relates to the interpolation factor L in interpolation Multirate filters. The order of an interpolation filter changes as L and perfectly implemented with the FIR filter.

2 Methodology

FIR filters are critical to design due to a more number of multipliers. Proposed techniques concentrated on decomposing of these multipliers by subtraction, addition, and shifting. The adders are reduced using the following concepts; consider the multiplication of multiple constants $11\times$ and $13\times$. It is observed that the multiplier realization without fractional product sharing needed four operations. It is depicted as Fig. 5.

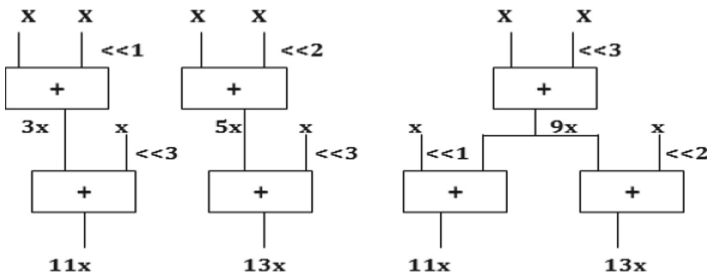


Fig. 5. Multiplier realization without fractional product

Generally, additions and subtractions have a comparable complication in the design. Let us consider the constant multiplications $29\times$ and $43\times$. It is observed that the sharing of fractional products $3\times$ and $5\times$ reduces the operations from six to four. It is given in Fig. 6.

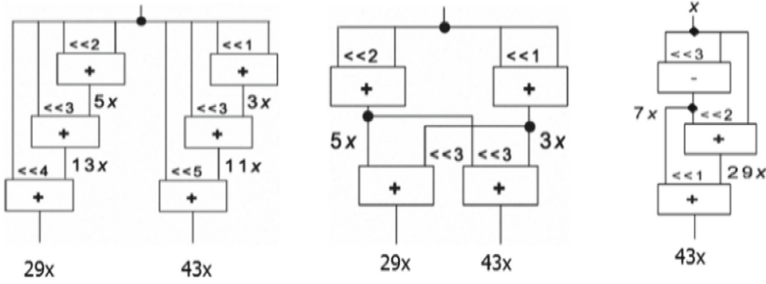


Fig. 6. Shift add implementation without partial product and with partial product

Let, the bit-parallel method applied single complete word of the input sample in the single clock cycle is best for high-speed applications (Fig. 7).

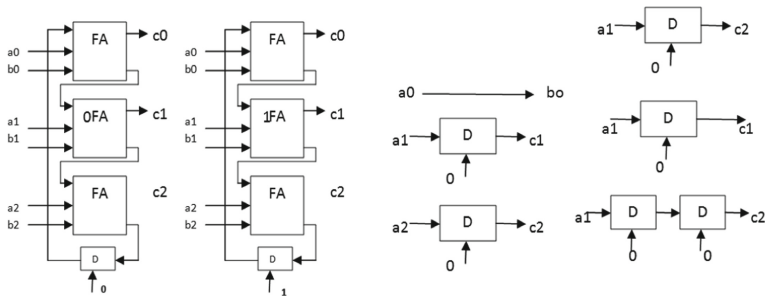


Fig. 7. Digit serial operations

3 Result

3.1 FIR Filter

Initially, the Finite impulse response Filter is considered in invert form with the shifting-adding concept which decreases power utilization and area. It is depicted in the Fig. 8, Fig. 9 and Fig. 10.

Then, the proposed concept is used to design the Finite impulse response Filter in invert form by maintaining several multipliers equal and replace the adders by transpose strategy. It reduces the area but increases the power dissipation.

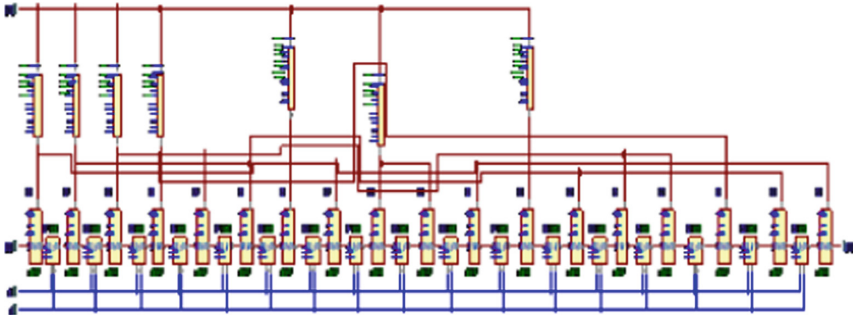


Fig. 8. Register Transfer Logic (RTL) View of FIR filter with the shifting-adding concept

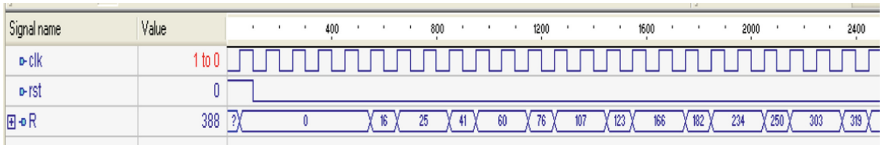


Fig. 9. Simulation waveform

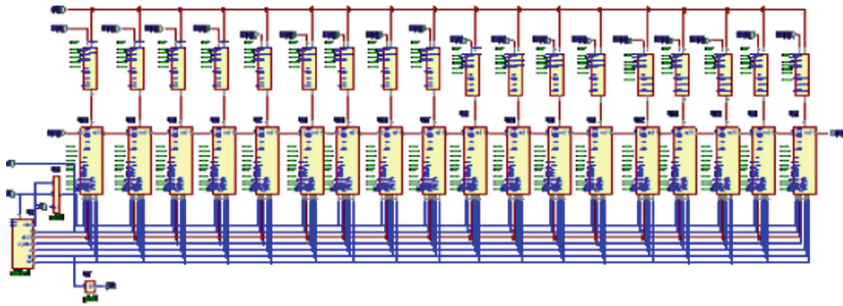


Fig. 10. RTL View of FIR Filter with Digit Serial Adder

Then, the proposed concept is used to design the Finite impulse response Filter in invert form by maintaining several multipliers equal and replace the adders by transpose strategy. It reduces the area but increases the power dissipation.

Finally, Multiple constant multipliers with a digit-serial adder reduce power and area by a large value. In the design, multiple constant multiplier blocks use a shifting-adding concept. The shifting unit does not occupy area hence the entire cell region is decreased. Also, the adder block area is decreased by digit-serial design. This method required only two adders instead of 32 adders. Therefore, it is the most efficient idea which improves area and power consumption by huge value maintaining high speed (Fig. 11, 12 and 13).



Fig. 11. Simulation waveform

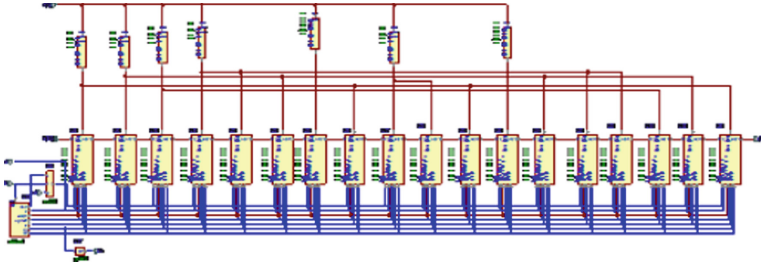


Fig. 12. RTL View using MCM, Digit Serial and shifting-adding.



Fig. 13. Simulation waveform

The Multirate FIR filter is implemented and results are verified using Field Programmable Gate Array (FPGA) as depicted in Fig. 14.

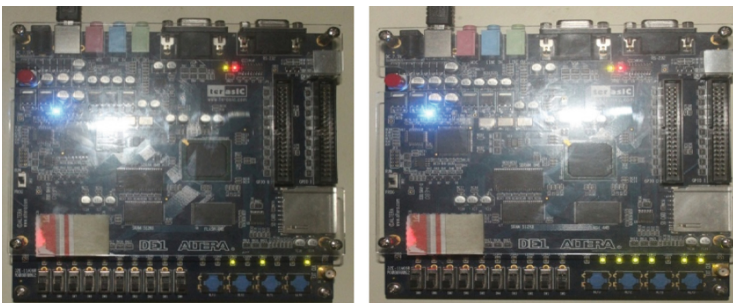


Fig. 14. Verification of Result 101001 and 1111011 of Multirate FIR Filter

The Table 1 specified comparative analysis of essential parameters of the Multirate FIR filter with various optimization techniques (Fig. 15).

Table 1. Testing result with various techniques

Filter	Type	Cell area (μm^2)	Dynamic power (mW)	Speed (MHz)
FIR Filter	0	28454.5976	5.828	32.62
FIR Filter	1	34626.3618	12.785	—
FIR Filter	2	20999.7668	2.345	194.66
FIR Filter- digit serial	0	28836.1385	5.890	—
Using MCM-digit serial	1	13424.3267	7.313	—
MCM-digit serial adder–shift adds	1	988.81509	0.1492	153.79

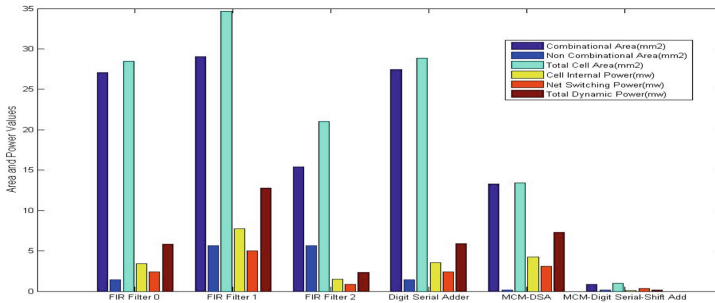


Fig. 15. Area-power-speed improvement graph

3.2 Polyphase Decimator

The technique multiple constant multiplications with digit-serial adder propose less complication in the circuit, require small area, less power dissipation by keeping greater speed. The effectiveness of the projected method and the investigation of dissimilar design have shown by testing results. The Fig. 16 shown Multirate Polyphase Decimator using the MCM technique (Fig. 17).

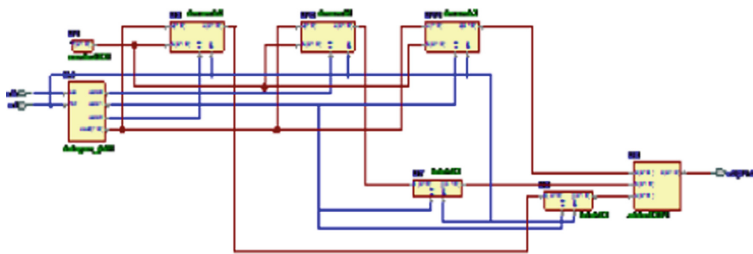


Fig. 16. RTL view of multirate polyphase decimator

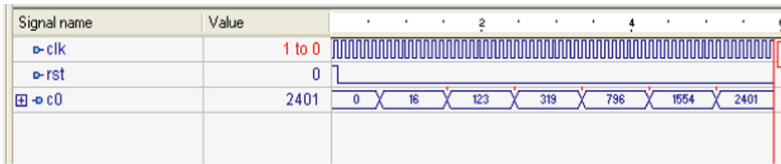


Fig. 17. Design waveform of polyphase decimator using MCM

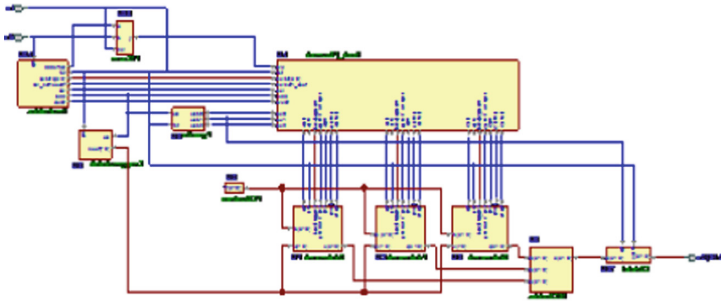


Fig. 18. RTL view of polyphase decimator using MCM-digit serial concept

Multirate Polyphase Decimator is designed using multiple constant multiplications with Digit Serial Architecture as depicted in Fig. 18 (Fig. 19).

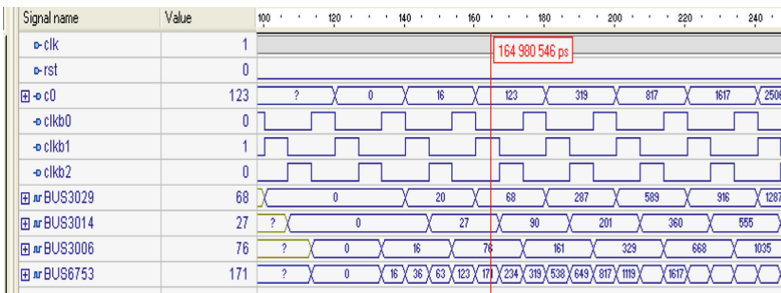


Fig. 19. Design waveform

The Multirate Polyphase Decimator is implemented and results are verified using FPGA as depicted in Fig. 20 (Table 2 and 21).

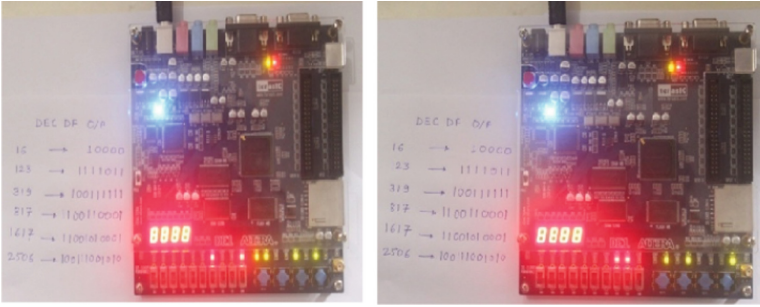


Fig. 20. Verification of result 100111001010 and 11001010001

Table 2. Multirate Polyphase decimator testing results

Type	Area (μm^2)	Power (μW)	Speed (MHz)
Direct Form	29740	70.5	190.6210
Transpose Form	22580	173.1	111.0251
Multiple constant multiplication	13257	180.2	100.6042
MCM and Digit Serial Adder	21271	330.1	151.5790

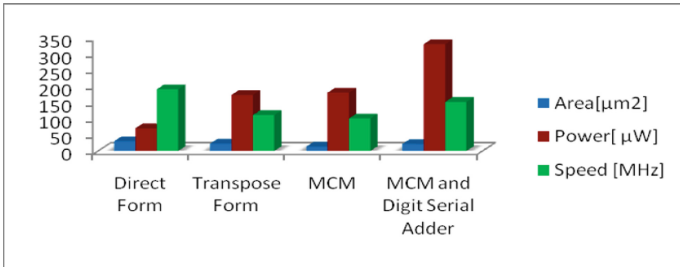


Fig. 21. Improvement graph

3.3 Polyphase Interpolator

The parameters of Multirate Polyphase Interpolator are improved by MCM-digit-serial adder which occupies less area, low power utilization maintaining higher speed. The efficiency of the proposed techniques has shown in the testing result.

The Fig. 22 shown a Multirate Polyphase interpolator using the MCM technique. This design consumed moderate power (Fig. 23).

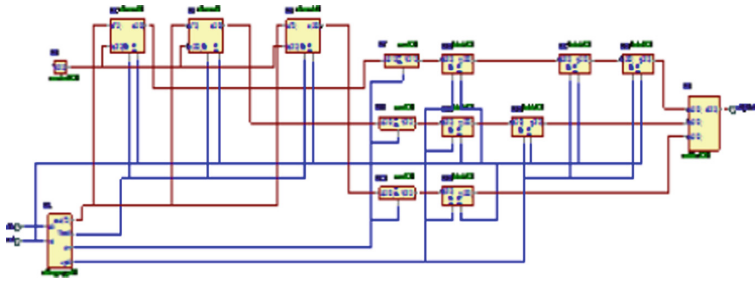


Fig. 22. RTL view of multirate polyphase interpolator

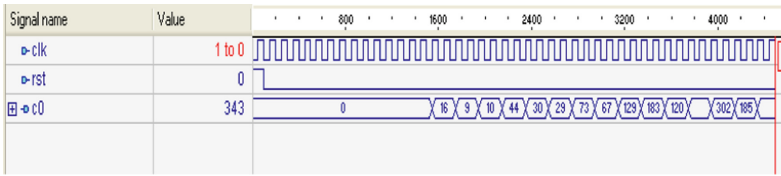


Fig. 23. Design waveform of multirate polyphase interpolator

Multirate Polyphase interpolator is considered with Multiple Constant Multiplication and Digit Serial Architecture as depicted in Fig. 24 (Fig. 25).

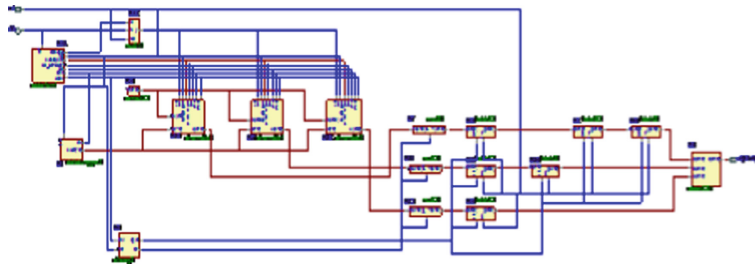


Fig. 24. RTL view using multiple constant multiplication and digit serial Adder concept

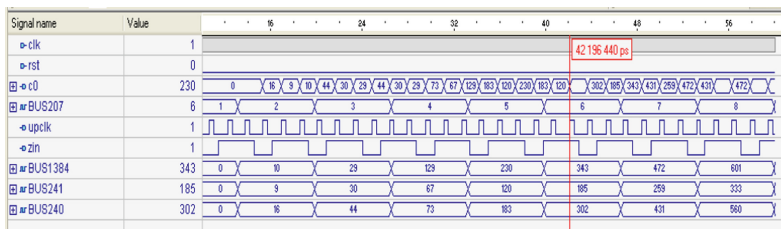


Fig. 25. Design waveform

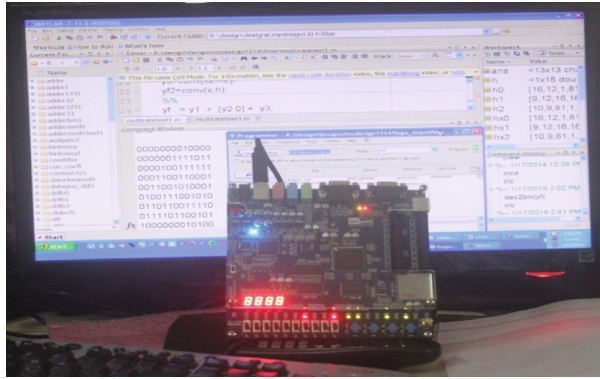


Fig. 26. Complete Setup of Result with verification

Results of the Multirate Polyphase Interpolator module are verified using FPGA. The complete setup of the design is depicted in Fig. 26 (Table 3 and Fig. 27).

Table 3. Testing Results with various techniques

Type	Area (μm^2)	Power (μW)	Speed (MHz)
Direct Form	31921	113	205.634
Transpose Form	37511	222	106.315
Using MCM	14747	220	103.189
Using MCM and Digit Serial Adder	23178	413	151.579

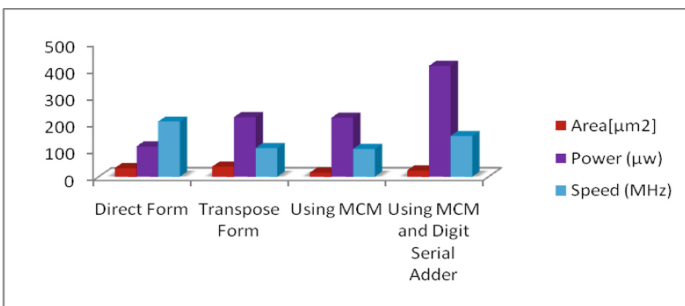


Fig. 27. Area-power-speed improvement graph

4 Conclusion

The proposed technique presented an efficient approach to derive less power, high-speed digital signal processing systems using Multirate. This method ensures the improvement in Power, Area, Speed and Complexity in the design. In FIR Filter, shifting and adding techniques decrease the area and power utilization to a great extent. On the other hand, Multirate Polyphase decimator with multiple constant multiplications and digit serial adder techniques overcome the difficulty of complication and occupying the very little area by keeping a higher speed. In Multirate Polyphase Interpolator design, the direct structure is suitable for the realization of a DSP System because it needed very low power consumption at a higher speed. The comparative results of the design parameters are given in the tabulation type which found acceptable. The Physical test of all designs is verified that implementation worked correctly.

References

1. Rajendra, M., Rewatkar, Badjate, S.L.: Optimization of FIR filter using digit serial architecture. In: IEEE Xplore IGCCE Conference, Chennai 14 December 2013
2. Aksoy, L., Lazzari, C., Costa, E., Flores, P., Monteiro, J.: Efficient shift-adds design of digit-serial multiple constant multiplications. In: GLSVLSI 2011, 2–4 May 2011, Lausanne, Switzerland (2011)
3. Thenmozhi, M., Kirthika, N.: Analysis of efficient architectures for FIR filters using common sub expression elimination algorithm. *Int. J. Sci. Technol. Res.* **1**(4), 422–435 (2012)
4. Gustafsson, O., Dempster, A.G.: On the use of multiple constant multiplications in polyphase FIR filters and filter. *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)* **1**(10) (2012)
5. Nguyen, H., Chattejee, A.: Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **8**(4), 419–424 (2000)
6. Aksoy, L., Costa, E., Flores, P., Monteiro, J.: Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **27**(6), 1013–1026 (2008)
7. Aktan, M., Yurdakul, A., Dündar, G.: An algorithm for the design of low-power hardware-efficient FIR filters. *IEEE Trans. Circuits Syst.-I: Regular Pap.* **55**(6), 1536–1545 (2008)
8. Aksoy, L., Lazzari, C., Costa, E., Flores, P., Monteiro, J.: Design of digit-serial FIR filters: algorithms, architectures, and a CAD tool. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **21**(3), 498–511 (2012)
9. Shahein, A., Zhang, Q., Lotze, N., Manoli, Y.: A novel hybrid monotonic local search algorithm for FIR filter coefficients optimization. *IEEE Trans. Circuits Syst.-I: Regular Pap.* **59**(3), 616–627 (2012)
10. Voronenko, Y., Püschel, M.: Multiplierless multiple constant multiplication. *ACM Trans. Algorithms* **3**(2), 1–39 (2007)



Simplified and Secure Session Key Sharing for Internet of Things (IoT) Networks

Krishan Pal Singh, Shubham Tomar, Usha Jain, and Muzzammil Hussain^(✉)

Department of CSE, Central University of Rajasthan, Ajmer, India
{2019mtcse005,2019mtcse009,2014phdcse03,mhussain}@curaj.ac.in

Abstract. Internet of Things is found everywhere in today's life, they are simplifying our life and are reliable. As they are connected to the Internet for communication, they are vulnerable to security attacks. Here, we have proposed a Key generation protocol for Internet of Things (IoT) devices for securing the communication between them. The proposed protocol is based on the discrete logarithm problem (DLP) and is secure. In the proposed mechanism the nodes/entities generate the session key and the communication is facilitated by server. The session key is generated by the individual nodes. The mechanism uses group theory and DLP. As the key is generated by the nodes and no exchange of key is done, the proposed mechanism is secure. It requires very fewer resources and is suitable for IoT networks.

Keywords: Internet of Things · Security · Session key · Key generation · Efficient mechanism

1 Introduction

“It’s about networks, it’s about devices, and it’s about data,” Caroline Gorski, the head of Internet of Things (IoT) at Digital Catapult explains. IoT allows devices on closed private internet connections to communicate with others and “the IoT brings the networks together and creates an opportunity for devices to communicate not only within close silos but across different networking types to connect the whole world.” [1]. Everything that’s connected to the internet can be hacked. So, Our major concern is to present a Safeguard to information authority through passkey cipher mechanism.

Wireless sensors and actuators, sensor data aggregation systems and analog-to-digital data conversion, the appearance of edge IT systems, analysis, management, and storage of data are essential laps of IoT [1].

It’s evident that the IoT is juvenile and soon the internet will be an essential part of our lives which will support life and transmit the data. So the need to secure every connected device is “very crucial” and looking forward in the future we can predict that industries will tickle in digital evolution, using IoT, AI and machine learning. The prospect of IoT is very largish and will increase, no pause, people will rely more on the Internet of Things [1]. As, IoT grows, threats on it will also grow. Association of an identity of an individual is a major threat, this may lead to profiling and tracking

and it also might be possible that a certain IoT device may try to find the location of an individual and track the individual and this may cause privacy issues [2].

The basic definition of IoT includes the fusion of dissimilar networks which contains the security threats of both traditional networks as well as some new threats due to its features. Our security objective with regard to IoT must be focused on availability, integrity, confidentiality, authentication, and non-repudiation and a key is only considered secure enough if it meets these objectives. Vulnerabilities in security mechanisms are exploited to expose the communication in IoTs which can be of two types - Active attacks and Passive attacks. In Passive attacks, the attacker listens and analyzes the traffic exchanged between objects without making any modification. This allows the attacker to determine useful information such as secrets or keys exchanged that can be used to launch active attacks. In Active attacks, the attacker modifies or removes the exchanged data or injects new messages [3].

To keep the communications secure and protect sensitive information, encryption is required and the secrecy of the keys is also essential. The key is used to establish trust between two entities and maintain the confidentiality of the communications between them. If the number of people who know the key increase then it is certain that the threat to our data increases too, hence the secrecy of the key is a must to boost the secure transmission of the data. Looking at the continuous and rapid growth in transmission of sensitive personal information such as PII (Personally Identifiable Information), health information and voice sample of an individual, a confidential and authenticated environment must be established between two entities. To genuinely regard the protection of communication in IoT at a high stake, we require a trusted environment to make sure only authenticated entities have access to the data i.e. to be transmitted. Hence, we require a secret session key. The key management in IoT includes the creation and renewal of the key after a session ends. It also involves managing the keys therefore it's the most crucial event in the whole transmission. If the key is compromised or exposed then it might cause serious damage to the confidential data [3, 4].

Here, We suggest a key generation mechanism that establishes a session key between two entities using a trusted server. The server provides a trusted and secure environment between the transmitter and the receiver before the secure session key is generated. The messages transmitted between a server and entities are encrypted with the symmetric key cryptography. While the key is being established, both the entities authenticate each other. Once the session key is generated, the secure communication session starts between the communicating entities using the key.

This paper has five sections with Introduction being Sect. 1. Section 2 is the Literature review, Proposed work is Sect. 3, Sect. 4 contains security analysis and Sect. 5 has the conclusion of our work.

2 Literature Review

Here, we review the related work done by researchers.

2.1 Station to Station Protocol [5, 14, 15]

This protocol exchanges signature with the message to add authentication to the renowned Diffie-Hellman protocol. Authentication can be done only when both the entities have safely shared their long-term asymmetric public keys.

Notation:

session key, $K=g^{xy}$

$S_x(.)$ =the signature by the principal X on the string in the brackets.

Successful protocol run is as follows.

$$\begin{aligned} A \rightarrow B &: A, B, g^X \\ B \rightarrow A &: B, A, g^Y, \{S_B(g^Y, g^X)\}_{K_{AB}} \\ A \rightarrow B &: A, B, \{S_A(g^X, g^Y)\}_{K_{AB}} \end{aligned}$$

Attacks involved:

I_x denotes I masquerading as principal X.

$$\begin{aligned} A \rightarrow I_B &: A, B, g^X \\ I \rightarrow B &: I, B, g^X \\ B \rightarrow I &: B, I, g^Y, \{S_B(g^Y, g^X)\}_{K_{AB}} \\ I_B \rightarrow A &: B, A, g^Y, \{S_B(g^Y, g^X)\}_{K_{AB}} \\ A \rightarrow B &: A, B, \{S_A(g^X, g^Y)\}_{K_{AB}} \end{aligned}$$

Intruder is doing little more than relaying each message that passes between A and B. What is the result? B does not indicate that A has engaged in the protocol and yet A has completed a successful run, apparently with B [19].

2.2 Needham - Schroeder Secret Key Protocol [6]

Needham - Schroeder protocol is used to share the session key secretly between the entities.

Successful protocol run as follow:

$$\begin{aligned} A \rightarrow S &: A, B, N_A \\ S \rightarrow A &: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\ A \rightarrow B &: \{K_{AB}, A\}_{K_{BS}} \\ B \rightarrow A &: \{N_B\}_{K_{AB}} \\ A \rightarrow B &: \{\text{dec}(N_B)\}_{K_{AB}} \end{aligned}$$

Authentication attack by Denning and Sacco [DS81].. Assume that Intruder has recorded the earlier sessions and the key K_{AB} is compromised. In step 3, B is convinced that he shares the secret key K_{AB} only with A [7, 8, 9].

$$\begin{aligned}
A \rightarrow S & : A, B, N_A \\
S \rightarrow A & : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\
A \rightarrow I(B) & : \{K_{AB}, A\}_{K_{BS}}
\end{aligned}$$

Assume that K_{AB} is compromised.

$$\begin{aligned}
I(A) \rightarrow B & : \{K_{AB}, A\}_{K_{BS}} \\
B \rightarrow I(A) & : \{N_B\}_{K_{AB}} \\
I(A) \rightarrow B & : \{\text{dec}(N_B)\}_{K_{AB}}
\end{aligned}$$

2.3 Session Key Sharing Using Public Key Cryptography [10]

This protocol uses public key mechanism to generate and share the secret session key.

$$\begin{aligned}
\{k_{UA}, k_{RA}\} & : \text{Public and Private keys of Entity A.} \\
\{k_{UB}, k_{RB}\} & : \text{Public and Private keys of Entity B.} \\
\{k_{US}, k_{RS}\} & : \text{Public and Private keys of Server S.}
\end{aligned}$$

$$\begin{aligned}
\text{Message 1. } A \rightarrow S & : \{\{A, B, N_A\}_{k_{RA}}\}_{k_{US}} \\
\text{Message 2. } S \rightarrow A & : \{N_A, A, B, k_{AB}, \{\{K_{AB}, A, N_A\}_{k_{RS}}\}_{k_{UB}}\}_{k_{UA}} \\
\text{Message 3. } A \rightarrow B & : \{\{k_{AB}, A, N_A\}_{k_{RS}}\}_{k_{UB}} \\
\text{Message 4. } B \rightarrow A & : \{N_A-1, N_B\}_{k_{AB}} \\
\text{Message 5. } A \rightarrow B & : \{N_B-1\}_{k_{AB}}
\end{aligned}$$

Alice initiates by sending a request to server S Which carries Id of Alice, Id of Bob and Nonce of Alice N_A . This complete message is encrypted using the Private key of Alice k_{RA} and again encrypted with the private key of the server k_{RS} . After receiving this message server decrypts it using its private key k_{RS} and Public key of Alice k_{UA} . The server generates the session key k_{AB} and forwards it to Alice who decrypts it by using its private key and finds the session key along with it also gets the component $\{\{k_{AB}, A, N_A\}_{k_{RS}}\}_{k_{UB}}$. Alice forwards this to Bob. Bob decrypts it using its private key and server's public key and finds the session key. Bob replies to Alice by decreasing the value of Nonce of Alice N_A along with Nonce of Bob N_B encrypted using session key k_{AB} . Alice decrypts this message using the session key and verifies the Nonce and responds to Bob by decreasing the value of Nonce of Bob N_B encrypted by session key hence session key is shared between Alice and Bob.

2.4 Yahalom Protocol [11]

The Yahalom protocol is used to share the symmetric key between any two entities which want to communicate with each other. This protocol requires a trusted server to generate the symmetric key.

A : Alice

B : Bob

S : Server

N_A : Nonce of A

N_B : Nonce of B

K_{AB} : Secret Session key between A and B

K_{AS} : Secret key between A and S

K_{BS} : Secret key between B and S

Message 1. $A \rightarrow B : A, N_A$

Message 2. $B \rightarrow S : B, \{A, N_A, N_B\}_{K_{BS}}$

Message 3. $S \rightarrow A : \{B, K_{AB}, N_A, N_B\}_{K_{AS}}, \{A, K_{AB}\}_{K_{BS}}$

Message 4. $A \rightarrow B : \{A, K_{AB}\}_{K_{BS}}, \{N_B\}_{K_{AB}}$

A initiates by sending the request to B along with its Id (A) and Nonce (N_A). B sends this message along with its Id (B) and Nonce(N_B) encrypted with K_{BS} (symmetric key between B and S). The server generates session key K_{AB} and sends it to A in Message 3. Now A decrypt the first component of the message by symmetric K_{AS} and verifies its Nonce and gets the key K_{AB} and nonce of B N_B . Now A sends the second component of the third message to B along with Nonce of B N_B encrypted with K_{AB} . Now by doing this B will also get the session key K_{AB} . Now A and B can start communication with this session key. There are two possible attacks in Yahalom protocol- Replay and Impersonation attack [11, 12].

3 Proposed Work

We are proposing a mechanism to establish a secure session key between two entities with a trusted server in the loop using the symmetric key approach. The session key is only generated by the communicating entities and security of the key is dependent upon DLP (Discrete logarithm problem) [13, 14, 15].

Discrete Logarithm Problem [13, 16].

For a cyclic group G with group operation by multiplication and generator g , its exponential is -

$$g^x = g.g.g \dots .x \text{ terms}$$

let $y = g^x$, then the discrete logarithm of y is x and written as $x = (\log y)_g$. The discrete logarithm of y is modulo n of exponent x , on the order of g in G . If $n = |g|$, then $(\log y)_g \equiv x \pmod{n}$.

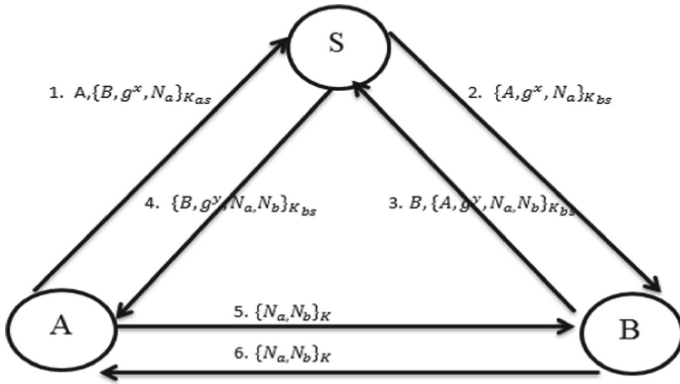


Fig.1. Proposed protocol

Prerequisites:

- K_{AS} : Secret key shared between Server S and Entity A.
- K_{BS} : Secret key shared between Server S and Entity B.
- K : Generated session key between A and B.
- g^X : Generator of Entity A.
- g^Y : Generator of Entity B.

Protocol:

- Message 1. $A \rightarrow S : A, \{B, g^x, N_A\}K_{AS}$
- Message 2. $S \rightarrow B : \{A, g^x, N_A\}K_{BS}$; $K = (g^x)^Y$
- Message 3. $B \rightarrow S : B, \{A, g^y, N_A, N_B\}K_{BS}$
- Message 4. $S \rightarrow A : \{B, g^y, N_A, N_B\}K_{AS}$; $K = (g^y)^X$
- Message 5. $A \rightarrow B : \{N_A, N_B\}K$
- Message 6. $B \rightarrow A : \{N_B, N_A\}K$

For A to generate a Secure session key to have a trusted communication environment with B, the process is followed -

Entity A generates a random number x to calculate generator g^x and sends a request message to Server S, which consists Id of A. It also consists Id of B, generator g^x and Nonce of A N_a encrypted with K_{AS} .

The Server S decrypts the message using K_{AS} and forwards the request message to Entity B. The message has Id of A, generator g^x and nonce of A N_a encrypted with K_{BS} .

Entity B receives the message and decrypts it using K_{BS} . Entity B reads the message and generate its generator g^y using a random number y. Using y Entity B also computes the key using the generator g^x it received in the message. The secure session key computed by Entity B is $K = (g^x)^y$. Then Entity B sends response message to server S which consists Id of B along with Id of A, generator g^y , Nonce of A N_a and it's own Nonce N_b encrypted with K_{BS} .

Server S receives the message and decrypts it using K_b and forwards the message to A encrypting it using K_a . The message consists of B, Nonce of A, generator g^y and Nonce of B N_b .

Entity A receives the message and decrypts it using K_a . Using its random number x Entity A computes the exponent of generator g^y as $(g^y)^x$ which is the secret session key K between Entity A and Entity B. Entity A then sends a message to Entity B consisting of N_a and N_b encrypted with K in the exact order.

Entity B receives the message from and decrypts it using K . It again encrypts a message with key K . Message consists of N_b and N_a and Entity B sends it to Entity A. A decodes the message and A and B verify each other after that the Secure communication between A and B starts.

Our Proposed Key generation method is based on symmetric key encryption therefore it requires less storage, it has fewer messages transmitted among entities and server. It's also a little less complex than asymmetric key encryption based mechanisms [10, 20].

4 Security Analysis

The security analysis of the proposed protocol is done using SPAN + Avispa tool.

Avispa is a tool which is used to analyse the security of the protocols. SPAN stands for Security. Protocol Animator for Avispa [17, 18].

A. Environment

Version	:	SPAN + Avispa 1.6
Tool type	:	Protocol Analysing Tool
Modelling language Language)	:	HLPSL(High Level Protocol Specification
Back-end	:	OFMC(On-the-fly Model-Checker)
Roles	:	Alice , Bob and Server

The protocol is designed with two composite and three basic roles.

Composite role	:	Session and Environment
Basic role	:	Alice , Bob , Server

The role of Alice is acted by agent A with the help of k_{AS} similarly the role of Bob is acted by agent B with the help of k_{BS} . The role of the Server is acted by agent S with key k_{AS} and k_{BS} .

Notation for Communication Channel

Sending message: SND_XY (where Y is the sending entity and X is the receiving entity).

Receiving message: RCV_XY (where X is the sending entity and Y is the receiving entity).

A and B have same generator (G) where A is initialised with state 0. Similarly B and S are initialised with state 5 and 1 respectively.

In transition section of all the roles, received and sent messages are defined.

In the composition section of role session, instances of all three basic roles are created, intruder knowledge and session composition are defined in the role environment.

Goals are defined in the goal section.

Goals : Secrecy of session key k,

Authentication on k,

Authentication on N_A ,

Authentication on N_B

```

|
role alice(A,S,B
    : agent,
    SND_S,RCV_S,SND_B,RCV_B : channel(dy),
    Kas      : symmetric_key,
    G        : nat)

played_by A
def=
    local State      : nat,
           X,Y       : nat,
           K          : message,
           Na,Nb     : text

init State := 0

transition
1.   State = 0 /\ RCV_S(start) =|>
     State' := 4 /\ X' := new()
                               /\ Na' := new()
                               /\ SND_S(A.{B.exp(G,X')}.Na')_Kas)
2.   State = 4 /\ RCV_S({B.exp(G,Y')}.Nb'.Na}_Kas) =|>
     State' := 6 /\ K' := exp(exp(G.Y'),X)
                  /\SND_B({Na.Nb'}_K')
                  /\ secret(K',k,{A,B})
                  /\ request(A,B,na,Na)
                  /\ request(A,B,alice_bob_k,K)
3.   State = 6 /\ RCV_B({Nb'.Na}_K) =|>
     State' := 8 /\ witness(B,A,nb,Nb')

end role

```

Fig. 2. HPSL code for role Alice.

If specified goals are fulfilled then it implies that the protocol is safe otherwise unsafe. The modeling of the proposed protocol is shown in Figs. 2, 3, 4, 5 and 6.

B Results generated

The protocol is modeled in HLPSL and executed in SPAN + Avispa while OFMC checks for the possible attacks and verifies the model by running the sessions for bounded number of times [17]. But every message can be seen by the intruder in Dolev-Yao channel. The protocol has achieved all the goals specified in the goal section and runs safely. Figure 7 shows the output generated by AVISPA and Fig. 8 and Fig. 9 show the simulation of the proposed protocol. It's evident that the protocol is SAFE and its execution is complete i.e. without errors.

```

role bob(B,S,A : agent,
        SND_S,RCV_S,SND_A,RCV_A : channel(dy),
        Kbs      : symmetric_key,
        G        : nat)
played_by B
def=
    local State      : nat,
          Y,X        : nat,
          K          : message,
          Na,Nb      : text

init State := 2
transition
1.      State = 2 /\ RCV_S({A.exp(G,X').Na'}_Kbs) =|>
        State' := 5 /\ Y' := new()
          /\ Nb' := new()
          /\ SND_S(B.{A.exp(G,Y').Nb'.Na'}_Kbs)
          /\ K' := exp(exp(G,X'),Y')
          /\ secret(K',k,{B,A})
          /\ witness(B,A,na,Na')
          /\ witness(B,A,alice_bob_k,K)
2.      State = 5 /\ RCV_A({Na.Nb'}_K') =|>
        State' := 8 /\ SND_A({Nb'.Na}_K)

end role

```

Fig. 3. HLPSL code for role Bob.

```

role server ( S,A,B      : agent,
              SND_A,RCV_A,SND_B,RCV_B : channel(dy),
              Kas,Kbs : symmetric_key,
              G       : nat)
played_by S
def=
    local State      : nat,
           X,Y       : nat,
           K         : message,
           Na,Nb     : text
init State := 1
transition
1.   State = 1 /\ RCV_A(A.{B.exp(G,X').Na'}_Kas) =|>
     State' := 3 /\ SND_B({A.exp(G,X').Na'}_Kbs)
2.   State = 3 /\ RCV_B(B.{A.exp(G,Y').Nb'.Na'}_Kbs) =|>
     State' := 7 /\ SND_A({B.exp(G,Y').Nb'.Na'}_Kas)
end role

```

Fig. 4. HLPSSL code for role Server.

```

role session(A,S,B      : agent,
             Kas,Kbs : symmetric_key,
             G       : nat)
def=
    local SND_S,RCV_S,SND_A,RCV_A,SND_B,RCV_B      : channel(dy)
composition
    alice(A,S,B,SND_S,RCV_S,SND_B,RCV_B,Kas,G)
    /\ server(S,A,B,SND_A,RCV_A,SND_B,RCV_B,Kas,Kbs,G)
    /\ bob(B,S,A,SND_S,RCV_S,SND_A,RCV_A,Kbs,G)
end role

```

Fig. 5. HLPSSL code for session.

```

role environment()
def=
    const a,s,b      : agent,
           kas,kbs,kis : symmetric_key,
           g         : nat,
           na,nb,k,alice_bob_k : protocol_id
    intruder_knowledge = {a,s,b,g}
composition
    session(a,b,s,kas,kbs,g)
    /\session(a,i,s,kas,kis,g)
    /\session(i,b,s,kis,kbs,g)

end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

goal

    secrecy_of k
    authentication_on na
    authentication_on alice_bob_k
    authentication_on nb

end goal

```

Fig. 6. HPSL code for environment and goals.

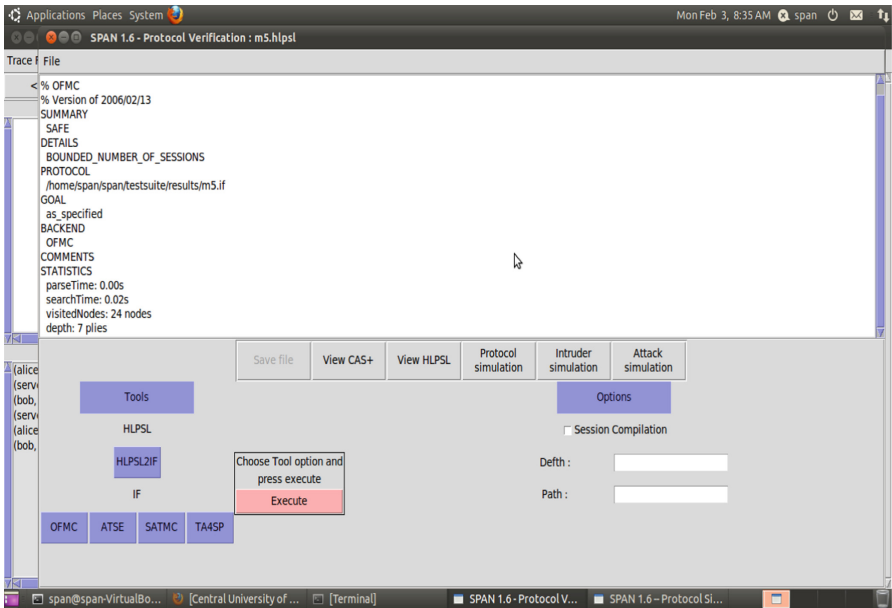


Fig. 7. Protocol run in back-end OFMC.

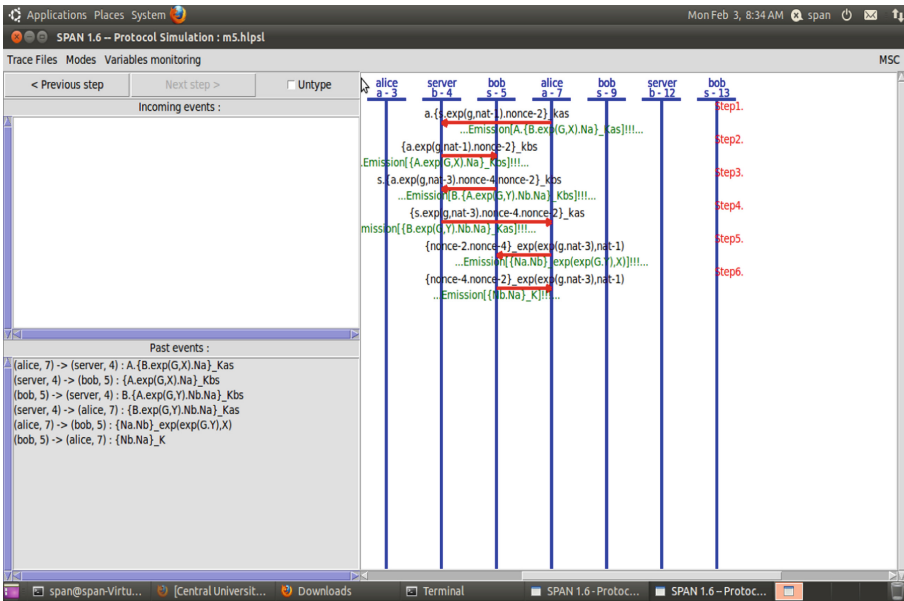


Fig. 8. Simulation of proposed protocol in a 'real message' mode.

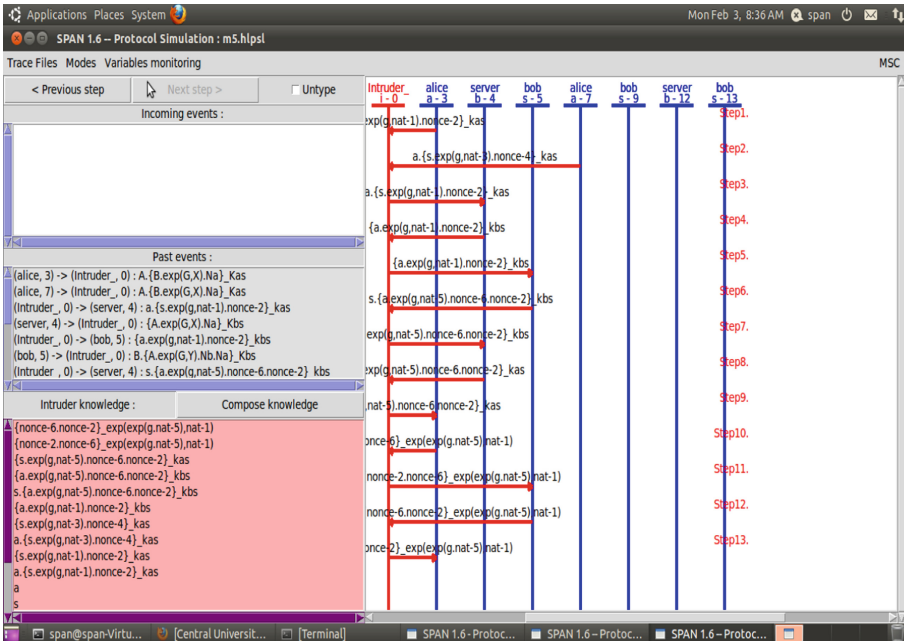


Fig. 9. Simulation of proposed protocol in a 'sender pattern message' and 'receiver pattern message' mode.

5 Conclusion

IoT is an emerging area and is producing convenience, the comfort of life, security, accessibility, and reliability to the users. As usual, security is a major challenge in the design of IoT systems as they are more vulnerable to security attacks. A secure key generation mechanism can ensure the confidentiality of communication and also improves reliability. Here, we proposed a key generation mechanism that securely generates and shares a session between IoT devices at less expense of energy, time and resources. The proposed mechanism is not only simple but secure as well. It establishes a secure common channel between IoT devices.

References

1. <https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot>
2. Hameed, S., Khan, F.I., Hameed, B.: Understanding security requirements and challenges in internet of things (IoT): a review. *J. Comput. Netw. Commun.* 1–14 (2019). <https://doi.org/10.1155/2019/9629381>
3. Naoui, S., Elhdhili, M., Saidane, L.: Security analysis of existing IoT key management protocols, pp. 1–7 (2016). <https://doi.org/10.1109/AICCSA.2016.7945806>
4. Sung, S.: Key management for secure internet of things (IoT) data in cloud computing. *J. Korea Inst. Inf. Secur. Cryptol.* **27**, 353–360 (2017). <https://doi.org/10.13089/JKIISC.2017.27.2.353>
5. <http://archive.dimacs.rutgers.edu/Workshops/Security/program2/boyd/node13.html>
6. Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. *Commun. ACM* **21**(12), 993–999 (1978)
7. Denning, D.E., Sacco, G.M.: Timestamps in key distribution protocols. *Commun. ACM* **24**(8), 533–536 (1981)
8. Lowe, G.: An attack on the Needham-Schroeder public-key authentication protocol. *Inf. Process. Lett.* **56**(3), 131–133 (1995)
9. Nesi, M., Rucci, G.: Formalizing and analyzing the Needham-Schroeder symmetric-key protocol by rewriting. *Electron. Notes Theoret. Comput. Sci.* **135**(1), 95114 (2005)
10. Gupta, A., Hussain, M.: Secure session key sharing using public key cryptography. In: Nair, I. (ed.) *Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI 2015)*, pp. 573–576. ACM, New York (2015)
11. Burrows, M., Abadi, M., Needham, R.: A logic of authentication. *ACM Trans. Comput. Syst.* **8**, 18–36 (1990)
12. Chen, L., Shi, M.: Security analysis and improvement of Yahalom protocol. In: *2008 3rd IEEE Conference on Industrial Electronics and Applications*, Singapore, pp. 1137–1140 (2008)
13. Menezes, A.J., Vanstone, S.A., Van Oorschot, P.C.: *Handbook of Applied Cryptography (Special Indian Edition)*. CRC Press Inc, Boca Raton (2010)
14. ANSI X9.42, “Agreement of Symmetric Algorithm Keys Using Diffie-Hellman”, working draft, May 1998
15. Blake-Wilson, S., Johnson, D., Menezes, A.: Key agreement protocols and their security analysis. In: Darnell, M. (eds.) *Cryptography and Coding*. *Cryptography and Coding*, pp. 30–45. *Lecture Notes in Computer Science*, vol. 1355. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0024447>
16. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **31**, 469–472 (1985)

17. Viganò, L.: Automated security protocol analysis with the AVISPA tool. *Electronic Notes Theoret. Comput. Sci.* **155**, 61–86 (2006)
18. Chevalier, Y., et al.: A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols. (Intern report) A04-R-067, chevalier04b 2004, p. 17, (inria-00100219)
19. Blake-Wilson, S., Menezes, A.: Unknown key-share attacks on the station-to-station (STS) protocol. Technical report CORR 98–42, University of Waterloo (1998)
20. Arora, S., Hussain, M.: Secure Session Key Sharing Using Symmetric Key Cryptography, pp. 850–855 (2018). <https://doi.org/10.1109/ICACCI.2018.8554553>



Machine Learning Based Network Slicing and Resource Allocation for Electric Vehicles (EVs)

Rohit Kumar Gupta¹, Anurag Choubey^{1(✉)}, Shlok Jain^{2(✉)},
R. R. Greeshma^{2(✉)}, and Rajiv Misra^{1(✉)}

¹ Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801103, India

{anurag.pcs17,rajivm}@iitp.ac.in

² Department of Computer Science and Engineering, Global Academy of Technology Bangalore, Bengaluru, India

Abstract. The introduction of electric vehicles (EVs) brings out various challenges like deploying more charging stations, building its supportive infrastructure, managing the EVs and their various resource requests. In order to maintain all the resource requests of the EVs, network slicing is used which provides an efficient way to satisfy various use case demands of the EVs. In this work, we perform network slicing that partitions the physical network into three slices, infotainment and safety message slices that belong to downlink communication and charge state information slice that belongs to uplink communication. If there are large number of resource requests made by the EVs to the slices, it might lead to collision in the channel. Unsupervised machine learning is performed on the EVs using local scaling as the scaling parameter which handles multi scale data and also performs better clustering. For an efficient communication between the clustered EVs and charging station, slice leaders of every cluster is determined. We have also proposed an algorithm that efficiently performs resource allocation to the EVs to increase the throughput with low latency. Slice leaders forward the resource requests made by the EVs of the respective clusters to the charging station through RSUs and a slice block allocation is performed by giving higher preference to the critical message requests.

Keywords: Network slicing · Latency · Throughput · Uplink · Downlink · EVs · Unsupervised machine learning

1 Introduction

In recent years, there is an inevitable transition from conventional vehicles to electric vehicles (EVs) as they provide ecological and reliable solution. As the automotive industry is transitioning from gasoline based vehicles to electric vehicles, the key challenge that the customer face which makes them reluctant to

purchase an EV is range anxiety. EV relies on the foundation of infrastructures, such as charging station, which is one of the most important factors of exploiting market for EV and building its network is necessary for enabling EVs to efficiently communicate with the charging stations for easy accessibility and time efficiency [5].

As the automotive industry is progressing towards autonomous vehicles, the vehicles are getting smarter and more connected to each other. Vehicle to Vehicle (V2V) communication and Vehicle to Infrastructure (V2I) communication play an important role in autonomous vehicles. V2V, V2I and V2P together constitute V2X which stands for Vehicle to everything communication which is deployed using 5G. The autonomous EVs communicate with each other sharing valuable information such as vehicular speed, destination, current location etc. The autonomous EV's can help tackle the heterogeneous traffic conditions by platooning and provide safety assists by detecting non line of sight vehicles and avoid collision. Ultra Reliable low latency communication (URLLC) [3] is a key enabler of autonomous EV's vision. Autonomous vehicular technology is over a decade old and was initially deployed using an IEEE 802.11p based technology termed as Delayed short range communication (DSRC) which had a range of 150 m with high latency which was unable to provide enough time for critical messages to be transmitted and avoid collision. Cellular V2X overcomes the shortcomings by providing double the range which provides more time for critical messages to be broadcast and critical actions to take place.

The main objective of this paper is performance analysis of heterogeneous V2X communication network and resource allocation operating in uplink and downlink direction. Infotainment and safety message services which are a part of enhanced mobile broadband (eMBB) and URLLC respectively in the downlink direction, charge state information which is part of URLLC in the uplink direction are the 5G use cases in the considered heterogeneous services. It also includes allocation of EVs to the charging stations with minimum collision and maximum usage of the network, using network slicing which is considered as one of the most significant technology challenges for 5G mobile networking infrastructure [10]. This can be achieved by partitioning the EVs based on certain factors into different clusters and allocating the clusters to their nearest charging station. This well-known problem is NP-hard [8] and also performs resource allocation for the various network slices [10]. This study considers infotainment packets which evolve at road side unit (RSU), safety message packets evolving at the slice leaders \mathcal{L} and charge state information packets which evolve at every EV. The RSU provides infotainment service and performs unsupervised machine learning to form clusters on the EVs and determines their respective slice leaders \mathcal{L} that relay the data to the EV's with weak QoS. The EV's transmit their charge state information to the RSU's which broadcast it to the connected charging stations.

Paper Organisation. In Sect. 2, we review the literature survey of resource allocation with vehicular communication and network slicing scenarios. Section 3

discuss system model i.e. sliced EVs. In Sect. 4, we discuss the methodology of electric vehicle spectral clustering via network slicing and presented the slice leader selection algorithm based on each clusters. Further in Sect. 5, we presents resource algorithm based on uplink and downlink. In Sect. 6, we present the experimental results of our work. Finally we conclude our work followed by future directions of further work in Sect. 7.

2 Related Work

2.1 Network Slicing

Network Slicing is one of the promising technology that has been used in various communication networks. It enables logical networks over a shared infrastructure to be customized cost effectively [15]. Realizing the potential of 5G networks has network slicing at its core. It is based on Network Function Virtualization (NFV) and Software-Defined Networking (SDN) driven queueing strategies that are used to enable communication with EVs with low SINR. Rapid network innovations in vehicular communications are enabled by architectures based on SDN [6]. In [5] authors H. Khan *et al.* have considered a scenario of autonomous vehicles in which they model a highway scenario with vehicles having heterogeneous traffic demands. Services involving vehicles exchanging data with infrastructure, with each other and any communicating entity can improve transport fluidity and safety by using 5G network slice(s) that are customized for vehicle-to-everything services [3]. We have formulated a network clustering and slicing algorithm to partition the vehicles into different clusters and allocate slice leaders to each cluster. We have related the idea of spectral clustering in our scenario for electric vehicles and have used a suitable scaling method called as local scaling [13] in which they propose that a local scale method should be incorporated to find the affinity between any two points.

2.2 Resource Allocation

The capacity of 5G networks and flexibility of network resource allocation based on network slicing can be improved by using efficient resource allocation schemes [14]. Flexibility and scalability provided by the 5G networks ensures that the network environment of a particular use case can be adopted appropriately by the mobile network [11]. Slice allocation in 5G means allocating resources throughout the network. 5G network can be modeled as multiple set of resources and spread across the whole physical network [4]. The key challenge faced is the computation of the best resource provisioning for the network slices in an efficient and robust manner [7]. When the EVs move from one domain to another, the service and communication sessions of EVs might suffer a drop in QoS [1]. These challenges can be overcome by using Network optimization algorithms and control techniques [12]. A distribution algorithm is proposed in [9] for resource allocation in uplink 5G networks where resources are allocated to

the uplink transmitters which results in higher spectral efficiency and maximize the users' data rate. We have related this scenario for resource allocation to the electric vehicles in both uplink and downlink communications.

There are large number of EVs present in a network. When multiple EVs wish to communicate with the charging station in a same channel, the collision increases and data rate decreases. Hence we have proposed algorithms that reduces the number of collisions within the channel and increases the data rate.

3 System Model

The scenario considered consists of three lane highway where EVs (\mathcal{E}) are moving in a direction with RSUs situated along the highway at equal distances. RSUs are further divided into RSU-CR and RSU-CS. RSU-CR is connected to the vehicles and RSU-CS is connected to the charging station. The EVs connected to the RSU-CR are clustered and the vehicles with highest V2V and V2I links are chosen as slice leaders of their respective clusters. Slice leaders forwards the safety message information to the clustered EVs. RSU-CR forwards the infotainment packets to the EVs in the downlink direction whereas RSU-CS forwards the charged state information from EVs to the charging station in the uplink direction.

Table 1. Notations

\mathcal{X}	Set of Charging Stations (CSs)
\mathcal{M}	Set of Road Side Units for Charging Station (RSU-CS)
\mathcal{N}	Set of Road Side Units for Cloud RAN (RSU-CR)
\mathcal{E}	Set of Electric Vehicles (EVs)
\mathcal{L}	Set of Slice Leaders (SLs)
\mathcal{K}	Set of Clustered EVs
\mathcal{R}^{CS}	Set of Charging Station (CS) resources
\mathcal{R}^{RSU-CS}	Set of RSU-CS resources
\mathcal{R}^{RSU-CR}	Set of RSU-CR resources
\mathcal{R}^{SL}	Set of Slice-Leader (SL) resources

The system model consists of a set of electric vehicles (\mathcal{E}) which are clustered to form \mathcal{K} clusters. Each cluster has a slice leader \mathcal{L} which communicates to the charging station (\mathcal{X}) through the Road Side Unit of Cloud-RAN (\mathcal{N}). \mathcal{N} is further connected to Road Side Unit of Charging Station (\mathcal{M}) which handles the set of charging stations (\mathcal{X}). Vehicles in the network can be categorized as

slice leaders(SLs) and clustered vehicles. RSUs, SLs and CSs have their set of resources $\mathcal{R}^{\mathcal{RSU}}$, $\mathcal{R}^{\mathcal{SL}}$, $\mathcal{R}^{\mathcal{CS}}$ respectively as shown in Table 1.

System Model Variables

$h_{x,m}^r$: Channel from CS x to RSU-CS m over subcarrier $r \in \mathcal{R}^{\mathcal{CS}}$
$h_{m,n}^r$: Channel from RSU-CS m to RSU-CR n over subcarrier $r \in \mathcal{R}^{\mathcal{RSU-CS}}$
$h_{n,l}^r$: Channel from RSU-CR n to SL l over subcarrier $r \in \mathcal{R}^{\mathcal{RSU-CR}}$
$h_{l,k}^r$: Channel from SL l to Clustered-EV k over subcarrier $r \in \mathcal{R}^{\mathcal{SL}}$
$y_{x,m}^r$: Signal received from the CS x to RSU-CS m
$y_{m,n}^r$: Signal received from the RSU-CS m to RSU-CR n
$y_{n,l}^r$: Signal received from the RSU-CR n to SL l
$y_{l,k}^r$: Signal received from the SL l to clustered EV k
$x_{x,m}^r$: Signal transmitted from the CS x to RSU-CS m
$x_{m,n}^r$: Signal transmitted from the RSU-CS m to RSU-CR n
$x_{n,l}^r$: Signal transmitted from the RSU-CR n to SL l
$x_{l,k}^r$: Signal transmitted from the SL l to clustered EV k

We consider the channel between the EVs and the charging stations where there is an uplink and downlink communication. In uplink communication, there is a channel $h_{x,m}^r$ from CS x to RSU-CS m where x transmits a signal $x_{x,m}^r$ to m and receives signal $y_{x,m}^r$. For communication between the RSUs of uplink and downlink, there is a channel $h_{m,n}^r$ from RSU-CS m to RSU-CR n where m transmits a signal $x_{m,n}^r$ to n and receives signal $y_{m,n}^r$. In downlink communication, there is a channel $h_{n,l}^r$ from RSU-CR n to SL l where n transmits a signal $x_{n,l}^r$ to l and receives signal $y_{n,l}^r$. For communication between the clustered EVs k and SL l , there is a channel $h_{l,k}^r$ where l transmits a signal $x_{l,k}^r$ to k and receives a signal $y_{l,k}^r$.

In this system model we consider orthogonal frequency distribution multiple access(OFDMA) modulation scheme for transmission. This study consists of network which has a set of electric vehicles \mathcal{E} , RSUs and charging stations(CSs). RSUs are categorized for charging stations as RSU-CS and for Cloud RAN as RSU-CR. Clustering of EVs in the network can be represented as (Table 2):

$$E = L \cup K \quad \text{with} \quad |E| = |L| + |K| \quad (1a)$$

$$L_n \cap L_{n'} = \phi \quad \forall n, n' \in N, n \neq n' \quad (1b)$$

$$K_l \cap K_{l'} = \phi \quad \forall l, l' \in L, l \neq l' \quad (1c)$$

Table 2. Clustering variables

A	Affinity matrix The similarity of one indicates that two objects are grouped into the same cluster and a similarity of zero otherwise
D	Degree Matrix It is a diagonal matrix where each diagonal element is the sum of all its row elements of A
L	Laplacian Matrix It is the difference between Degree Matrix and Adjacency Matrix
$d(e_i, e_k)$	Euclidean distance function $e_i, e_k \in E$
σ	Scaling parameter between two similar data points

There is an uplink and downlink communication between the clustered EVs and the charging stations. The EVs communicates to the charging stations through the RSU-CR which is further connected to the RSU-CS.

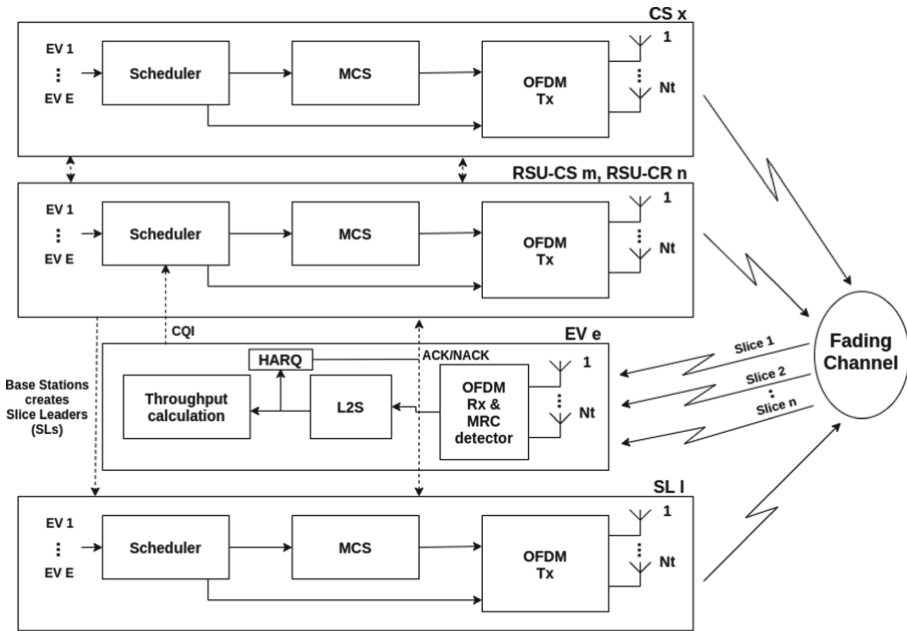


Fig. 1. Modeled wireless link for sliced EV network

The communication channel from RSU-CR n to SL l over subcarrier $r \in \mathcal{R}^{\mathcal{R}^{SU-CR}}$ is denoted as $h_{n,l}^r$. Similarly the channel from slice leader l to the clustered EV k over subcarrier $r \in \mathcal{R}^{\mathcal{S}^{\mathcal{L}}}$ is denoted as $h_{l,k}^r$. The received signal by RSU-CR at time t is given by Eq. (2) and the received signal by slice leader at time t is given by Eq. (3)

$$y_{n,l}^r(t) = h_{n,l}^r(t) \cdot x_{n,l}^r(t) + \sum_{n' \in \mathcal{N} \setminus n} h_{n',l}^r(t) x_{n',l}^r + Z_{n,l}^r(t) \quad (2)$$

$$y_{l,k}^r(t) = h_{l,k}^r(t) \cdot x_{l,k}^r(t) + \sum_{l' \in \mathcal{L} \setminus l} h_{l',k}^r(t) x_{l',k}^r + Z_{l,k}^r(t) \quad (3)$$

where $x_{n,l}^r$ is the transmitted signal from RSU-CR n to SL l , $h_{n',l}^r$ is the channel vector from interfering RSU to the l -th vehicle, and $z_{n,l}^r$ is the additive noise. Similarly, $x_{l,k}^r$ is the transmitted signal from access point l to clustered EV k and the interfering channel vector is $h_{l',k}^r$.

The uplink communication is represented as: The communication channel from charging station x to RSU-CS m over subcarrier $r \in \mathcal{R}^{CS}$ is denoted as $h_{x,m}^r$. Similarly the channel from RSU-CS m to the RSU-CR n over subcarrier $r \in \mathcal{R}^{\mathcal{R}^{SU-CS}}$ is denoted as $h_{m,n}^r$. The received signal by charging station at time t is given by Eq. (4) and the received signal by RSU-CS at time t is given by Eq. (5)

$$y_{x,m}^r(t) = h_{x,m}^r(t) \cdot x_{x,m}^r(t) + \sum_{x' \in \mathcal{X} \setminus x} h_{x',m}^r(t) x_{x',m}^r + Z_{x,m}^r(t) \quad (4)$$

$$y_{m,n}^r(t) = h_{m,n}^r(t) \cdot x_{m,n}^r(t) + \sum_{m' \in \mathcal{M} \setminus m} h_{m',n}^r(t) x_{m',n}^r + Z_{m,n}^r(t) \quad (5)$$

where $x_{x,m}^r$ is the transmitted signal from CS x to RSU-CS m , $h_{x',m}^r$ is the channel vector from interfering x th vehicle to the RSU, and $z_{x,m}^r$ is the additive noise. Similarly, where $x_{m,n}^r$ is the transmitted signal from RSU-CS m to RSU-CR n , the interfering channel vector is $h_{m',n}^r$ and $z_{m,n}^r$ is the additive noise.

The EVs continuously calculates the channel quality information(CQI) at every time interval and provides it to the RSUs. The CSs are connected to the RSU-CS as shown in Fig. 1. The scheduler in the RSUs transmit the packet of data to the EVs. When there is a degradation in the CQI, the data packet is set to the modulation code scheme(MCS) which modulates the data packet so that it can be transferred to the EV with its current CQI level. Similarly, the EV repeatedly sends its CQI to the SL which provides safety messages to the EV. The EV communicates with CS in the uplink direction and provides it with its charge state information.

4 Algorithm for Optimal K-Slice Leader Selection via Spectral Clustering

Network slicing provides efficient and configurable services for various use case demands of electric vehicles. It splits the resources into logical or virtual networks which are called as 'slices' to address use cases with distinct characteristics and service level agreement (SLA) requirements. It also has the capacity to provide different services within the same infrastructure. The proposed approach of network slicing involves partitioning the physical network into three slices- infotainment slice, safety message slice and charge state information slice.

Infotainment and safety message slices are assigned for the downlink communication, while charge state information slice is used for the uplink communication.

Infotainment slice provides the streaming videos to the vehicles through RSU, safety message slice passes the required safety messages to the respective vehicles through slice leaders and the charge state information slice has to satisfy the charge requests made by the EVs to the RSUs. If there are large number of EVs in the network and every EV sends resource requests to the RSUs, it might result in collision in the network. Hence the electric vehicles are partitioned to form a number of clusters based on various scaling parameters, so that only the slice leader of the clustered EVs communicates with the RSUs which reduces collision. To handle multi scale data, a suitable scaling method must be used. We have considered local scaling as it performs a better clustering if the data is multi scaled and when the clusters are unordered.

Affinity matrix A is a distance based similarity matrix which is defined as:

$$A_{ij} = \exp\left(\frac{-d^2(e_i, e_j)}{\sigma_i \sigma_j}\right) \quad (6)$$

$d(e_i, e_j)$ is a distance function. Euclidean distance measure is used to calculate the distance between the vectors e_i and e_j . σ is the scaling parameter usually used when two points are considered similar. In local scaling, rather than selecting a single scaling parameter σ , local scaling parameter σ_i for each data point e_i is calculated [13]. The advantage of using local scaling is, as a specific scaling parameter is found for every data point, it leads to self tuning between the distance between every two points.

The scaling parameter of each point is calculated by considering the local statistics of the neighborhood point of e_i , which can be defined as follows:

$$\sigma_i = d(e_i, e_k) \quad (7)$$

where e_k is the k_{th} neighbor of point e_i .

Algorithm 1: Local Scaled Clustering of EVs

Input: Coordinates of EVs

Result: Clustered EVs ($K = \{1, 2, 3, \dots, k\}$)

Given a set of points $E = E_1, E_2, \dots, E_n$ in P that we want to cluster:

1. Compute the local scale σ_i for each point $E_i \in E$ using equation 7.
 2. From the locally scaled affinity matrix $A \in P^{n \times n}$ where A_{ij} is defined according to equation 6 for $i \neq j$ and $A_{ii} = 0$
 3. For finding the optimal number of cluster:
 - (a) Define D to be a diagonal matrix with $D_{ii} = \sum_{j=1}^n A_{ij}$ and construct the normalized affinity matrix $L = D^{-1/2} \cdot A \cdot D^{-1/2}$
 - (b) Find the eigen values and their associated eigen vectors
 - (c) Identify the maximum gap which corresponds to the number of clusters by eigengap heuristics Index corresponding to the maximum gap between eigen values will be the optimal number of cluster k .
 4. Cluster the points $(E_i)_{i=1, 2, 3, \dots, n}$ with the k -means algorithm into cluster $C_1, C_2, C_3, \dots, C_k$.
 5. Find the k clusters and corresponding clustering labels.
-

Before clustering, the optimal number of clusters must be determined which is usually assumed manually. We have introduced an approach of determining the

number of clusters which is based on the organization of the eigen vectors. A diagonal matrix D is determined from which the normalized laplacian matrix L is constructed. Using the eigen values and corresponding eigen vectors, the number of clusters can be estimated by finding the data points which have the maximum gap between its eigen values. The index corresponding to the maximum gap is assumed to be the number of clusters k . After finding the value of k , the points (E_i), where $i = 1, 2, \dots, n$ can be clustered by k -means clustering algorithm. K -means clustering is a better approach for clustering as it can even be used when the data is noisy. The output of clustering is the set of clustered vehicles $K : 1, 2, \dots, k$ in Algorithm 1.

Algorithm 2: Slicing Algorithm

Result: $L = \{l_1, l_2, l_3, \dots, l_k\}$

Input: Clustered EVs ($K = \{1, 2, 3, \dots, k\}$)

Output: Set of Slice Leaders $L = \{l_1, l_2, l_3, \dots, l_k\}$

1. For each cluster $i = 1$ to k do
 2. Let y_i represent the center of cluster i
 3. Find the distance between y_i and vehicles $e \in E/\#k$
 4. Find $l_i = \text{argmin}_{\forall e \in E/\#k} \{y_i - d_e\}$, where d_e is the location of vehicle e
 5. end for
-

Slice leaders are determined to handle the set of clustered vehicles. They represent the entire cluster and makes a direct communication with the RSUs for requesting resources. The cluster center is determined by the RSU using geographical information such as the distance of all the vehicles $e \in E$. The clustered vehicles $i : 1, 2, \dots, k$ and its centers (y_i) determined by the spectral clustering algorithm is taken as the input. The Euclidean distance between the cluster center y_i and all the vehicles (e) within the cluster is calculated. The vehicle which is closest to the cluster center is considered as the slice leader (L). This is done for all the clusters, and a set of slice leaders $L = \{l_1, l_2, l_3, \dots, l_k\}$ is the output of the Algorithm 2.

The offloading mechanism helps to convert the slice leader that is determined by Algorithm 2 of every cluster that is determined by Algorithm 1 into a virtual RSU by SDN and NFV as they have the highest quality V2I and V2V links. Clustered vehicles at the cell edge coverage provided by the fixed radius RSU suffer from degradation of performance. As a result these vehicles have low SINR. RSU assigns the slice leader which serves as an offloaded vehicle to become the access point for the low SINR vehicle. The resources available at the RSU are directly shared to the clustered vehicles through the slice leader in the form of slice blocks. Slice Block allocation algorithm is required to allocate resources to the EVs, prioritised based on amount of resource required for the critical messages such as safety messages and charged state information messages.

5 Efficient Slice Block Allocation Algorithm

We have considered slice block allocation to be in discrete time slots between the slice leaders \mathcal{L} and the clustered vehicles \mathcal{K} . $\mathcal{R}^{S\mathcal{L}}$ is the resource to be allocated to \mathcal{K} at time interval $t = 1$.

Algorithm 3: Algorithm for Slice Block allocation

Input: Resource available [R], EV= EVs resource requested [I, SM] where SM = S+C
Output: Set of allocated EVs with resources EV_{alloc}
initialization: Left_over_resources (L) = Resource available (R)
 Create a performance profile (P) $\in EVs$ based on descending SM values Initialize
 $EV_{alloc} = 0 \forall EV \in P$
for every EV in preference profile (P):
if $SM[EV_m] > L$: **then**
 Continue;
 if $sum(P[EV_m]) \leq L$: **then**
 $EV_{alloc_m} = sum(P[EV_m]);$
 $L = L - sum(P[EV_m]);$
 else
 $EV_{alloc_m} = SM[EV_m];$
 $L = L - SM[EV_m];$
 end
end
end for
Return: EV_{alloc}

In Algorithm 3, we consider two slices between the clustered EV. The first slice is the infotainment slice [I], and the second slice is Safety Message slice (SM) which contains both the safety message (S) and the charging message (C).

Algorithm 4: Slice block allocation for uplink and downlink

Input: Resource available for downlink (RD), Resource available for uplink (RU), EVs Resources requested [I, S, C]
Output: Set of allocated EVs at downlink and uplink with resources $[E_{allocD}, EV_{allocU}]$
initialization: left over resource at downlink $[LD] = RD$
 Left over resource at uplink $[LU] = RU$
 Create a preference profile [PD] of EVs for downlink based on descending S values
 Create another preference profile [PD] of EVs for downlink based on descending C values
 Initialize $EV_{alloc} = 0$ and $EV_{allocU} = 0$ for every EV in PD and PU respectively
for every EV in PD
if $S[EV_m] > LD$: **then**
 Continue;
 if $sum(PD[EV_m]) \leq LD$ **then**
 $EV_{allocD} = sum(PD[EV_m]);$
 $LD = LD - sum(PD[EV_m]);$
 else
 $EV_{allocD} = S[EV_m];$
 $LD = LD - S[EV_m];$
 end
end
end for
for every EV in PU
if $C[EV_m] > LU$ **then**
 Continue;
 if $PU[EV_m] < LU$ **then**
 $EV_{allocU} = PU[EV_m];$
 $LU = LU - PU[EV_m];$
 else
 $EV_{allocU} = C[EV_m];$
 $LU = LU - C[EV_m];$
 end
end
end for
Return: EV_{allocD}, EV_{allocU}

Given the resource R which has to be distributed among the clustered EVs \mathcal{K} in a non-preemptive manner such that the EVs with higher SM are allotted resource first. Left over resource (L) is initialized to total number of available resources $[R]$. A preference profile $[P]$ of EVs is created based on the value of $[SM]$ such that the EV requesting for larger $[SM]$ is given a higher preference. For every EV in the preference profile, resource is allocated if the left over resource $[L]$ can satisfy the requested resource of the EV $[I+SM]$, else resource only for SM is allocated to EV if it is less than L . The value of L is updated after allocation. The allocated EVs with their resources are returned as output. Since the charging state information slice $[C]$ is for uplink communication while infotainment $[I]$ and safety message $[S]$ slices are for downlink communication, we separate the resource allocation for uplink and downlink scenarios.

In Algorithm 4, EV resources requests $(I, S$ and $C)$ along with the resources available for downlink and uplink are taken as input. Left over resources in the downlink and uplink direction $(LD$ and $LU)$ are initialized to their respective available resources $(RD$ and $RU)$. Two preference profiles are created. A downlink $[I+S]$ preference profile $[PD]$ and an uplink $[C]$ preference profile $[PU]$ such that EVs with greater $[S]$ values and greater $[C]$ values are given higher preference respectively. Similar to Algorithm 3, resource allocation is made to the EVs based on the preference profiles of uplink and downlink direction separately. For every EV in preference profile of downlink $[PD]$, resource is allocated if the left over resource $[LD]$ can satisfy the requested resource of the EV $[I+S]$, else resource only for S is allocated for EV if it is less than LD . Similarly, for every EV in preference profile of uplink $[PU]$, resource is allocated if LU can satisfy the requested resource of EV $[C]$.

6 Experimental Study

6.1 Slice Leader Selection in Each Cluster

The following graphs demonstrate Local Scaled Clustering algorithm which determines the optimal number of clusters and performs clustering shown in Fig. 2. In 2(a), we have considered a set of EVs generated by the sample generator. The slice leader of each cluster is determined by the slicing algorithm where each cluster has one slice leader. We have considered information of 300 vehicles in 2(b) where 32 clusters and their respective slice leaders are determined. In 2(c), a shared vehicle dataset [2] is considered, which consists of co-ordinates of 5000 vehicles. Clustering is performed and 244 clusters are determined. The slice leaders obtained are very close to each other due to less inter vehicular distance between the EVs.

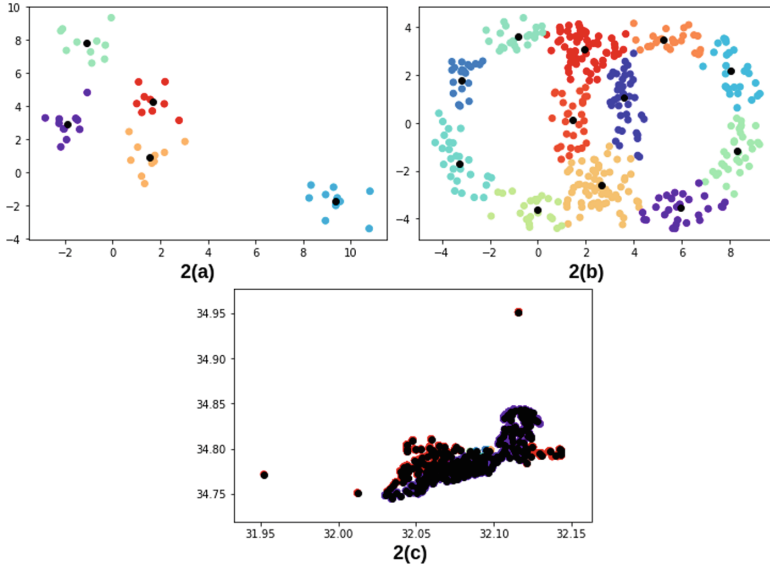


Fig. 2. Slice leaders within cluster on various datasets

In the Fig. 2, coordinates of EVs are represented by different colored data points where each color represents a single cluster and the slice leaders are represented by black colored data points.

6.2 Slice Block Allocation

We have made a comparison of our proposed algorithm with First-Come-First-Serve (FCFS) algorithm. In our proposed algorithm, we have given a higher preference to the critical messages such as safety messages and charge state information messages over infotainment messages. Whereas in FCFS, the resources are allocated in the order of the request made and it does not give any preference to the critical messages.

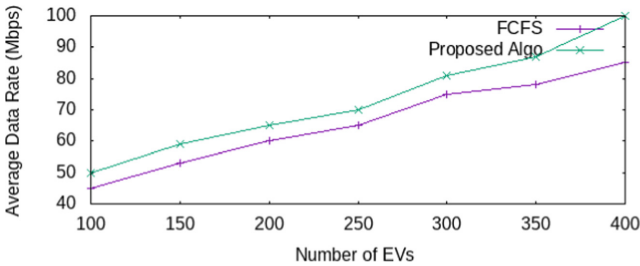


Fig. 3. Average data rate comparison of proposed and existing algorithm

In Fig. 3, we have shown the performance of resource allocation algorithm with FCFS by comparing the number of EVs requesting for resources over average data rate (in Mbps). The average data rate refers to the number of resources allocated to every cluster of EV. It varies as the number of EVs requesting for the resources vary. The amount of resources allocated by the proposed algorithm is greater than that of FCFS considering same number of EVs in both cases.

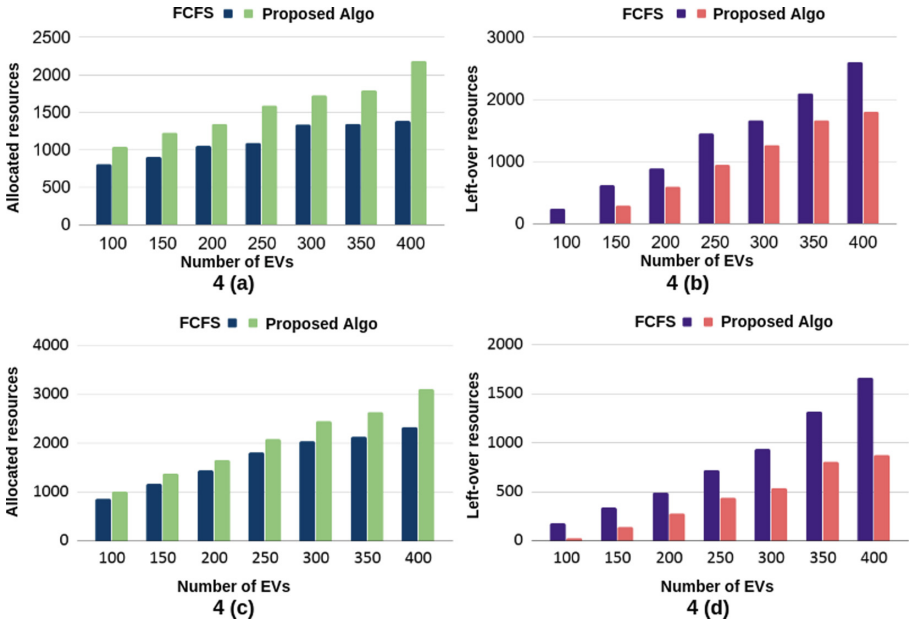


Fig. 4. Average data rate comparison of proposed and existing algorithm

We have considered two scenarios to show the performance of proposed algorithms with FCFS Fig. 4. In 4(a) the number of EVs requesting for resources is compared over the number of resources allocated. The allocated resources refers to the critical messages (safety message slice and charged state information slice). Our proposed resource allocation algorithm allocates more number of critical messages to the EVs when compared to FCFS. If we consider one scenario from the figure, where there are 250 EVs requesting for resources, proposed algorithm allocates 1600 Mb of critical messages to the EVs whereas FCFS allocates 1100 Mb of critical messages. In 4(b), the number of EVs is compared over the left over resources. Left over resources refers to the difference between the total number of resources considered before allocation and the total number of allocated resources. The lesser is the value of left over resource, the greater is the efficiency. Our proposed resource allocation algorithm is giving a better result than FCFS because there are more number of left over resources in FCFS compared

to the proposed algorithm. Similarly, we have considered the resource allocation algorithm for uplink and downlink. In 4(c), the number of EVs requesting for resources is compared over the number of resources allocated. The allocated resources refers to the sum of number of safety messages allocated in downlink direction and the number of charge state information messages allocated in uplink direction. Our proposed algorithm allocates more number of critical messages to the EVs when compared to FCFS. In 4(d), the number of EVs is compared over the left over resources. Left over resources refers to the sum of resources left after resource allocation in uplink and downlink respectively. The figure shows that there are more number of resources left after allocation in FCFS over the proposed algorithm.

7 Conclusion

In this work, we have studied the performance of the proposed network slicing in EV environment. The scenario consists of vehicles which are connected to the RSUs which are further connected to the charging stations. We have considered slices where charged state information and safety message rely on URLLC and the higher data rate is used by the infotainment slice. The proposed algorithm of creating the slice leaders and clustered vehicles provide huge throughput and reliability by maximising the utilization of V2I and V2V links. The resource allocation algorithm helps to maximize the allocation of resources to the vehicles based on the priority defined by higher requirement of critical messages such as safety messages. In the future, the time-series deep learning models could be added for network slicing and the proposed algorithm can be extended to make predictions of the resources required by the EVs.

References

1. Addad, R.A., Taleb, T., Flinck, H., Bagaa, M., Dutra, D.: Network slice mobility in next generation mobile systems: challenges and potential solutions. *IEEE Netw.* **34**(1), 84–93 (2020)
2. Benram, G.: Shared cars locations - location history of shared cars (2019). <https://www.kaggle.com/gidutz/autotel-shared-car-locations>
3. Campolo, C., Molinaro, A., Iera, A., Menichella, F.: 5G network slicing for vehicle-to-everything services. *IEEE Wirel. Commun.* **24**(6), 38–45 (2017)
4. Jiang, M., Condoluci, M., Mahmoodi, T.: Network slicing in 5G: an auction-based model. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2017)
5. Khan, H., Luoto, P., Samarakoon, S., Bennis, M., Latva-Aho, M.: Network slicing for vehicular communication. *Trans. Emerg. Telecommun. Technol.* **32**, e3652 (2019)
6. Kurtz, F., Bektas, C., Dorsch, N., Wietfeld, C.: Network slicing for critical communications in shared 5G infrastructures-an empirical evaluation. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 393–399. IEEE (2018)

7. Leconte, M., Paschos, G.S., Mertikopoulos, P., Kozat, U.C.: A resource allocation framework for network slicing. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 2177–2185. IEEE (2018)
8. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is NP-hard. *Theoret. Comput. Sci.* **442**, 13–21 (2012)
9. Mathur, R.P., Pratap, A., Misra, R.: Distributed algorithm for resource allocation in uplink 5G networks. In: Proceedings of the 7th ACM International Workshop on Mobility, Interference, and MiddleWare Management in HetNets, pp. 1–6 (2017)
10. Nakao, A., et al.: End-to-end network slicing for 5G mobile networks. *J. Inf. Process.* **25**, 153–163 (2017)
11. Rost, P., et al.: Network slicing to enable scalability and flexibility in 5G mobile networks. *IEEE Commun. Mag.* **55**(5), 72–79 (2017)
12. Vassilaras, S., et al.: The algorithmic aspects of network slicing. *IEEE Commun. Mag.* **55**(8), 112–119 (2017)
13. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems, pp. 1601–1608 (2005)
14. Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A.H., Leung, V.C.: Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. *IEEE Commun. Mag.* **55**(8), 138–145 (2017)
15. Zheng, J., de Veciana, G., Banchs, A.: Constrained network slicing games: achieving service guarantees and network efficiency. arXiv preprint [arXiv:2001.01402](https://arxiv.org/abs/2001.01402) (2020)



A Peak Bulk Deal Tracing Based Comparative Analytical Study for Optimize Investment Strategy

Amit Suthar, Hiral Patel^(✉), and Satyen M. Parikh^(✉)

Faculty of Computer Applications, Ganpat University, Mehsana, India
{Abs01,hrp02,satyen.parikh}@ganpatuniversity.ac.in

Abstract. Investing in the financial market is not a new or exceptional case in real. Recently, many people are interested to invest in financial market to maximize the ROI. To invest in financial market is not always gives a profit. So for investing in market need more analytical research or study or expert knowledge to identify the best investment which maximize the ROI. Every investor doesn't has expertise to perform analytical study so these type of modelling helpful them to increase the profit. This paper focus on the investment strategy base available methods and perform comparative analytical study to provide the optimize investment strategy for peak bulk deal selection. Predominant bulk deal picking is basic among dynamic value subsidize directors. The explanation bulk deal picking aptitude isn't identified by numerous exhibition contemplates is over-enhancement, which overpowers the prevalent execution of top possessions. Utilizing self-announced model for arranging the dynamic value universe liberates supervisors to seek after a barely characterized technique and aides in distinguishing effective chiefs inside every bulk deal.

Keywords: Bulk deal · Herding · Investment strategy · Predictive modelling · Return of investment

1 Introduction

Institutional investment from insurance companies, pension funds, mutual funds, hedge funds and private equity firms makes up the majority of the overall trading volume. By pooling the resources of others, they serve as intermediaries between investors and companies, thereby supplying finance markets with the necessary source of capital. Using professional knowledge of investment, they exchange positively to optimize returns from the investment. Historically, price manipulation and the unscrupulous position of brokers on capital markets has been a subject of great concern to market participants as well as governments, and have a significant impact on market effectiveness. Amidst a large number of small retail investors, the presence of few institutional investors is like a tiny herd of elephants marching through a big herd of rabbits. Retail investors with limited resources are unable to exert much impact on the chosen stocks and, at the same time, are most influenced by market swings usually created by institutional

investors. Many investors pursue strategies of constantly picking up and trading stocks instead of merely searching for and holding the portfolio on the marketplace. Shopping for and encouraging decisions can also spread stock prices as investors transact actively. Therefore, understanding the actions of stock expenses requires knowledge in successful investor's investment strategies. In principle, extracting data effectively from multiple strategies of technical analysis would yield higher predictive results than one strategy of technical analysis. This paper styles a bulk deal formula which supports machine learning and technical indicators in order to explore the data posed in technical analysis. The model uses machine learning algorithms to mine popular technical diffusion indicators. The model uses machine learning algorithms to mine a set of specific technical indicators to predict the direction of the stock value once several days (rise or fall) so constructing a portfolio supported the direction of forecasting.

2 Literature Review

The technique and related forecasts rely upon the time-size of the speculation, as exemplified by the accompanying classes of quantitative systems: major investigation when a stock is exchanging under natural worth. The long haul systems spurred quant makes some quarterly memories scale. Efficient large scale based on macroeconomics investigation or market occasions and patterns to recognize open doors for speculation. Efficient full scale systems are model-based and executed by programming with constrained human contribution. The above efficient full scale has a month to month timescale. Combination or relative worth exchanges and other factual exchange (StarArb) procedures allude to exchanging comparable resources that are relied upon to unite in esteem. These techniques are instances of StatArb systems which have time scales running from minutes to months. High recurrence exchanging (HFT) as known as purchasing/selling a lot of offers in a brief timeframe. The time size of HFT in milliseconds and holding time of the exchanged offers is generally less than one second [6].

As indicated by Keim and Madhavan (1996) the value effects of square exchanges are bigger for loads of little measured organizations (showcase capitalization). Ruder et al. (1988) likewise find that it is simpler to control penny (little) stocks which are meagerly exchanged, on account of low perceivability in the business sectors and consequently the benefits to the merchants who enjoy controlling the costs of these stocks can be enormous. We try to discover if size furthermore, amount of the portions of the firm exchanged (liquidity) have any effect on anomalous comes back from enormous exchanges. Aktas and Kryzanowski (2014) likewise clarify the significance of enormous exchanges and express that educated exchanging is more for bigger exchanges [1].

An investigation in Korean securities exchange affirms this conviction where they discovered individual speculators crowd more than institutional financial specialists (Kim and Wei 2002). In any case, there is another way of thinking which accepts that it is the institutional speculators who group more than singular speculators as a result of their craving to keep up their notoriety in the market. Institutional speculators who enter the market late are seen as less intense to take solid choices on the dread of losing the notoriety if the choices turn out badly (Truman 1994; Zwiebel 1995; Prendergast and Stole 1996; Teh and De Bondt 1997). The discoveries of Choi and Skiba (2015) bolstered

the above proclamation that institutional financial specialists crowd more than singular financial specialists in the market [2].

This investigation plans to display a handy philosophy to structure an ideal portfolio methodology of the worldwide securities exchange with monetary system markers. Despite the fact that there have been numerous examinations on the significance of system investigation, particularly in high unpredictability states (Billio et al. 2012; Diebold and Yilmaz 2016), there have been just a couple of endeavors that develop venture methodologies through these system pointers (Peralta and Zareei 2016). Late investigations of system pointers for speculation (Chuluun 2017; Barunik et al. 2017) give the proof of the system impacts of speculation techniques. In any case, these papers are additionally not extended to pragmatic speculation methodology furthermore, impacts in portfolios. Additionally, there have been no endeavors to use them in the most significant and troublesome worldwide budgetary market viewpoint and portfolio system [3–5].

3 Bulk Deal Trading Methodology

Bulk deal means the performing the transaction on mass arranged stocks. A mass arrangement is where complete amount of offers purchased or sold is over 0.5% of the quantity of portions of a recorded organization. Mass arrangements occur during typical exchanging window gave by the broker. The intermediary who deals with the mass arrangement exchange needs to give the subtleties of the exchange to the stock trades at whatever point they occur. Not at all like square arrangements, are mass arrangement orders noticeable to everybody. Usually it’s owned by profound stashed financial specialists like reserve houses, remote institutional speculators, banks, protection firms and HNIs given the high sum required to go into such exchanges and the level of offers included (Fig. 1).

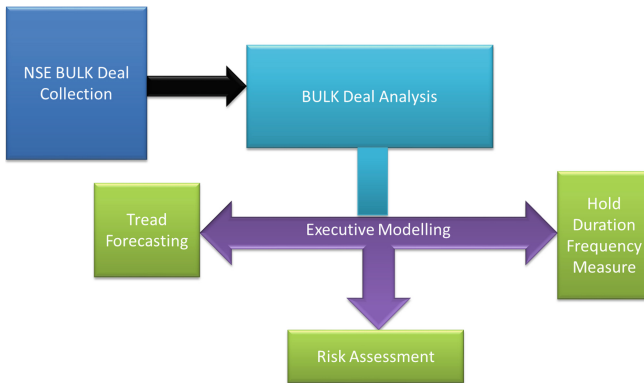


Fig. 1. Bulk deal trading model

(A) ***NSE Bulk Deal Collection***

Volume Shockers incorporates the stocks that are seeing a gigantic flood in volume exchanged today when contrasted with their 3/7/15-day normal volume showing

abrupt increment in financial specialist enthusiasm for a stock. This typically happen when there is some news stream related with a stock. The work collects the NSE bulk deals and performs the analytics survey.

(B) ***Bulk Deal Analytics***

Dissect NSE mass arrangements, mass arrangements by fii, mass arrangements by common assets, NSE square arrangements, square Deal of NSE stocks, square arrangement by fii and square arrangement by shared assets with the assistance of equity friend apparatus to discover which FII, FPI, DII or Mutual Fund is purchasing and selling stocks. Presently you can channel mass arrangement and square arrangement information on three essential criteria's for example Date, Company and Mutual Fund or FII/FPI. By separating Bulk Deals based on organizations, you can likewise make sense of administrator driven counters and spare yourself from speculation calamities. If you don't mind pick Deal Type and any of the three channel criteria from the dropdown beneath and click on Get Bulk Deal/Block Deal Data catch to get the date shrewd mass arrangement information, organization savvy mass arrangement information or shared store astute mass arrangement information. You can utilize the Reset Data catch to reset the information and cause the vanished fields to return.

(C) ***Hold Duration Frequency Measur***

The key points for investors to keep in mind for investing in bulk deals. Speculators should screen NSE mass arrangements and NSE square arrangements every day and keep a watch on following information focuses Search for stocks with solid basics which are showing up all the time in the mass arrangements and square arrangements. On the off chance that enormous establishments are reliably purchasing stocks with solid basics in huge amount at that point it's a decent pointer of future value appreciation. Likewise, on the off chance that the enormous establishments are selling stocks on a reliable premise, at that point one can be genuinely certain about future value devaluation. Search for stocks with frail basics reliably showing up in the mass and square arrangement list. They may be administrator driven and speculators should twofold check their choice before putting resources into them. Concentrate the profile of money related foundations engaged with the nse mass arrangements and NSE square arrangements. Colossal purchase or sell figures by huge establishments are regularly upheld by strong research and may show the future value development reasonably precisely. The mass arrangement action can likewise go about as a reasonable marker of winning business sector notion. In the event that the general market notion is bearish, mass arrangements on the sell side will be moderately enormous and if the market feeling is bullish, purchase mass arrangements will dwarf the sell mass arrangements. NSE also provides the top 20 top bulk deals for making easy to investment in bulk deals for investors. So for experimental purpose the model utilize the nifty 20 top most bulk deals of Mutual fund.

(D) ***Risk Assessment***

As per the experimental data, model focuses on Mutual fund based bulk deals investment strategy. So the risk in investing MF as well as the market trend and sentiments plays vital role in this model forecasting. MF work on different subjects which opens them to

various sort of dangers. Despite the fact that they are expertly overseen however component of hazard despite everything remains. These dangers can be credited to financial execution, broadening, part development and individual organization execution. Preceding taking speculation choice financial specialists should crosscheck supports execution as for different hazard measures. In this model, it tempt to comprehend the parameters utilizing which subsidizes execution is estimated and chance investigation is finished. Financial specialists must perform near examination of these parameters before settling on a speculation choice.

The following measures will carried out for risk assessment. Standard Deviation, Correlation analysis, Mean, Sharpe Ratio, Beta risk map, Treynor Ratio, Sortino, Fama Model and Jensen Model will calculated.

4 Conclusion

After analyzing the literatures in this era, the Data set taken from Nifty top 20 deals of Mutual fund. The data properly tested via statistical analysis for goodness fitness. The rules of bulk deal analytics are trained in ML algorithm to perform more accurate forecasting. The predictive modeling is applied on the data to train the model and forecast the value. After forecasting the risk assessment is also apply for defining best optimized investment strategy.

References

1. Aktas, O.U., Kryzanowski, L.: Market impacts of trades for stocks listed on the Borsa Istanbul. *Emerg. Mark. Rev.* **20**, 152–175 (2014)
2. Choi, N., Skiba, H.: Institutional herding in international markets. *J. Bank. Financ.* **55**(1), 246–259 (2015)
3. Billio, M., Getmansky, M., Lo, A.W., Pelizzon, L.: Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *J. Financ. Econ.* **104**, 535–559 (2012)
4. Peralta, G., Zareei, A.: A network approach to portfolio selection. *J. Empirical Financ.* **38**, 157–180 (2016)
5. Chuluun, T.: Global portfolio investment network and stock market comovement. *Glob. Financ. J.* **33**, 51–68 (2017)
6. Ta, V.-D., Liu, C.-M., Addis, D.: Prediction and portfolio optimization in quantitative trading using machine learning techniques. In: *Proceedings of the Ninth International Symposium on Information and Communication Technology - SoICT (2018)*. <https://doi.org/10.1145/3287921.3287963>
7. <https://www.equityfriend.com/investor-tools/nse-bulk-deal-data-nse-block-deal-data.html>
8. Patel, H., Parikh, S.: Automated news based ulip fund switching model. In: *Presented and Published with International Conference GCEMP 2012, Awarded as Best Technical Paper (2012)*
9. Patel, H., Parikh, S.: A comparative study on financial stock market prediction models. Published by: *Int. J. Eng. Sci. (IJES)*, vol. **1**(2), 188–191 (2012). ISSN: 2319 – 1813 ISBN: 2319 – 1805, Indexed in ANED (American National Engineering Database) Impact Factor 7.2
10. Patel, H., Parikh, S.: A proposed prediction model for forecasting the financial market value according different factors. *Int. J. Comput. Technol. Appl. (IJCTA)* ISSN 2229–6093, vol. **5**(1) (2014). Jan-Feb 2014. Impact Factor – 2.015 IC Value 5.17

11. Patel, H., Parikh, S.: A technical and fundamental parameters analysis for financial market prediction using semantic analysis (2014)
12. Patel, H., Parikh, S.: Dynamic IS based asset allocation on crude trend analysis – exploring a hedging concept. Presented and Publishing in GCEMP-15, GFJMR (2015). ISSN 2229 -4651
13. Patel, H., Parikh, S.: Comparative analysis of different statistical and neural network based forecasting tools for prediction of stock data. Presented At ICTCS – 2016, Publication (ACM Conference) (2016). <http://dl.acm.org/citation.cfm?id=2905055.2905186>
14. Patel, H.R., Parikh, S.: Comparative analytical study for news text classification techniques applied for stock market price extrapolation. In: Unal, A., Nayak, M., Mishra, D.K., Singh, D., Joshi, A. (eds.) SmartCom 2016. CCIS, vol. 628, pp. 239–243. Springer, Singapore (2016). https://doi.org/10.1007/978-981-10-3433-6_29
15. Patel, H., Parikh, S.: Prediction Model for Stock Market using News based different Classification, Regression and Statistical Techniques (PMSMN) ICTBIG 2016 IEEE Publication, 18, 19 October 2016. Indore
16. Patel, H., Parikh, S.: Experimental study on stock market to analyse the impact of the latest demonetization in India. In: CiiT International Journal and Publication (2017)
17. Patel, H., Parikh, S.: Prediction model based on NLP and NN for financial data outcome revelation 3rd International Young Scientist Congress (IYSC-2017), Awarded as Young Scientist Award, Published in Research Journal of Computer and Information Technology Sciences (2017). E-ISSN: 2320 – 6527



Path Planning Algorithms for Different Scenarios

Saurabh Singh^(✉) and Namita Tiwari

Maulana Azad National Institute of Technology, Bhopal, India

Abstract. Searching of the target(s) in unknown Scenarios is a very complex task in the field of Robotics. This paper is an introduction to the works that are seminal in the field of Swarm Robotics. Swarm Robotics is the application of the principles used in swarm Intelligence. Different types of algorithms are used to decide the path in different types of situations or environments. Due to the robustness, scalable, fault-tolerant, and many other properties of the swarm of robots, the SRS is used in many search and tracking applications. This is the review that focuses on the different problems in SRS. Firstly an algorithm that deals with the local minima problem is explained, further, there are two more algorithms that are used to search the target and guide the robots to the target are explained. These algorithms perform differently for different scenarios. Further, seven different types of algorithms are compared with each other based on some parameters. Most of these algorithms perform well in some situations but may fail to perform well in some other situations.

Keywords: Particle swarm optimization · Ant colony optimization · Swarm intelligence

1 Introduction

The Target Search or tracking of the target is a very complex problem mostly when the surrounding environment of the target or search area is unknown or is too complex i.e. containing many different kinds of obstacles. In recent times a variety of efficient applications have been made in this field including search and tracking in the disaster area, exploration of Natural resources, monitoring of Natural resources and environment, air traffic control, etc. [1].

The common thing in all of the target search and tracking applications is path planning which is to be followed by the multi-robots to reach the desired target position. This problem is also previously partially solved by the use of multi-robot systems, by the use of the mobile robots which can dynamically adapt to the changing position of the target in some complex environments [2].

In this paper, a discussion of various search and tracking algorithms is given which can be used in different environments. In recent years, there are many reviews on the path planning in swarm intelligence but most of them are the broad reviews, instead of going into the details of the previous papers.

2 Overview of Swarm Robotics

The term Swarm Robotics defines itself as it includes the swarm of homogeneous robots which have the capability of working in coordination with each other. They do not include any centralized system or no head robot or system is controlling all other systems or robots. In 1989 the term ‘swarm intelligence’ was introduced by Beni and Wang in the context of cellular robotic systems. The first cellular robotic system was introduced by Fukuda which was termed as CEBOT [4, 5].

Basically, Swarm robotics is introduced because of the advantages it provides as compared to a single robot.

- **Scalable:** Scalability in swarm robotics means the size of the swarm can be increased or decreased as needed. This does not have any effect on the performance of the swarm as each robot is independent in itself [6, 7].
- **Flexible:** Here in the context of swarm robotics, flexible means that the robots are flexible with the changes in the environment [6], surroundings, and/or nature of the task given. For SRSs it is essential to be flexible so that there is no need to reprogram the robots when the problem changes.
- **Robust:** The swarm of robots can be termed as robust by the measure by which the swarm can continue to perform the assigned task even if some of the robots of the swarm fails due to some reason or if there arise some non-suitable conditions for robots [6]. This is possible because of the design of every individual robot. So some technical issues don’t affect the complete swarm that much.

3 Parameters Used in Searching and Tracking

Parameters used to compare different path planning algorithms in Swarm robotics are given below.

- No. of Targets.
- Target Mobility
- Tracker Mobility
- Environment Complexity
- Knowledge of target motion in advance
- Cooperation among the robots
- Coordination between the trackers.

4 Path Planning Algorithms for SRS

In Sect. 1, it is mentioned that target searching and tracking is a complex task in swarm intelligence. In this section, the methods used for searching and tracking the targets in different environments are discussed. Few problems related to SRS are also discussed in this section.

4.1 Local Minimum Problem solution Based on PSO

The local minimum problem in SRS means that the robot is surrounded by obstacles from both sides. Basically, there are two kinds of path planning algorithms in SRS – local path planning and global path planning [16–18]. The problem of local minimum arises in local path planning algorithms. The local path planning is preferred over global path planning because local planning works in real-time and is more accurate than global planning due to the presence of more adaptability in a dynamic environment.

The further subsections include the dynamic model of the robot and the method to set up the virtual robot in a virtual environment and then the cost function used in the method.

Dynamic Model of Mobile Robot

In this paper the dynamic position and dynamic velocity of the robot are represented as:

$$\begin{cases} p(x + 1) = v(x) + p(x) \\ v(x + 1) = v(x) + u(x) \end{cases} \tag{1}$$

Virtual Robot

Here a virtual robot is used which is just the projection of the real robot over the edge of the obstacle as shown in Fig. 1.

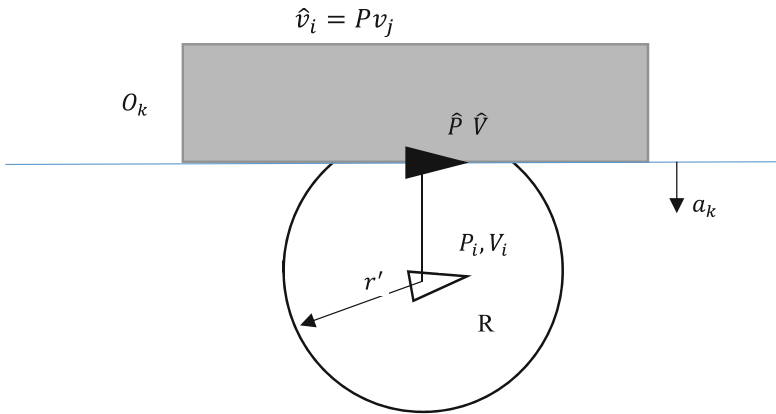


Fig. 1. Path of virtual Robot along the edge

The position and the velocity of this virtual robot is given by \hat{x}_i and \hat{v}_i . These are calculated as follows:

$$\hat{p}_i = Pp_j + (I - P)y_i \quad (2)$$

$$\hat{v}_i = Pv_j \quad (3)$$

The Particle Swarm Optimization (PSO) Algorithm

Particle Swarm Optimization is an optimization algorithm proposed by Kennedy and Eberhart [23]. In this system, each candidate works in co-operation with each other and give their solution (particle) which are compared with each other to get global optimized value. Here particle moves in problem space and seeks for its best position. This space may be multi-dimensional say d -dimensional and there can be n -number of particles. So by using these, the position and velocity of the i^{th} particle at k^{th} iteration can be given by:

$$\vec{x}_i(k) = (x_{i,1}(k), x_{i,2}(k), \dots, x_{i,d}(k)) \quad (4)$$

$$\vec{v}_i(k) = (v_{i,1}(k), v_{i,2}(k), \dots, v_{i,d}(k)) \quad (5)$$

These positions and velocities are updated at each iteration by using below:

$$\vec{x}_i(k+1) = \vec{x}_i(k) + \vec{v}_i(k+1) \quad (6)$$

$$\vec{v}_i(k+1) = \omega \vec{v}_i(k) + c_1 r_1 (\vec{Pb}_i(k) - \vec{x}_i(k)) + c_2 r_2 (\vec{Gb}_i(k) - \vec{x}_i(k)) \quad (7)$$

where $\vec{Pb}_i(k) = (Pb_{i,1}(k), Pb_{i,2}(k), \dots, Pb_{i,d}(k))$ is the local optimum location of the i^{th} particle and $\vec{Gb}_i(k) = (Gb_{i,1}(k), Gb_{i,2}(k), \dots, Gb_{i,d}(k))$ is the global optimum location. c_1, c_2 are the acceleration coefficients. They control the step size of the particles. ω is the inertia [24]. r_1 and r_2 are the random numbers between 0 and 1.

A Solution to the Local Minima Problem

The solution to the local minima problem is given by using a virtual target point method [25]. In this method, a virtual target point is set up outside the boundaries of the detected obstacle. Now the original robot tries to reach this virtual target point. If it is unable to reach this virtual target point then this target point is changed repeatedly. After reaching this virtual target, the robot moves to its original destination. For this, the PSO algorithm is used.

4.2 PSO Based path Planning in Dynamic Environments

Deciding a path for multi-service robots traveling in dynamic environments is a very complex problem.

In this paper, a PSO based optimized path planning is done for multi-service robots. The aim is to take all the robots from their source to a destination without collision between themselves or with any obstacle on the way. A simulation platform Virtual Reality Educational Pathfinder (VREP) is used to experiment.

The Approach of PSO

The particle swarm optimization is explained in (4.1.4). The approach used here is shown in the flowchart below (Fig. 2).

In this study, PSO steps followed are given as:

1. The current position and target of all the multi-service robots are determined.
2. The planning is done to decide the path from the current position of the robots to the destination position of the robots.

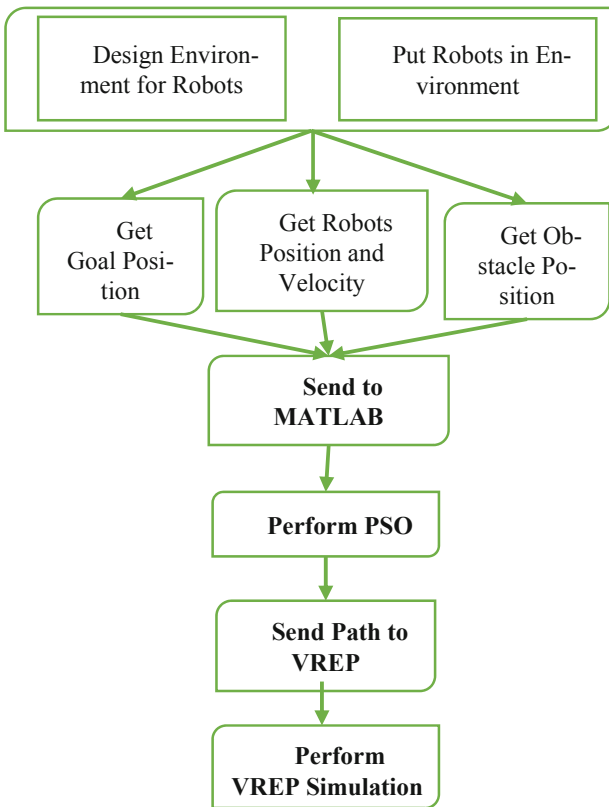


Fig. 2. Flow of the proposed algorithm

3. When met with any obstacle, each robot re-plan the path in the direction of the destination.
4. If any robot does not encounter any obstacle on the way, each robot follows the same path plan as decided earlier.

This fitness function is given as follows:

$$f(x) = \min \sum_i^n \left[(x_i - x_{goal})^2 + (y_i - y_{goal})^2 \right] \tag{8}$$

where (x_i, y_i) is the current position of the i^{th} robot and (x_{goal}, y_{goal}) is the co-ordinations of the target point. Here in this algorithm, the population size is decided by the user and can be changed accordingly.

The simulation has been done by using 4 robots. The attributes and their values of the simulation environment are given in Table 1 shown (Table 2):

4.3 A Path Planning of Multi-destination for Mobile Robots

There are many algorithms when it comes to deciding the path from one point to other or in complete coverage of the environment in path planning, but the question always arises when it comes to decide the path in case of multi-destinations. So in this paper, a new algorithm that focuses on the multi-destination path planning is mentioned. This algorithm is the combination of the particle swarm optimization and the Vortex Search Algorithm (VS) [36]. In the first place, the sequence of the destinations is decided/optimized by using the PSO algorithm. After the VS algorithm is applied to get the collision-free path

Table 1. Simulation 2D environment

Attributes	Values
2D space dimensions	5 m x 5 m
No. of robots	4
No. of obstacles	18
Obstacle dimensions	0.1 m × 0.8 m × 0.1 m

Table 2. Simulation result

Robot No	Distance traveled (m)	Time taken (sec)
1	2.11	23
2	3.06	38
3	2.65	29
4	3.42	49

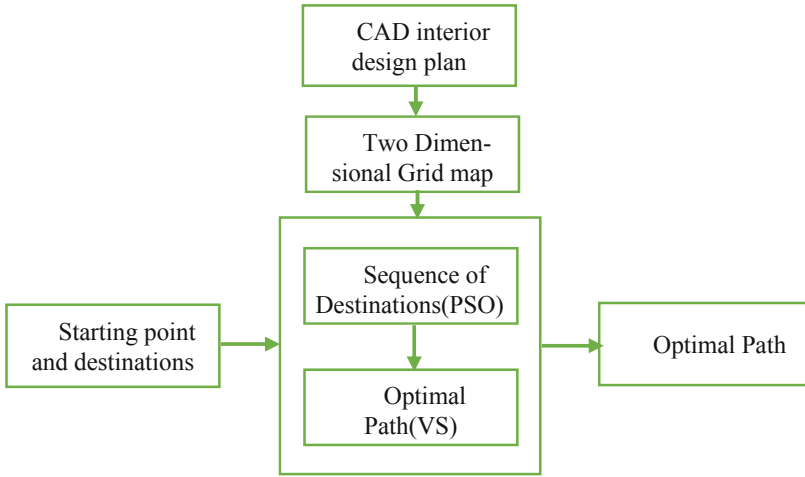


Fig. 3. The proposed algorithm based on PSO and VS

from one destination to the other. By using this algorithm a smooth and short path can be decided efficiently.

Problem Statement and Definitions

The algorithm proposed here is much similar to the Travelling Sales Person (TSP) problems [37].

The framework proposed here is shown in Fig. 5. It is based on PSO and VS algorithms (Fig. 3).

Here a mobile robot is taken as a point. The boundaries of each obstacle are increased by the sum of the size of the robot and the safety distance. After this, the optimal sequence of the destinations is found with the help of PSO. And at last, the path between the destinations is found with the help of the VS algorithm without any collision.

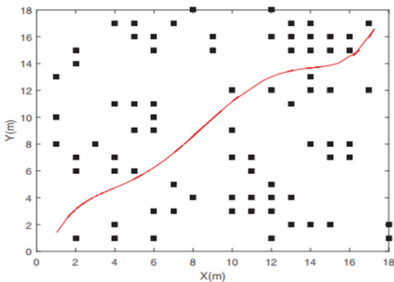


Fig. 4. Path between the source and the target.

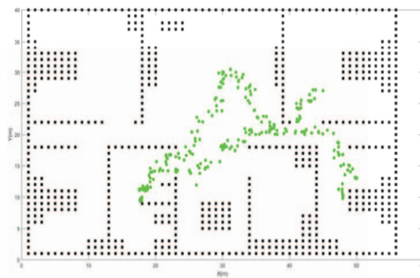


Fig. 5. Iterations in 4 target path planning

The Multi-destinations Approach

The approach goes as: Firstly the best path is decided from source to multi-destination using PSO. Then this sequence is passed to VS as input. Lastly, the path between these destinations is decided using the VS algorithm such that there is no collision [39].

- I. Optimizing the destinations.
- II. Generating the Initial Solution.
- III. Replacing the current solution.
- IV. Adaptive radius Decrement
- V. The smoothness of the path.

The Result

To verify the effectiveness of the proposed algorithm, it is compared with the PSO, VS, and A* algorithm in the same environment (Fig. 4).

As shown in Fig. 10 the iteration speed of VS is more than PSO, which is more than the A* algorithm. The Fig. 14 Shows the iterations involved in the path planning when the number of targets is 4.

5 Result Analysis

See (Tables 3 and 4).

Table 3. Problem classification in various research

Parameters compared							
	No. of targets	Trackers to target ratio	Target mobility	Environment complexity	Prior knowledge of target motion	Cooperation	Coordination
[25]	1	1	Stationary	Complex	No	Uncertainty reduction	Implicit
[35]	≥ 1	≥ 1	Stationary	Complex	No	Uncertainty reduction	Implicit
[38]	> 1	≤ 1	Stationary	Complex	No	Target allocation	Implicit
[40]	> 1	≥ 1	Stationary	Complex	No	Uncertainty reduction	Implicit
[41]	1	≥ 1	Stationary	Complex	Yes	Uncertainty reduction	Implicit
[42]	1	≥ 1	Stationary	Complex	Yes	Uncertainty reduction	Explicit
[43]	1	≥ 1	Stationary	Not complex	Yes	Uncertainty reduction	Implicit

Table 4. Algorithm comparison

	[25]	[35]	[38]	[40]	[41]	[42]	[43]
Decentralized	✓	✓	✓	✓	✓	✓	✓
Local/No communication	✓	✗	✓	✗	✓	✓	✗
No robot Identifiers	✓	✓	✓	✓	✓	✓	✓
No common coordinates	✓	✗	✗	✗	✓	✗	✗
Simple computations	✓	✓	✓	✓	✓	✓	✓
No memory of previous state	✓	✗	✗	✓	✗	✗	✓
Mathematically proven properties	✓	✓	✓	✓	✓	✓	✓
Verified through simulation	✓	✓	✓	✓	✓	✓	✓

6 Conclusion

Due to the robustness, scalable, flexible, fault tolerance, and many other properties of SRSs can be used in many different applications. Many algorithms are described and discussed in this paper. Among the different problems discussed in the paper, the most challenging problem is to solve the local minimum problem. This problem is the most challenging because if the robot is trapped in the local minima the already taken path by the robot becomes erroneous. So the robot has to firstly get out of the local minimum and then again take some other path to reach its target position. For this purpose, a virtual target point method is discussed in this paper.

Few other problems related to optimized path planning for robots in a complex environment are also discussed. Few algorithms which are suitable for some specific type of situations are discussed in the paper. Although there is no algorithm in swarm robotics which optimizes the path for every situation or which can solve all of the problems related to SRS. This is just because of the complex nature of swarms and the highly complex calculations related to swarm.

References

1. Chong, C.-Y., Garren, D., Grayson, T.: Ground target tracking—a historical perspective. In: IEEE Aerospace Conference Proceedings, vol. 3, pp. 433–448 (2000)

2. Parker, L.: Distributed algorithms for multi-robot observation of multiple moving targets. *Auton. Robots* **12**(3), 231–255 (2002)
3. Arkin, R.C.: *Behavior-Based Robotics, Intelligent Robots and Autonomous Agents*. MIT Press, Cambridge (1998)
4. Fukuda, T., Nakagawa, S., Kawauchi, Y., Buss, M.: Self organizing robots based on cell structures—cebot. In: *IEEE International Workshop on Intelligent Robots*, 1988, pp. 145–150 (1988)
5. Fukuda, T., Kawauchi, Y.: Cellular robotic system (cebot) as one of the realization of self-organizing intelligent universal manipulator. In: *1990 IEEE International Conference on Robotics and Automation*, 1990, Proceedings, vol. 1, pp. 662–667 (1990)
6. Bayindir, L., Şahin, E.: A review of studies in swarm robotics. *Turk. J. Electr. Eng. Comput. Sci.* **15**(2), 115–147 (2007)
7. Şahin, E.: Swarm robotics: from sources of inspiration to domains of application. In: Şahin, E., Spears, W.M. (eds.) *Swarm Robotics. Lecture Notes in Computer Science*, vol. 3342, pp. 10–20. Springer, Heidelberg (2005)
8. Martinoli, A.: Collective complexity out of individual simplicity: a review of swarm intelligence: from natural to artificial systems, by eric bonabeau, marco dorigo, and guy theraulaz. *Artif. Life* **7**(3), 315–319 (2001)
9. Sharkey, A.J., Sharkey, N.: The application of swarm intelligence to collective robots. In: Fulcher, J. (ed.) *Advances in Applied Artificial Intelligence*, pp. 157–185. IGI Global (2006)
10. Shimizu, M., Ishiguro, A.: A self-reconfigurable robotic system that exhibits amoebic locomotion. In: *IEEE/ICME International Conference on Complex Medical Engineering*, 2007, CME 2007, pp. 101–106 (2007)
11. Shimizu, M., Ishiguro, A.: An amoeboid modular robot that exhibits realtime adaptive reconfiguration. In: *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 1496–1501. IEEE Press, Piscataway (2009)
12. Derr, K., Manic, M.: Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. In: *2nd Conference on Human System Interactions 2009, HIS 2009*, pp. 81–86 (2009)
13. Pugh, J., Martinoli, A.: Inspiring and modeling multi-robot search with particle swarm optimization. In: *Swarm Intelligence Symposium, 2007, SIS 2007*, pp. 332–339. IEEE (2007)
14. Jevtić, A., Gazi, P., Andina, D., Jamshidi, M.: Building a swarm of robotic bees. In: *World Automation Congress (WAC)*, 2010, pp. 1–6 (2010)
15. Jevtić, A., Gutierrez, A., Andina, D., Jamshidi, M.: Distributed bees algorithm for task allocation in swarm of robots. *IEEE Syst. J.* **6**(2), 296–304 (2012)
16. Garcla, M.A.P., Montiel, O., Castillo, O., Seplveda, R.: Optimal path planning for autonomous mobile robot navigation using ant colony optimization and a fuzzy cost function evaluation. *Appl. Soft Comput.* **9**(3), 1102–1110 (2009)
17. Masehian, E., Sedighizadeh, D.: Classic and heuristic approaches in robot motion planning: a chronological review. In: *Proceedings of World Academy of Science Engineering & Technology*, pp. 101–106 (2007)
18. Lu, Q., Han, Q.-L., Zhang, B., Liu, D., Liu, S.: Cooperative control of mobile sensor networks for environmental monitoring: an eventtriggered finite-time control scheme. *IEEE Trans. Cybern.* (2017). <https://doi.org/10.1109/TCYB.2016.2601110>
19. Wang, M., Wang, X.R., LI, C.G., Zhang, Z.F.: Study of local path planning of mobile robot based on improved artificial potential field method. *Comput. Eng. Des.* **29**(6), 48–53 (2008)
20. Rui, M., Su, W.J., Lian, X.F.: Mobile robot path planning based on dynamic fuzzy artificial potential field method. *J. Inf. Comput. Sci.* **9**(17), 5233–5240 (2010)
21. Wang, M., Liu, J.N.K.: Fuzzy logic based robot path planning in unknown environment. In: *Proceedings of the International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005*, pp. 813–818 (2005)

22. Chen, Y., Lu, Q., Yin, K., Zhang, B., Zhong, C.: PSO-based receding horizon control of mobile robots for local path planning. In: Proceedings of the 43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November, pp. 1–6 (2017)
23. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
24. Ma, L., Forouraghi, B.: A modified particle swarm optimizer. In: Advances in Natural Computation, pp. 439–439 (2012)
25. Chen, D., Lu, Q., Yin, K., Chen, Y.: A method for solving local minimum problem of local path planning based on particle swarm optimization. In: 2017 Chinese Automation Congress, pp: 4944–4949 (2017)
26. Aouf, A., Boussaid, L., Sakly, A.: TLBO-based adaptive neurofuzzy controller for mobile robot navigation in a strange environment. *Comput. Intell. Neurosci.* (2018)
27. Faisal, M., Algabri, M., Abdelkader, B.M., Dhahri, H., Al Rahhal, M.M.: Human expertise in mobile robot navigation. *IEEE Access* **6**, 1694–1705 (2018)
28. Karakose, M., Baygin, M., Baygin, N., Murat, K., Akin, E.: An intelligent reconfiguration approach based on fuzzy partitioning in PV arrays. In: Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium, pp. 356–360 (2014)
29. Karakose, M., Baygin, M., Parlak, K.S.: A new real-time reconfiguration approach based on neural network in partial shading for PV arrays. In: International Conference in Renewable Energy Research and Application (ICRERA), pp. 633–637 (2014)
30. Yaman, O., Karakose, M., Akin, E.: PSO based diagnosis approach for surface and components faults in railways. *Int. J. Comput. Sci. Softw. Eng.* **5**, 89–96 (2016)
31. Hu, H., Cui, X., Bai, Y.: Two kinds of classifications based on improved gravitational search algorithm and particle swarm optimization algorithm. *Adv. Math. Phys.* (2017)
32. Tillett, J., Rao, T., Sahin, F., Rao, R.: Darwinian particle swarm optimization (2005)
33. Lu, S., Zhao, J., Jiang, L., Liu, H.: Solving the time-jerk optimal trajectory planning problem of a robot using augmented Lagrange constrained particle swarm optimization. *Math. Probl. Eng.* (2017)
34. Grandi, R., Falconi, R., Melchiorri, C.: A particle swarm optimization-based multi robot navigation strategy. In: International Workshop on Bio-Inspired Robots (2011)
35. Baygin, N., Baygin, M., Karakose, M.: PSO based path planning approach for multi service robots in dynamic environments. In: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (2018)
36. Dogan, B., Olmez, T.: A new metaheuristic for numerical function optimization: vortex search algorithm. *Inf. Sci.* **293**, 125–145 (2015)
37. Chen, X., Zhou, Y., Tang, Z., Luo, Q.: A hybrid algorithm combining glowworm swarm optimization and complete 2-opt algorithm for spherical travelling salesman Problems. *Appl. Soft Comput.* **58**, 104–114 (2017)
38. Liu, P., Wang, C., Qin, H., Sun, J.: A optimal path planning of multi-destinations for mobile robot in complex environment. In: 2018 Chinese Automation Congress (CAC) (2018)
39. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **SSC4** **4** (2), 100–107 (1968)
40. Heo, S., Lu, S., Shin, J., Lee, H.: Multi-robot-multi-target path planning and position estimation for disaster area. In: 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT) (2018)
41. Liao, L., Cai, X., Huang, H., Liu, Y.: Improved dynamic double mutation particle swarm optimization for mobile robot path planning. In: 2016 Chinese Control and Decision Conference (CCDC) (2016)

42. Qi, B., Xiong, L., Wang, L., Chen, Z., Huang, L.: A weights and improved adaptive artificial fish swarm algorithm for path planning. In: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC) (2019)
43. Yong, L., Yu, L., Yipei, G., Kejie, C.: Cooperative path planning of robot swarm based on ACO. In: 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (2017)
44. Li, X.L.: A New Intelligent Optimization Method—Artificial Fish School Algorithm. Zhejiang University (2003)
45. Yao, Z., Ren, Z., Chen, Y.: Path planning for mine rescue robot based on AFSA. *Coal Mine Mach* (2014)
46. Yao, Z., Ren, Z.: Path planning for coalmine rescue robot based on hybrid adaptive artificial fish swarm algorithm. *Int. J. Control Autom.* **7**(8), 1–12 (2014)
47. Huang, Y.Q., Peng, K., Yuan, M.L.: Path planning for mobile robots based on multi-strategy hybrid artificial fish swarm algorithm. *Inf. Control* **46**(3), 283–288 (2017)
48. Zhang, L.L., Dai, Y.M.: Adaptive artificial fish swarm algorithm. *Comput. Eng. Sci.* **38**(09), 1894–1900 (2016)
49. Wu, C.Y.: An improved artificial fish swarm optimization algorithm. *J. Intell. Syst.* **3**, 465–469 (2015)
50. Zhang, Y., Guan, G., Pu, X.: The robot path planning based on improved artificial fish swarm algorithm. *Math. Probl. Eng.* **2016**(11), 1–11 (2016)
51. Peng, J., Li, X., Qin, Z.Q., et al.: Robot global path planning based on improved artificial fish-swarm algorithm. *Res. J. Appl. Sci. Eng. Technol.* **5**(6), 2042–2047 (2013)



IPL: From Lens of Data Science

Rahul Pradhan, Drashti Maheshwari, Mayank Aggarwal[✉], Ankur Chaturvedi,
and Dilip Kumar Sharma

Department of Computer Engineering and Applications, GLA University, Mathura, India
{rahul.pradhan, ankur.chaturvedi, dilip.sharma}@gla.ac.in

Abstract. Cricket is most lovable game in the world. It is played in many formats. One of the most popular format is IPL (Indian Premier League) which is hosted by BCCI (Board of Control for Cricket in India) every year. IPL is involved in business and money as the players across the globe are auctioned by the businessmen to form their teams. This paper presents the analytics of IPL match using the dataset of previous year matches i.e. from year 2008 to 2019. Various attributes like DLS method applied, venue, toss decision, toss winner and many more have been analyzed to check whether they contribute in predicting the winner or not. In this work, some graphs has also been plotted to visualize the performance of teams and players on the basis of different attributes.

Keywords: Data analysis · Python · Pandas · Numpy · Matplotlib · Seaborn · IPL · Cricket analysis

1 Introduction

Data analysis is taking its place in almost every field or in every industry. Sports is also one of the field in which data analysis proved itself very useful in analyzing the performance, predicting the future outcomes and taking some decisions based on the analytics of the game. As we know, Cricket is the most popular game in the world. In almost every country, we can find the fans of this game. According to International Cricket Council, the different formats in which cricket can be played are One Day International matches, the Test matches and the T20 International. Meanwhile, in the year 2008 in India, the Board for Control of Cricket in India (BCCI, the organization which takes care of cricket) introduced a new format of cricket “Indian Premier League (IPL)” in which the businessmen can make their teams by auction Data analysis is taking its place in almost every field or in every industry. Sports is also one of the field in which data analysis proved itself very useful in analyzing the performance, predicting the future outcomes and taking some decisions based on the analytics of the game. As we know, Cricket is the most popular game in the world. In almost every country, we can find the fans of this game. According to International Cricket Council, the different formats in which cricket can be played are One Day International matches, the Test matches and the T20 International. Meanwhile, in the year 2008 in India, the Board for Control of Cricket in India (BCCI, the organization which takes care of cricket) introduced a new format of cricket “Indian Premier League (IPL)” in which the businessmen can make

their teams by auctioning the players around the globe. The different teams then play this league every year. Every team has a composition of players of different countries across the globe.

Because of involvement of businessmen in this particular format of the game, this league becomes the part of business. Analytics is the backbone of some businesses. This league is one of them, as the team owners need to analyze the performance of their team and the players playing for them. There are various tools by which we can analyze any data. Python is one of those tools. Python has many inbuilt packages, with the help of pre-defined functions of these packages, we can analyze any data set. For example, in pandas, we have many pre-defined functions like, shape, reshape, drop, fillna, and many more which helps us in manipulating and analysis of the data.

The approaches used in this paper help us in determining the attributes which counters any effect on the performance of the team or the player. This work also presents the statistical analysis of the IPL data which includes the extraction of information from various attributes.

The structure of this paper is described below:

Section 2 is the related work done in this field, Sect. 3 is the background study done in this research, Sect. 4 is the methodology of the research, Sect. 5 is the experimental outcomes obtained in this research and in Sect. 6 the research has been concluded.

2 Related Work

Python has become a popular programming language from a few years. It is widely used in the field of Data Science for analyzing the data and for prediction. Many other tools can also be used for analysis such as R, SQL. In [5], it is mentioned by the author that Python has a variety of inbuilt packages for mathematical and scientific purposes such as numpy, pandas, sklearn, scipy, etc. As data is increasing exponentially, the need of data analysis is continuously evolving, so we need some techniques for data analysis. Data analysis is used to turn the data into some useful information [6]. Visualizing data makes the analysis of data more interesting. Python has a variety of data visualization libraries. According to [7], Data visualization is done with the help of charts, graphs, plots, etc. In [8], the author says that data visualization is as important task as to reading or writing the data.

Analytics has covered almost all the fields. Sports has become a major field in them [9]. Being very popular sport, Cricket has a huge fan base around the globe. Cricket has different formats like ODI, Test, T20 International. In India, IPL is most popular format. It is similar to T20I. The author in [10], analyses the attributes which effects the performance of players. In [11], various machine learning techniques and predictive results has been compared. The different approaches used in [12] have done the analysis of T20 matches in depth. Data Analysis has an interesting application in predicting and analyzing sports [16].

To analyze the IPL matches, we have some other approaches also [13, 14]. All the players cannot perform well simultaneously, so in IPL as well as in other formats, the change in composition of players in team is necessary [15]. In this paper, we have analyze the attributes of IPL dataset with the help of libraries of Python by plotting different graphs, plots and executing different queries written in Python to get the desired results.

3 Background Study

Data analysis is the major process involved in this project. Analyzing any dataset is a bit more complex task as compared to other tasks such as collecting data, manipulating it, making end results in the form of graphs and reports, etc. This IPL data is analyzed with the help of pre-built libraries available in Python such as pandas, numpy, matplotlib, seaborn. The various libraries used in this project as described below.

3.1 Libraries Used

- i. pandas
Pandas is a pre-built Python library which is used to manipulate data in terms of changing or modifying the data types and data formats, handling missing values, smoothening of the data, maintaining the consistency of the data. According to [1], pandas is a library which is developing since 2008, it acts as a bridge of data analysis tools between python and scientific and statistical computing.
- ii. numpy
Numpy is also helpful in wrangling or manipulating data. It focuses on mathematical and logical manipulation in the data. Numpy has a variety of inbuilt mathematical functions to perform various operations in the dataset. It can create multidimensional arrays to store and manipulate the data [2] while in pandas, it is preferable to store data in series and data frames before applying pandas functions.
- iii. matplotlib
Matplotlib is a package in Python with the purpose of plotting the multidimensional graphs (2D or 3D). It is capable enough to plot scientific as well as mathematical graphs and figures in different formats [3]. The motivation for designing it was to provide a simple and easy package to plot simple plots with only lines of code for the users. It also has some pre-built functions for the ease of users.
- iv. seaborn
Seaborn is also a library for plotting the multidimensional graphs. This library is basically based on matplotlib. The only difference is that it provides a better and high level interface for plotting attractive and clear quality scientific and statistical figures [4]. Python and R have many similarities among them. If we compare libraries then, seaborn is similar to ggplot2 available in R.

3.2 Dataset

This dataset on IPL is downloaded from www.kaggle.com (a website where users can publish as well as download the data sets, participate in various data-science challenges or can work other machine learning engineers, data science enthusiasts). The dataset is available in CSV format. This dataset consists of 756 records where each record signifies a match conducted in the league. The dataset has records from the year 2008 to 2019. The dataset can be downloaded from the link <https://www.kaggle.com/nowke9/ipldata#matches.csv>.

The dataset have 18 different attributes. The various attributes available in this dataset are:

- id: the unique id of the match conducted
- season: the year in which match was conducted
- city: city in which match was conducted
- date: date on which match was conducted
- team1: first team which was playing the match
- team2: opponent team playing the match
- toss_winner: toss winner team
- toss_decision: toss decision taken by team captain i.e., batting or bowling
- result: result of match i.e., normal, tie or no result
- dl_applied: whether the Duckworth-Lewis-Stern method (DLS) applied or not
- winner: winning team of the match
- win_by_runs: winning team won by how many runs
- win_by_wickets: winning team won by how many wickets
- player_of_match: the player awarded as “player of match”
- venue: stadium in which match was played
- umpire1: 1st umpire of the match
- umpire2: 2nd umpire of the match
- umpire3: 3rd umpire of the match

4 Methodology

First the dataset or input data i.e. IPL dataset was imported into Jupyter Notebook with the help of pandas library of Python. This Jupyter Notebook file will be saved as a JSON file in the same directory. The extension of this Jupyter Notebook is.ipynb. After that, some queries were executed to analyze this IPL data. The different queries analyses the different attributes of the dataset such as winning toss, toss decision, win by runs, win by wickets, etc.

The queries which we have performed in this project are as follows:

In the following queries, df stands for the data frame in which the input data is stored.

1). Aim:

Count the total number of matches conducted in the year 2008.

Query:

```
df[df['season'] == 2008].season.count()
```

2). Aim:

Find the city name where maximum and minimum number of matches conducted.

Query:

```
temp = pd.DataFrame(df.city.value_counts())
temp.loc[temp['city'] == max(temp['city'],'city')]
temp.loc[temp['city'] == min(temp['city'],'city')]
```

3). Aim:

Find total count of matches citywise.

Query:

```
print(pd.DataFrame(df.city.value_counts()))
```

4). Aim:

Team which is maximum and minimum toss winner.

Query:

```
temp=pd.DataFrame(df['toss_winner'].value_counts())
temp.loc[temp['toss_winner']== max(temp['toss_winner'])]
temp.loc[temp['toss_winner']==min(temp['toss_winner'])]
```

5). Aim:

Check the toss decision that the team has taken.

Query:

```
df.loc[:,['toss_winner','toss_decision']]
```

6). Aim:

Count the normal and tie matches.

Query:

```
df['result'].value_counts()
```

7). Aim:

Find the team names where the match result is tie.

Query:

```
df.loc[df['result']=='tie',['team1','team2']]
```

8). Aim:

Count the 0 and 1 in dl_applied column.

Query:

```
df['dl_applied'].value_counts()
```

9). Aim:

Find the team name who won the match by highest runs.

Query:

```
df.loc[df['win_by_runs'].idxmax(),['winner','win_by_runs']]
```

10). Aim:

Find the team name who won the match by lowest runs.

Query:

```
df.loc[df[df['win_by_runs'].ge(1)].win_by_runs.idxmin(),['winner','win_by_runs']].
```

11). Aim: Players who was awarded “Player of the match” more than 3 times.

Query:

```
temp_data=pd.DataFrame(df['player_of_match'].value_counts()).
```

```
temp_data.loc[temp_data['player_of_match']>3,'player_of_match'].
```

12). Aim:

Find the player who was awarded as player of the match maximum times.

Query:

```
pd.DataFrame(temp_data.loc[temp_data['player_of_match']==max(
temp_data['player_of_match']), 'player_of_match']).
```

13). Aim:

Venue where the team won the match by highest runs.

Query:

```
pd.DataFrame(df.loc[df['win_by_runs']==max(df['win_by_runs']),'venue']).
```

14). Aim:

Venue where the team won the match by lowest runs.

Query:

```
pd.DataFrame(df.loc[df['win_by_runs']==min(df['win_by_runs']),'venue']).
```

15). Aim:

Umpires who did umpiring maximum times.

Query:

```
data1=df['umpire2'].value_counts().
```

```
data2=df['umpire1'].value_counts().
```

```
df2=pd.concat([data1,data2],axis=1,join='outer').
```

```
df2.fillna(0,inplace=True).
```

```
df2['new']=df2['umpire1'] + df2['umpire2'].
```

```
print(df2[df2['new']==max[df2['new']]]['new']).
```

5 Experimental Results

Other than the queries mentioned above, we have also plotted some graphs to clear the scenario of analysis.

Figure 1 represents the total number of matches played in different seasons and this graph clearly shows that the maximum number of matches conducted was in year 2013.

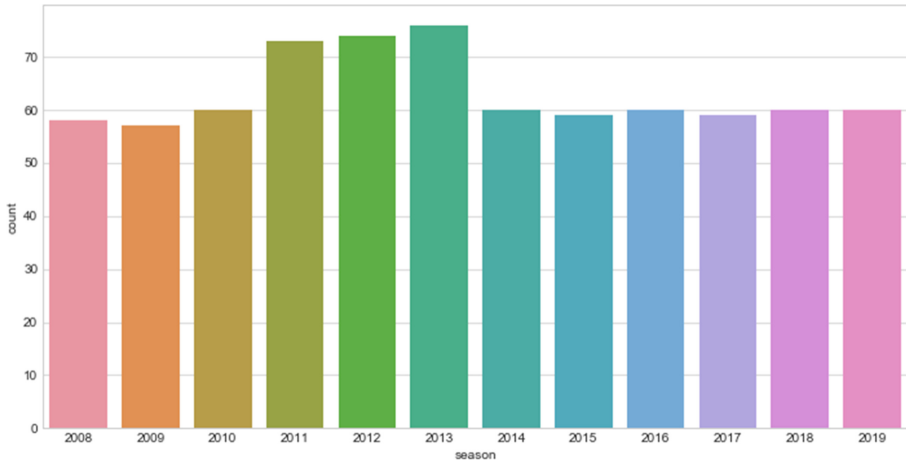


Fig. 1. Number of matches played in different seasons.

Figure 2 shows that how many matches have won by different teams. We can see that Mumbai Indians has won the maximum number of matches.

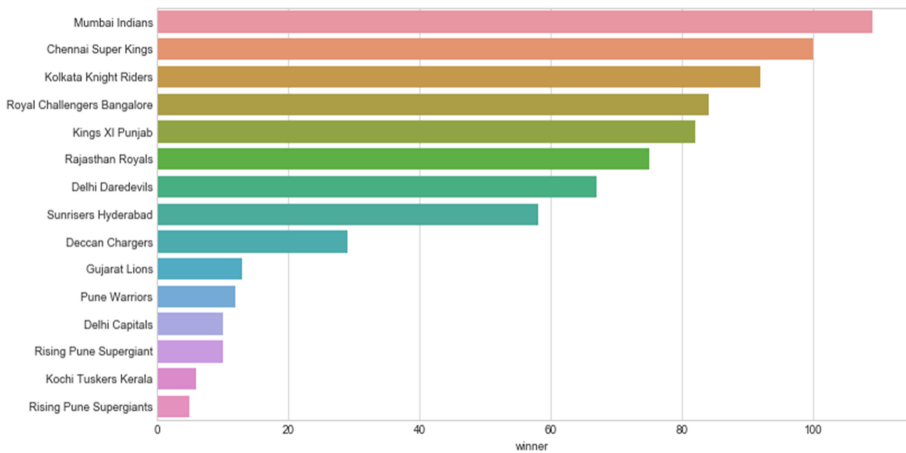


Fig. 2. Number of match winnings of different teams.

The above figure Fig. 3 shows number of “player of the match” title won by any player. CH Gayle has won maximum number of “Player of the Match” title.

In Fig. 4, we have find the relationship between match winning and toss wining and it shows that the toss winning does not have relationship with match winning as the number of match won and match lose after winning the toss are almost equal.

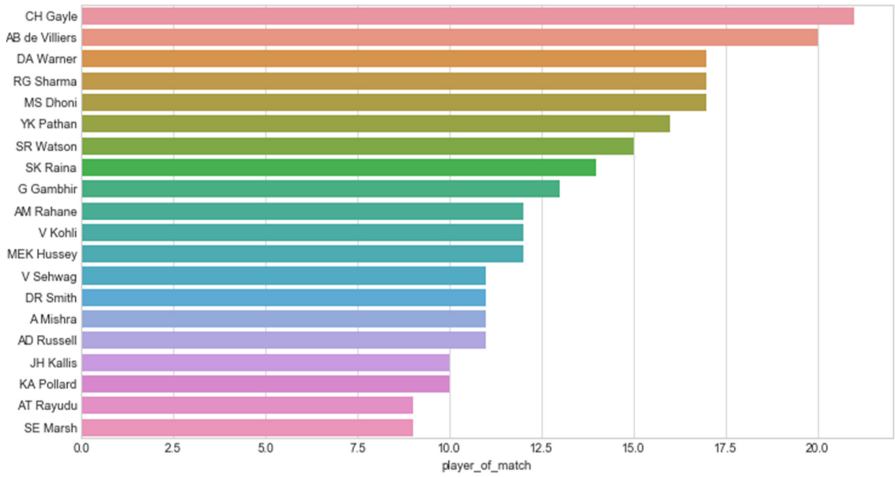


Fig. 3. Count of players awarded as “Player of the match” title.

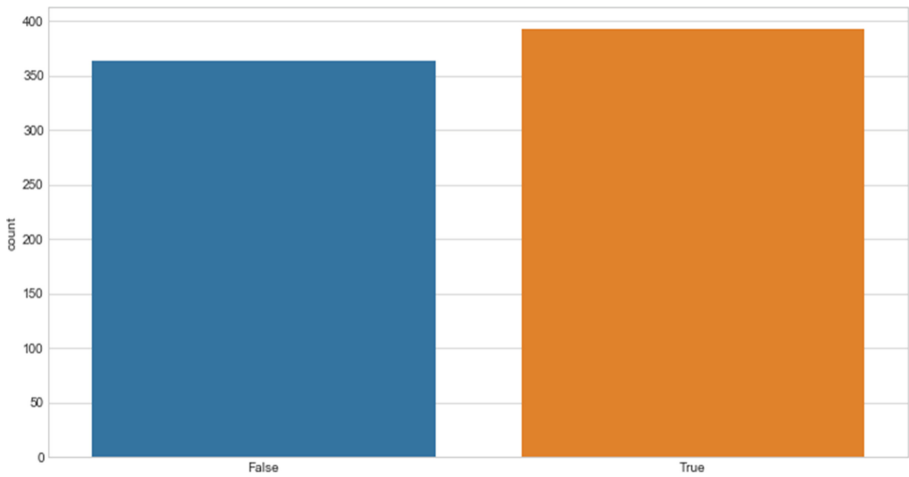


Fig. 4. Relationship between match winning and toss winning

Fig. 5(a)–(l) shows the performance or match winnings of the teams in different cities and it can be observed that any team has won the maximum number of matches in their home ground. These results can be seen from year 2008 to 2019. In all the seasons this influence remained unchanged. Hence, we can say that teams perform better at their home grounds.

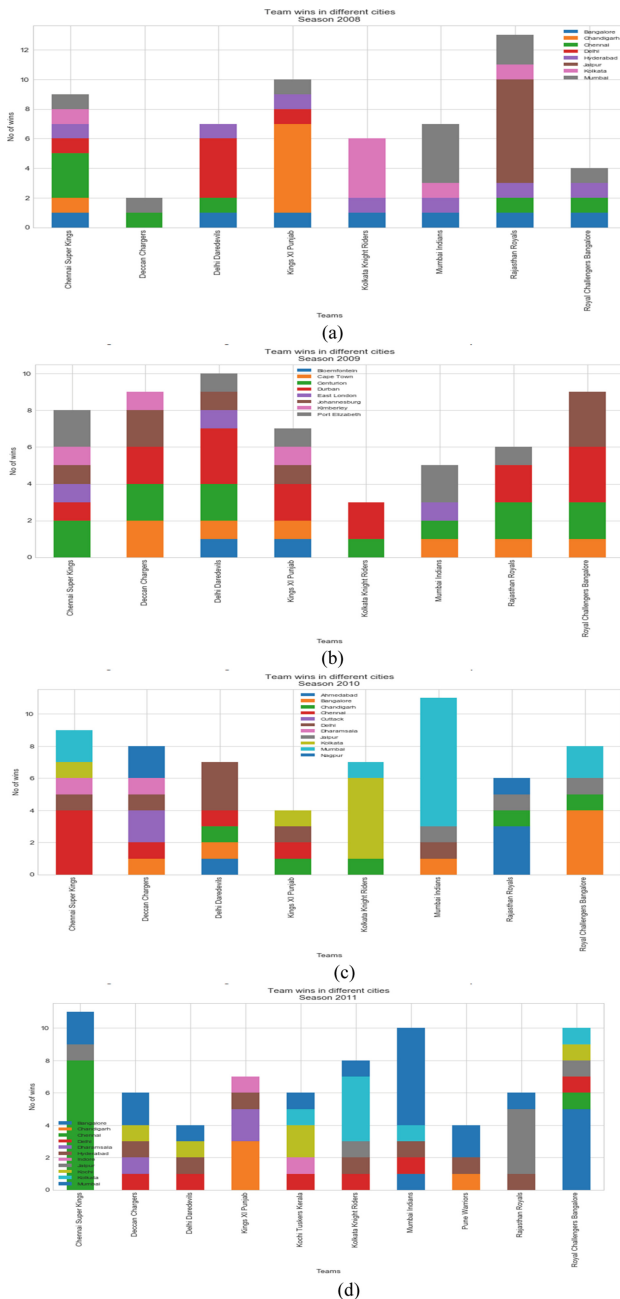
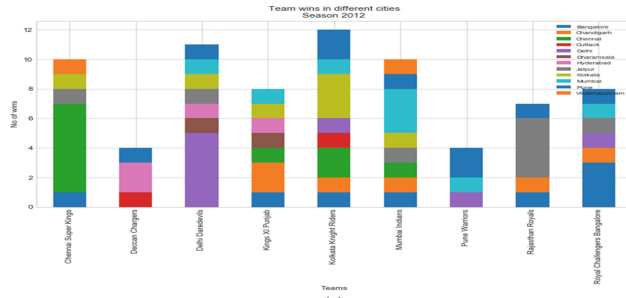
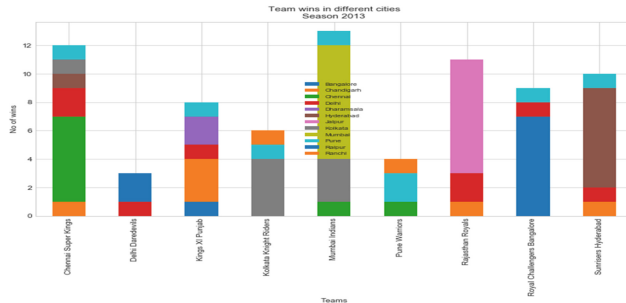


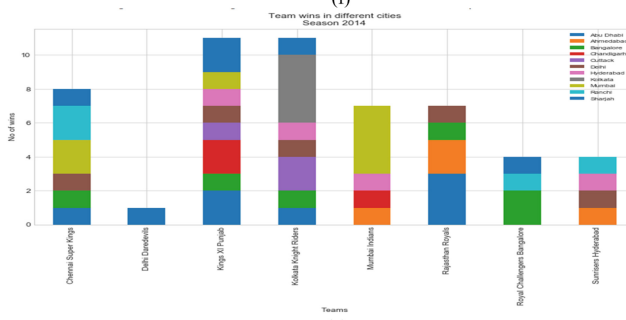
Fig. 5. (a) Winnings of teams in different cities in year 2008. (b) Winnings of teams in different cities in year 2009. (c) Winnings of teams in different cities in year 2010. (d) Winnings of teams in different cities in year 2011. (e) Winnings of teams in different cities in year 2012. (f) Winnings of teams in different cities in year 2013. (g) Winnings of teams in different cities in year 2014. (h) Winnings of teams in different cities in year 2015. (i) Winnings of teams in different cities in year 2016. (j) Winnings of teams in different cities in year 2017. (k) Winnings of teams in different cities in year 2018. (l) Winnings of teams in different cities in year 2019



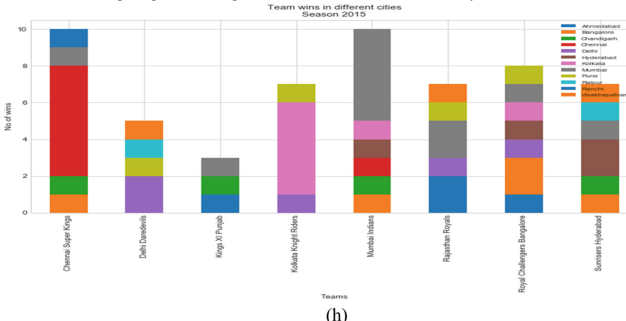
(e)



(f)

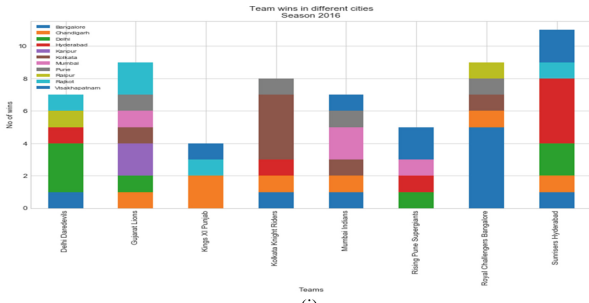


(g)

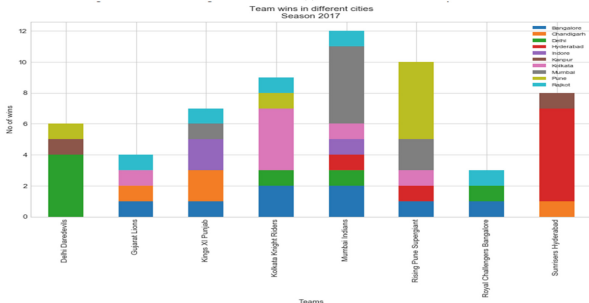


(h)

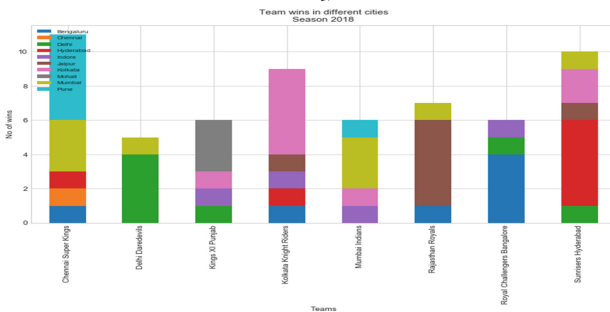
Fig. 5. (continued)



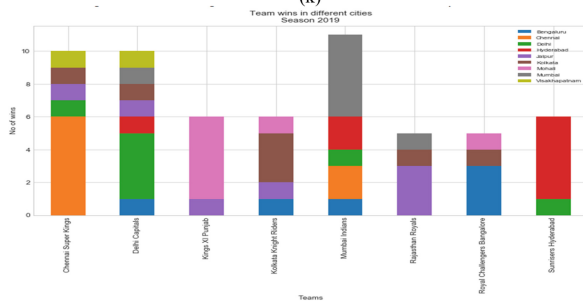
(i)



(j)



(k)



(l)

Fig. 5. (continued)

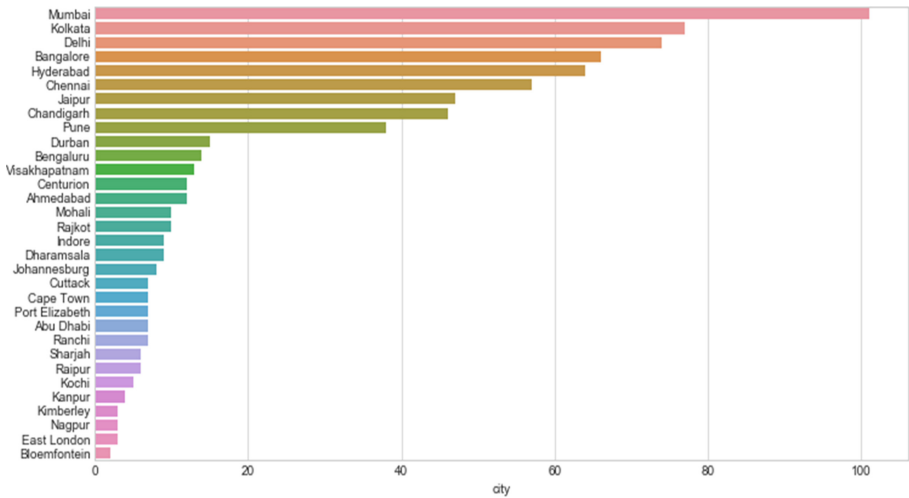


Fig. 6. Number of matches played in different cities

In Fig. 6, it shows number of matches played in different cities and it can be seen that maximum number of matches played were in Mumbai.

The analysis done in this project can be summarized as:

- There is no effect of toss winning on match winning.
- Mumbai Indians has won maximum number of matches.
- Chris Gayle has won maximum number of “Player of the match title”.
- Maximum number of matches played were in season 2008.
- The performance of team on their home ground is better than other venues.

6 Conclusion and Future Work

In this paper, we have showed analysis on IPL dataset using Python libraries. The analysis done in this paper has the purpose to find out some hidden insights and pattern in this dataset which can help in decision-making. For this particular dataset, we have found that performance of the teams in their home ground is better than other venues. We have showed the almost no relationship between toss winning and match winning. Mumbai Indians has won the maximum number of matches and on this basis, we can say that Mumbai Indians is the most successful team of the league. This work can further be extended to more accurate analysis on bigger datasets having more attributes, also more queries can be executed by Python or any other tool to have more insights of the league.

References

1. McKinney, W.: pandas: a foundational Python library for data analysis and statistics. Python High Perform. Sci. Comput. **14**, 1–9 (2011)

2. Lemonkova, P.: Processing oceanographic data by Python libraries numpy, scipy and pandas. *Aquatic Res.* **2**(2), 73–91 (2019)
3. Ari, N., Ustazhanov, M.: Matplotlib in Python. In: 11th International Conference on Electronics, Computer and Computation (ICECCO) (2014). <https://doi.org/10.1109/ICECCO.2014.6997585>
4. Singh, P.K., Panigrahi, B.K., Suryadevara, N.K. Sharma, S.K., Singh, A.P. (eds.): Proceedings of ICETIT 2019. *Lecture Notes in Electrical Engineering* (2020). <https://doi.org/10.1007/978-3-030-30577-2>
5. Venayak, N., Raj, K., Mahadevan, R.: Impact framework: a Python package for writing data analysis workflows to interpret microbial physiology. *Metab. Eng. Commun.* **9**, e00089 (2019)
6. Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., Nillaor, P.: IoT and agriculture data analysis for smart farm. *Comput. Electron. Agric.* **156**, 467–474 (2019). <https://doi.org/10.1016/j.compag.2018.12.011>
7. Cardno, A., Ingham, P., Lewin, P., Singh, A.: Data Visualization methods. Google Patents
8. Borner, K., Bueckle, A., Ginda, M.: Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *PNAS* **116**(6), 1857–1864 (2019). <https://doi.org/10.1073/pnas.1807180116>
9. Singh, S., Kaur, P.: IPL visualization and prediction using HBase. In: *Information Technology and Quantitative Management (ITQM 2017)*
10. Kumar, G.: *Machine learning for soccer analytics* (2013)
11. Fournier-Viger, P., et al.: The SPMF Open-Source Data Mining Library Version 2. In: Berendt, B., et al. (eds.) *Machine Learning and Knowledge Discovery in Databases*, pp. 36–40. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46131-1_8
12. Hijmens, A., Bhulai, S.: Dutch football predictions using machine learning classifiers (2017)
13. Prakash, D., Patvardhan, C., Singh, S.: A new machine learning based deep performance index for ranking IPL T20 cricketers. *Int. J. Comput. Appl.* **137**(10), 42–49 (2019)
14. Kalgotra, P., Sharda, R., Chakraborty, G.: Predictive modelling in sports league: an application in Indian premier league. *SSRN Electron. J.* (2013). <https://doi.org/10.2139/ssrn.24655300>
15. Jhanwar, M., Pudi, V.: Predicting the outcome of ODI cricket matches: a team composition based approach. In: *European Conference of Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2016)* (2016)
16. Chaudhary, K., Gupta, M., Kaur, P.: Analyzing IPL dataset with MongoDB. In: 9th International Conference on Cloud Computing, Data Science and Engineering (Confluence), Noida, India, pp. 212–216 (2019)



Outlier Detection in Wireless Sensor Networks with Denoising Auto-Encoder

Bhanu Chander^(✉) and Kumaravelan

Department of Computer Science and Engineering, Pondicherry University, Pondicherry, India

Abstract. Anomaly detection is an effective approach of dealing with problems in decision making process. Rapid development in technology has elevated the requirement of resourceful detection system using machine learning, deep learning in order to detect new and advanced outliers. Most of the real time applications employed with wireless sensor networks, which are positioned in unkind and unattended atmospheres, where these situations turns to a major causes on the production of anomalous or low quality sensor readings. The erroneous and unreliable readings may increase generation of false alarms and erroneous decisions; hence it is essential to identify outliers resourcefully and exactly to make sure the authentic decision-making. In this paper, first simple cluster algorithm performed based on residual energies of sensor nodes and then denoising auto-encoder with Gaussian kernel applied on each cluster head to detect outliers. Experimental analysis shows that designed technique achieves high detection rate as well as low false alarm rate.

Keywords: Outlier detection · Clustering · Auto-encoder · Network life-time

1 Introduction

A data instance can be acknowledged as Outlier, when it turn aside from normal or known behaviors of data, when its characteristics are not related to other data instances and when assumed values are extreme away from the predictable values [1–4]. Hence, outlier show signs of various out of the way activities; such kind of activities plays an essential task in decision-making. Like a character assumes it as a leadership and manipulate others to follow, identifying a person’s cruel behavior in a group of people. Therefore, depending on the appliances outlier can be used at both positive as well as negative roles. From past years, utilization of Wireless Sensor Networks (WSNs) has been increased in a variety of domains such as environmental monitoring, civil and military appliances, smart cities, agriculture and smart grid etc. [1–6]. But the harsh deployment surroundings and the natural constrained resources of sensor nodes including power, CPU processing, bandwidth, memory, computation and transmission channel make the WSNs more vulnerable to dissimilar kind of attacks, misbehaviors and outliers [4–8]. In the view of WSNs, anomaly or outlier is defined as the measurement that significantly deviates from the ordinary blueprint of the sensed data and chances of generating suspicious atmosphere. In WSNs, the main reasons for outliers are imperfect sensor

nodes, security threats in system connections, or unexpected event in the monitoring range. Hence, it clears that inadequate resources plus installed harsh environments may possibly reasons for raw observations composed by the sensors to have deprived data feature and trustworthiness, which will manipulate the accurateness of the analysis and decision-making in WSNs [4, 5, 7–9]. Thus, outlier exposure must be put into action on data instances composed by sensor nodes. The correct detection of outliers in sensor data may increase the exact detection of upcoming events, more prominently valuable decisions can be prepared by information gatherers.

2 Related Work

At present outlier detection gains so much of interest from researchers around the world. Because outlier detection has been used in various research fields such as environmental event detection, industrial fault detection, health disease detection, intrusion detection, fraud detection etc., where detection of outliers depend on the dissimilar characteristics of data or datasets [10–15]. An outlier detection technique discovers any irregular, non characteristic prototypes in any dataset that do not fit in with estimated activities. still, a basic problem is that there is no simple exceptional definition that permits us to calculate how related are two data points, and hence how special is one data point from others in the data set. Many researchers has designed several outlier detection models especially for WSNs, such as statistical base models, nearest neighbor base models, clustering base models and classification base models [4–6, 8–12].

Preliminary work of outlier's detection comes on the bases of statistical methods. Based on the principles of statistics, outliers are the measurements or annotations those appear statistically incompatible with the rest of data records. In statistical base models, a statistical prototype is applied to estimate data distribution, and estimate each and every data sample in terms of their aptness for the model. [8] Planned histogram based outlier detection in WSNs. In place of transform every sensed raw data to cluster head; every individual sensor collects hits of data distribution and forward to central decision node. Model shows great detection rate but this process evaluates one-dimensional readings. [9] offered a statistical technique designed to classify suitable annotations in data streams plus distinctive sensor failures in WSNs; technique utilizes spatial, temporal correspondences of the sensed data in real-time. Due to the employ of real-time approach, it has high computation complexity of the outlier detection, which will use more power and shrink the existence of the set-up. But this kind of approaches face some issues like most of the approaches build on single-dimensional simply best results for univariate data and finding the right model for high dimensional data is a hectic problem. Present era problems must represent multi-dimensional data to accomplish outlier detection capacity.

Nearest neighbor techniques are frequently applicable in data mining as well as machine learning domains where data instance analysis prepared with respect to its nearest neighbors. According to these methods, normal data points very much close to the neighborhood while outliers are those far away from their neighbors. [10] in this authors employed distributed saddle point learning procedure, where sensor nodes in separate area contribute the targets and learning statistics to other sensor through

communication. When a sensor far away from target location notified as outlier and it has good detection rate but computation is high. [11, 12] are some other distance outlier techniques with good detection rate. But these approaches not healthier to WSNs because the computation of distance among multivariate datasets is too much expensive which could not be offered by resource-constrained WSNs. Distance to the K nearest neighbor, Average distance to the K -nearest neighbor, the sum of the distances to its k nearest neighbor is some of the frequent techniques available on this approach.

Clustering approaches broadly applicable for group similar data instances into clusters. Basically, in clustering, data divides into dissimilar parts and similar instances groups into one cluster and data instances of one cluster are completely dissimilar from other clusters, here the entire process done exclusive of any prior information of data distribution. Clusters which are smaller in size than remained clusters or those data points which are not taken part in any clusters are recognized as outliers. [13] developed Ordering Points to Identify the Clustering Structure (OPTICS) and [14] designed K -medoids for outlier detection with low false positive rates. These approaches facing difficulties inappropriate cluster width and distance among data points. In addition, calculating space between data points in multivariate data is computationally costly. Modified K -means clustering, density-based subspace clustering, single linkage hierarchical clustering, cluster-based local outlier factor, and local density-based spatial clustering are some of the well-known techniques available under this approach. [20] based on the significance of sensed measurements collected by WSNs, a new Outlier detection (OD) process dubbed density-based spatial clustering of applications with noise (DBSCAN) – OD has progressed on the basis of DBSCAN as backdrop. On the working condition of DBSCAN, two procedures united, first one compute parameters, and second concerns class identification in spatial temporal databases. From these two procedures one judge as real-time application in central station to separate outliers from normal data.

Classifications approaches try to learn or build a trained model by training with a couple of data instances, thereafter; make differentiate when an unseen data instance comes into learned or trained model class. But the issue is models consume high computation resources and complex to obtain the label data. However, the one-class classifier is suitable for WSNs with some limitations, appropriate parameters and choice of accurate kernel functions. Recent studies stated that, as a talented machine learning tool to handle the accurate pattern recognition from complex raw data, deep learning (DL) is fetching as potent technique to insert intelligence to WSNs despite of their multipart topology, communication conditions. DL uses the similar structure of human brain to achieve sensitive feature extraction from high-dimensional data. DL finds accurate network dynamics like traffic bottlenecks, interference distribution, spectrum availability, hotspots etc. based on the analysis of a large amount of network parameters like as loss rate, delay etc. Hence, DL can evaluate exceptionally composite WSNs with numerous nodes and dynamic link quality.

[15] Authors developed deep-structure auto-encoder neural networks to identify outlier spectrum, and time–frequency illustration is acted as the features of the learning model. Threshold is employed to differentiate the outliers from normal data. Lastly, authors study the performance of the proposed model with dissimilar number of hidden layers. [16] Authors comes with the intension of forward a non-linear dimensionality

reduction with auto-encoders for outlier detection, moreover compared it with linear and kernel component analysis. The comparison results shown those auto-encoders have a superior precision with a fewer complex computation. [17] Presented acoustic event detection with use of convolution neural networks and experimental end results has great detection rate. [18] Authors analyzed audio spectrum features by denoising auto-encoder with bidirectional LSTM networks. Here, authors use reconstruction error of input–output signals of auto encoder to identify novel events. [19] planned a one-class collective LSTM-RNN base model to notice outliers. Designed model train with regular data, and gain capability to anticipate input which was few time-steps in advance, same process continues for every step. Here observed prediction errors from certain number of time steps utilized to build a fresh model and threshold used to detect collective anomalies. [21] Proposed a novel auto-encoder variant, smooth auto encoder (SmAE) which learns robust discriminative feature representation. SmAE learns encoding of each sample to reconstruct it local neighbors.

3 Background Preliminaries of Cluster Head Selection Procedure

In recent times, most of the researchers found that data transmission among sensors exhausts valuable energy in comparison to the other tasks of the sensor. Reduction of unnecessary data communication can resolve the loss of energy and enhance the over all network lifetime [3, 6]. Clustering is one of the procedures to extend the network life span. In clustering protocol, geographically deployed adjacent nodes are structure into effective groups as clusters, among this one of the nodes selected as cluster head based on some concepts and remaining its adjacent nodes as cluster members. Clustering techniques has its advantages over other procedures, some of them are: reduce collision in the period of intra-cluster formation by its cluster members, reduces information updates such as inclusion and exclusion of nodes need to be updated simply by their cluster members, by rotation principle changes load balancing on cluster head. Cluster head works as a local coordinator for its adjacent members as aggregates all members' correlated data into a smaller set of information, forwards data to the base station [3, 14]. Many techniques such as swarm intelligence, neural networks, and distance-based metrics, artificial immune systems, fuzzy logic, and evolutionary algorithms have been proposed for Cluster head selection. Coming to WSNs, sensors installed in a compact way that directs to the excessive degree of spatial relationships between the sensed data in the adjacent nodes, this also increases the unnecessary data communication over the network. And it was also found that weather factors hold strong temporal correlations, hence, the inclusion of the prediction method on each sensor node could reduce the unnecessary data diffusion by utilizing the temporal association of sensed data. So, it's useful to implement a data-aware cluster technique based on the spatial relationship of data cloud reduces the transmission over the network and enhance the substantial energy savings in WSNs [3].

3.1 Procedure for Cluster Head Selection

1. In the beginning, every sensor has its residual energy as well as holds some sensed data from surrounding environments, after waiting for a particular time instance

higher energy sensor announce themselves as the Cluster Head (CH), this time instance is constant and decided by the time of network deployment. It is the maximum transmission time to announce messages among nodes.

2. If any of the nodes get a message before its announcement, it stops further announcements and considers it as the non-Cluster head. Stores all received messages in its database in the order of probability CH. If a non-cluster head node receives more than one message, it calculates the difference among two data points with help of point symmetry distance (PSD). In PSD Non-cluster head node takes its data points as a reference data point, if the distance is more than a fixed threshold value then it discards from its CH probable list and tries with another message. If the dissimilarity is below the threshold value it again evaluates the dissimilarity in cluster head residual energy with its residual energy. Finally, the non-cluster head node selects the CH with the highest differential factor in the CH list.
3. If the non-cluster head receives only one message then it declares the source of the message node as its cluster head.
4. Node joins into the CH through sending a message which contains sensor ID and residual energy differential factor.
5. CH receives the message and stores the sensor ID and definitional factor in its database.

4 Background Preliminaries of Auto Encoder Outlier Detection Method

An auto-encoder is deep learning based neural network which is skilled to make an effort to copy its input to its output. It has a hidden layer H which explains a code to symbol of the input [18–21]. Auto-encoder set-up contains two parts namely encoder and decoder: an encoder task $H = f(x)$ as well as a decoder that fabricates reconstruction $r = g(H)$. The pictorial representation of auto encoder shown in Fig. 1. But it not so useful if an auto-encoder implemented to be a successful model in simply learning copying input to output. If the intention is to copying with auto-encoder then it's not so useful. So, as an alternative to above mentioned description, auto-encoders designed in a specific way where they not capable to learn to copy completely, they are restricted in a specific way that allows to copy only something like, and to copy only input that look a lot like the

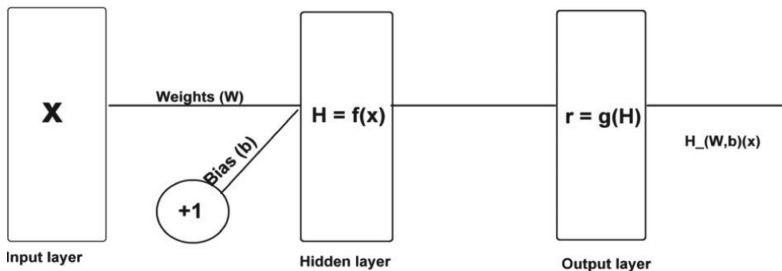


Fig. 1. General structure of auto-encoder

training data. Since the model is forced to prioritize which portions of the input must be copied, it often learns constructive properties of the information [22].

From the Diagram: Input takes D-dimensional input data that represents time series of sensor readings $x = x_1, x_1, x_1 \dots \dots x_d$. An output layers denoted with $\bar{x} = \bar{x}_1, \bar{x}_2, \bar{x}_3 \dots \bar{x}_D$, in auto-encoders, outcome has the similar dimension as the input which is useful in reconstructing the original input. So, by means of $y = x$ involuntarily achieve training samples. Correctly in reaction to the input value x hidden code representation H will be as $H(x) = f(W_1x + b_1)$. Here $f(\cdot)$ is a non-linear activation function; here we employed Gaussian based kernel function. In this Weights and Bias are the two hyper-parameters which represent W, b respectively. $W_1 \in R^{m \times n}$ Weight matrix and $b_1 \in R^m$ bias vector. The network output maps the hidden representation H reverse to a reconstruction $\bar{x} \in R^n$.

$$\bar{x} = f(W_2 H(x) + b_2)$$

Here $W_2 \in R^{m \times n}$ is weight matrix, $b_2 \in R^m$ is bias vector. Here final output \bar{x} is also expressed as $H(W, b)(x)$.

4.1 Denoising Auto-Encoder

The denoising auto-encoder forces the hidden layer to regain strong features and avoid it from simply learning individuality. In such arrangement the auto-encoder is skilled to renovate the input from a degraded report of it. Officially, the primary input x is tainted or degraded by means of additive isotropic Gaussian noise in order to gain: x' The corrupted input x' is then map, as with the fundamental auto-encoder, to a hidden symbol is $H(x') = f(W'x' + b')$, from this original data instance will be recon-structed as $\hat{x}' = f(W'x' + b')$.

Given a training sample $(X_1, X_2, X_3, X_4, \dots X_n)$ where each sample $X_i = (x_{1i}, x_{2i}, x_{3i} \dots \dots x_{di})$ here the intention of auto-encoder is to reduce the cost function.

$$J(W, b) = \frac{1}{T} \sum_{i=1}^T \left(\frac{1}{2} \|H_{W, b}(\bar{X}_i)\|^2 \right) + \frac{\lambda}{2} \sum_{l=1}^{L-1} \sum_{j=1}^{n^l} \sum_{i=1}^{n^{(l+1)}} \left(W_{ij}^{(l)} \right)^2$$

The first expression describes reconstruction error regard to novel inputs, and second expression is a regularization phrase to avoid over-fitting.

After formation of proper cluster head (CH), each non-cluster head sensor node individually computes denoising auto-encoder with Gaussian kernel for the reconstruction error of its n-th input–output sensed observations up to some specific time. Then after computed input and reconstructed error transformed to the prescribed cluster head. Here, cluster head calculate the mean and variance from the collected inputs from cluster member nodes and forward the updated parameter values to member nodes. With the updated parameter values each non cluster node again calculate the mean and variance and detect outliers when the $\gamma_n(s, d) = 1$.

Each sensor node S calculates the reconstruction error of n-th input–output of observation $(1, 2, 3 \dots N)$ of a particular time period on a day.

$$r_n(s, d) = x_n(s, d) - \bar{x}_n(s, d)$$

Each sensor transform the collected $r_n(s, d)$ to the appropriate cluster head.

The cluster head calculates the statistics of reconstruction errors mean $\text{Mean}(\mu_n)$ and Variance (σ^2) .

Cluster head forwards the calculated mean and variance to appropriate sensor nodes. Each sensor autonomously detects outliers by calculating $\gamma_n(s, d)$. Here 0 indicate normal and 1 indicate outlier.

$$\gamma_n(s,d) = \begin{cases} 0, & \text{if } |r_n(s,d) - \mu_n| \leq p\sigma_n \\ 1, & \text{otherwise} \end{cases}$$

5 Experimental Analysis

For the experimental results, we consider a data set congregate from WSN positioned at Intel Berkeley Research Laboratory (IBRL). The positioned WSNs contain 54 Mica2Dot sensor nodes and placed in the IBRL for 30 days nearly 720 h. Sensors collect data with five measurements voltage in volts, the temperature in degree Celsius, and network topology position for every 30-s gap, light in Lux and humidity ranging from 0 to 100%. Here, the data gathered with the TinyDB in-network query processing system that built on the Tiny-OS platform. Node 0 is the starting node; remaining nodes transmit data with several hops to node 0. The extreme nodes transform the sensed data with a maximum of 10 hops. During 720 h these 54 nodes composed about 2.3 million readings. Out of this 54 nodes we have taken a small subnet 1, 2, 3, 35 and 37 are forms a cluster network (CH1, S2, S3, S35, S37) where node 1 is cluster head and remaining nodes as non cluster head members. Here we used IBRL data set collected from four non cluster head members with every four hour likely 4, 8, 12, 16, 20 from 4–6 march 2004 with two features humidity and temperature. In implementation analysis we followed the experimental set up in [22], We employed k fold cross validation to compress the samples to half the size. For verification purpose we have collected two sensor nodes data from dissimilar days, results showed that reconstruction error mostly coincide with original data.

For the experiment on the proposed model we prepared a testing set because the original atmosphere data did not contain any labels as to which data is normal and outlier. So two labels normal and outlier are prepared and this data set holds nearly 5000 normal and 400 anomalous samples. We employed k fold cross validation to compress the samples to half the size. For verification purpose we have collected two sensor nodes data from dissimilar days and load hyper parameters, results showed that reconstruction error mostly coincide with original data Fig. 2. For detailed checking of the proposed model we analyze both false positive rate (FPR) and True positive rate (TPR), after various experiments we make mean 0.6 and variance 0.01, here variance less than mean measured as changes in atmosphere and variance greater than mean measured as outliers. From the Fig. 3 true positive rates are departing to jump down and the reason for this is, in training data we inserted more amount of outliers for evolution but when coming to random data

which is not affected as like trained data (Priority), most part of data less influence by the latest atmospheric alterations. Coming to false positive rate, priority data shows lower results than random atmosphere data, since auto-encoder learns enough information from the trained or priority data by periodically updating the hyper-parameters after each re-training which allows more scope to accept changes without raising false alarm signs.

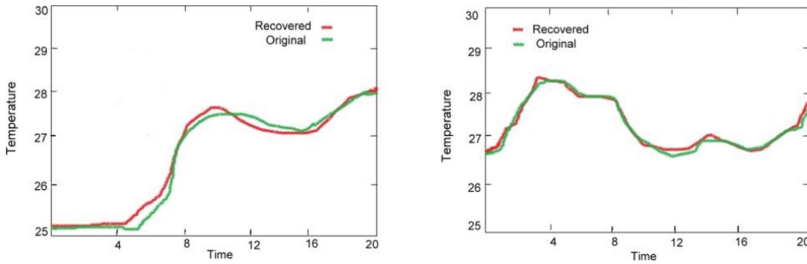


Fig. 2. Verification of proposed auto-encoder model with original and recovered data

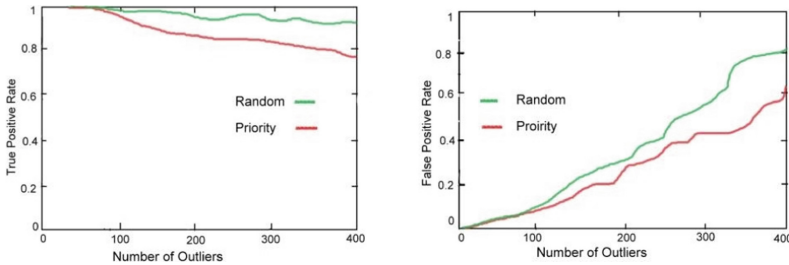


Fig. 3. Pictorial representation of model checking with random environment data

6 Conclusion

Automatic abnormality detection in various wireless sensor networks is tricky work. The sensed data detained by the sensors are influenced by node innate attributes as well as unexpected ecological differences. Here we proposed a cluster based denoising auto-encoder to detect outliers at sensor end as well as at the cluster end. Proposed technique reduces the communication overhead as well as the computational load of parameter updating done at the cluster head which has high energy resources; this will increase the life of the network.

References

1. Chander, B., Kumaravelan: Introduction to Wireless Sensor Networks, Soft Computing in Wireless Sensor Networks. CRC Press (2018)

2. Chander, B., Kumaravelan: A analysis of machine learning in WSN. *Int. J. Eng. Technol.* **7**(4.6), 185–192 (2018)
3. Chowdhury, S., Roy, A., Benslimane, A., Giri, C.: On semantic clustering and adaptive robust regression based energy-aware communication with true outliers detection in WSN. *Ad Hoc Netw.* **94** (2019)
4. Xie, M., Han, S.: Anomaly detection in WSN: a survey. *J. Comput. Appl.* **34**(4), 1302–1325 (2011)
5. O'Reilly, C., Gulhak: Anomaly detection in WSN in a non stationary environment. *IEEE Commun. Survey Tutor* **16**(3), 1413–1432 (2014)
6. Chander, B.: Clustering and Bayesian Networks. *Handbook of Research on Big Data Clustering and Machine Learning*, pp. 50–73. IGI Global (2020)
7. Chander, B.: Deep learning network: deep neural networks. In: *Neural Networks for Natural Language Processing*, pp. 1–30. IGI Global (2020)
8. Sheng, B., Li, Q., Qun: Outlier detection in sensor networks. In: *Proceedings of the 8th ACM International Symposium on Mobile Ad-Hoc Networking and Computing –MobiHoc*, pp. 212–219 (2017)
9. Dereszynski, E.W., Dietterich, T.G.: Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns. *ACM Trans. Sens. Netw. (TOSN)* **8**(1), 3 (2011)
10. Minh, N.H.: Outlier detection and correction for cooperative online learning. In: *IEEE Conference Proceedings* (2017)
11. Zhang, K., Shi, S., Gao, H., Li, J.: Unsupervised outlier detection in sensor networks using aggregation tree. In: *Proc. ADMA* (2007)
12. Branch, J.S.: In-network outlier detection in wireless sensor networks. In: *6th IEEE International Conference on Distributed Computing Systems (ICDCS 2006)*, vol. 34, pp. 51–61 (2006)
13. Abid, A., Masmoudi, A., Kachouri, A., Mahfoudhi, A.: Outlier detection in wireless sensor networks based on OPTICS method for events and errors identification. *Wirel. Pers. Commun.* (2017). <https://doi.org/10.1007/s11277-017-4583-7>
14. Ahmad, B., Jian, W., Ali, Z.A., Tanvir, S., Sadiq Ali Khan, M.: Hybrid anomaly detection by using clustering for wireless sensor network. *Wirel. Pers. Commun.* **106**, 1841–1853 (2018)
15. Feng, Q., Zhang, Y., Li, C., Dou, Z., Wang, J.: Anomaly detection of spectrum in wireless communication via deep auto-encoders. *J. Super Comput.* (2017)
16. Sakurada, M., Yairi, T.: Anomaly detection using auto-encoders with nonlinear dimensionality reduction. In: *Proceedings of the MLSDA, 2nd Workshop on Machine Learning for SensoryData Analysis*, pp. 4–15. ACM (2014)
17. Espi, M., Fujimoto, M., Kinoshita, K., Nakatani, T.: Exploiting spectro-temporal locality in deep learning based acoustic event detection. *EURASIP J Audio Speech Music Process* (1), 26 (2015)
18. Marchi, E., Vesperini, F., Eyben, F., Squartini, S., Schuller, B.: A novel approach for automatic acoustic novelty detection using a denoising auto-encoder with bidirectional lstm neural networks. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1996–2000. IEEE (2015)
19. Thi, N.N., Cao, V.L., Le-Khac, N.-A.: One-class Collective Anomaly Detection based on LSTM-RNNs
20. Abid, A., Kachouri, A., Mahfoudhi, A.: Outlier detection for wireless sensor networks using density-based clustering approach. *IET Wireless Sensor Systems* (2016)
21. Liang, K., Chang, H., Cui, Z., Shan, S., Chen, X.: Representation learning with smooth autoencoder. In: *Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS*, vol. 9004, pp. 72–86. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16808-1_6

22. Luo, T., Nagarajan, S.G.: Distributed anomaly detection using auto-encoder neural networks in WSN for IoT. IEEE Conferences (2016)
23. Trinh, V.V., Tran, K.P.: Anomaly detection in wireless sensor networks via support vector data description with mahalanobis kernels and discriminative adjustment. In: 2017 4th NAFOSTED Conference on Information and Computer Science (2017)



Malicious Webpage Classification

Kushagra Krishna^(✉), Jaytrilok Choudhary, and Dhirendra Pratap Singh

Maulana Azad National Institute of Technology, Bhopal, India
jaytrilok@manit.ac.in

Abstract. Attacks through the web pages containing malicious content have become an increasingly threat to the web security in the recent years. Thus, detection of the malicious URL is an important task to reduce the security threats. To detect malicious URL or web pages, there are several ways but the most traditional technique is through the Black List detection. The Black list contains the list of malicious web pages that are maintained so that user can be aware about the web pages before accessing any webpage. But, the problem with the black list is that it is not an effective method as malicious web pages change frequently, and also growing numbers of web pages that pose scalability issues. A part from black-list technique, various research techniques have been proposed that use machine learning technique and some use CNN (Convolution Neural Network) to classify web pages into category: malicious or benign. In the paper, a literature survey on classification of malicious web pages is presented that compares various machine techniques with parameter: precision, recall, and F1 score. This survey shows that the Machine learning techniques are better if the features used are textual but when there are images in web page, CNN performs better for the malware image classification.

Keywords: Malicious · Benign · EDCNN (Event De-noising CNN) · CNN (Convolution Neural Network) · SVM (Support Vector Machine) · RB-SVM · MDT (Markov detection tree)

1 Introduction

An advent of new technologies has increased the importance of WWW (World Wide Web) continuously. Various methods are used to attack on a web page such as XSS (Cross Site Scripting) that infuses a malicious code in the webpage using unexpected source. Other such types of attacks are through hacking, drive-by exploits, social engineering, phishing, denial-of-service, distributed denial-of-service, and many others. To prevent these types of attacks, many attempts have been made [1]. The entrance to a website is through URL only so if URL is detected as malicious before the entrance, it will prevent users from getting affected. Malicious URL detection [2–4] is a typical classification application scenario. These approaches are based on different types of features of web pages extracted and optimal feature selection.

In recent years, static and dynamic analysis is the two most common techniques for the detection of malicious web pages. The static analysis examines the executable file

without viewing original file. At first distinct static features are extracted from different web pages, then trains a classifier and it predicts whether web pages are benign or malicious. Now-a-days, due to increase in popularity of URL shortening services, malicious URL can be easily hidden behind a short URL and the user vulnerable to different types of attacks when access it. Sometimes, attackers complicate their codes in order to prevent themselves from getting detected from some signature based tools, so that their identity remains hidden and also helps in hiding the malware that can damage a system. To classify such web pages, dynamic analysis is required.

Several researchers have applied different machine learning techniques for Malicious URL detection. Using Machine Learning approaches, set of URLs are given as training data, and based on the statistical properties, the predictive model classifies a URL as malicious or benign. After getting data set, features such as URL, lexical, contextual and visual features are extracted from data. The feature selection describes about different features used in the malicious web detection process. Now-a-days while browsing sites, there are some malicious images also present on the site and for their feature are not needed the URL features, then the classification of images are needed using Neural network algorithms as neural network gives better performance than any machine learning algorithms. Several image processing techniques analyze malware images as gray-scale images. After malware binary file is visualized into digital image, the malware binary detection converts the model into multi-classifier model. In order to detect malware binaries, features can be extracted manually from malware images and then feed them into different classifiers such as SVM (support vector machine) or KNN (k-nearest neighbor's algorithm). To be more discriminative in extracting features, one can utilize CNN to automatically extract features and perform classification in an end-to-end fashion.

The rest of the paper is organized as follows: Sect. 2 includes a brief literature review along with the research gap associated with different classification model and related work in this area. Section 3 concludes the work and suggests future directions in this area of research.

2 Related Work

Different methods have been proposed for classification and detection of different web attacks through drive-by-download, malicious JavaScript attacks, cross-site scripting attacks, code injection attacks etc. The methods used by researchers are all feature based malicious web detection.

Kumar et al. [5] Classified webpage as malicious or benign using a multi-layer detection model and it is composed of total four layers. First layer is defined as the stratified filter layer which consists of Black and White list filter and Naïve Bayes filter. Second layer performs threshold training using Alpha-N-Bayes. Third layer consists of CART decision tree filter [6, 7] and the final layer consists of SVM filter which ultimately classifies page as benign or malicious. Features extracted are all the URL features. The features used by them are as follows:

- The domain names which are containing more than four consecutive numbers.
- Domain names having special characters.

- Top five domain name.
- Number of “.” in domain name.
- Total length of domain name.
- Length of longest domain name segment.
- Meaningful coefficient in primary domain names.

The limitation associated with this model is that the accuracy rate obtained by the Multi-layer filtering model is 79.55% which is not that much good and it can be increased by selecting more lexical features, contextual features, visual features and URL features using correlation based feature selection method.

Kazemian et al. [8] has done comparative analysis of supervised and unsupervised learning techniques. They have used three supervised machine learning techniques: KNN (K –nearest neighbor), SVM (Support Vector Machine) and Naïve Bayes Classifier; and two unsupervised machine learning algorithms: K-means and Affinity Propagation. Total 100,000 web pages were downloaded, feature were extracted and stored in form of vectors. These vectors were given as inputs to WAC (Web Application Classifier), and the proposed machine learning techniques were used to create a predictive model. The predictive models operates on feature vectors of the webpage that produces an output telling if the given webpage is safe or not. Safe webpages dataset has been gathered from Alexa and malicious one are collected primarily from Phish tank. The category of features extracted is:

- Semantic Features represented using vector space model
- URLs
- Page Links
- Visual features extracted using SURF (Speeded Up Robust Features) technique [9]

TF-IDF is calculated for each webpage to get the weights of semantic features. The combination of URL, semantic TF-IDF and SURF (Speeded up Robust features) gives highest accuracy. The supervised learning techniques: KNN, RBF SVM, LSVM and Naïve Bayes accuracy is calculated and compared. The supervised learning classification was evaluated using precision and recall. For analyzing the efficiencies and performance of supervised algorithm ROC (Receiver operating characteristic) curve and confusion matrix is used. The Linear Support vector machine and RBF support vector machine performance is better than the KNN and Naïve Bayes in supervised learning classification with accuracy of Linear SVM = 92.4 and RB SVM = 97.8.

Silhouette coefficient was used to calculate the performance of unsupervised learning techniques: K-means and Affinity Propagation. Affinity Propagation = 0.963 outperforms the K-means = 0.877 classification model. The accuracy of unsupervised models is not better than the supervised models. The limitation of this model is that instead of using machine learning techniques for visual features extraction, Deep learning could have been used to get better performance result.

Yue et al. [10] proposed a method for detecting malicious webpage that is basically of two types: static and dynamic detection. Static detection uses pattern matching, static code analysis, heuristic rules, and other static methods to detect the webpages code [11]. Dynamic detection is a method of detecting malicious web attacks such as honey client

based on behavior analysis [12]. The proposed technique detects malicious web pages and also identifies the specific threat types. As the forms of malicious web pages are various and unknown, the method used categorizes various types of malicious webpage. Four different categories of webpage datasets are collected: normal, phishing, spamming and malware. The phishing web pages data set is obtained from Phish Tank, spam webpages dataset has been taken from WEBSpam-UK2007 and the dataset for malware webpages is obtained from Malicious Web site Labs.

30 discriminative features were extracted in three fields including HTML, JavaScript and URL feature. KNN and “libsvm” model is used for training and classification. The Accuracy, True positive and False positive metrics have been used for evaluating detection performance. They proposed two prediction model: first model directly predicts the type of threat and the second model first classifies the web page as malicious or benign, and if malicious then further it classifies the threat type. Both models were used to do comparative analysis of KNN and SVM. The performance and accuracy of KNN was better than SVM in both model. Thus, KNN = 5 is used to implement binary classifier and KNN = 13 is used to implement multiple classifier total accuracy obtained is 88.1%. The limitation associated with this model as the value of KNN increases accuracy decreases and false positive show increased value, which means in multi-classification step the probability of incorrectly labelling of the webpage increases, and for performing multi-classification using KNN, higher KNN value is needed so KNN is not the best option.

Shibahara et al. [13] proposed a technique to detect Drive-by download attacks. Drive-by download is responsible for attacks that infect the hosts present on any network. For detecting Drive-by download attacks, they have proposed malicious URL sequence detection technique. Lists of all URLs that link to several other URLs are created and the malicious URL is detected using the EDCNN (Event de-noising CNN) technique. In EDCNN (Event De-noising CNN), the same combination of two malicious URLs features are extracted. The false alert generated by EDCNN is 47% less as compared to CNN (Convolution neural network) in detecting malware infection.

The methods for malicious URL detection are mostly focused on fetching the IP addresses corresponding to the domain [12, 14]. During training, URL sequences along with their labels are gathered, and the classifier is trained using the proxy logs. To achieve high detection performance, two categories of features are extracted:

- Historic domain-based features: It consists of two approaches. First approach is based on correspondence [14] and the second one is based on behavior approach [15, 16]
- Momentary URL-based features: several momentary based features of URL are suggested in [2, 12]. They selected four types of features: the length of a part of a URL, presence of a malicious or benign trace in a URL, information related to a domain extracted from a URL and information related to “geoiip”.

The URL sequences was classified using three distinct approaches: individual based approach, CNN based approach and the last approach based on EDCNN. The individual based approach classifies given sequence of URL as malicious even if at least any one of the URL in the sequence is getting classified as malicious. In CNN based approach,

the CNN model comprises of total eight layers out of which it has one input layer, two convolution layers, one max pooling layer, one SPP (Spatial Pyramid Pooling) layer, two fully connected layers, and the output layer. EDCNN reduces the negative effect produced due to benign URLs in the given malicious sequence by adding an allocation layer, enabling fast training and classification of URL. To evaluate the efficiency and performance of the three approaches and compare their detection performance, parameters such as: false alert, and calculation time is computed.

The data set for URL sequences are collected from: Malware domain list present in blacklist, Malware bytes, “hphosts” and popular websites lists from “Alexa.com”. On the basis of above parameters, EDCNN lowers the false alert as compared to individual approach and CNN. Also, the classification time of EDCNN = 0.07 is also less compared to individual approach and CNN (0.21, 0.09). Thus, EDCNN based approach performs classification in a reduced time as compared to individual-based approach and CNN. The limitation of the model is that they conducted classification without considering the content corresponding to its URLs. So, if in case an attacker removes the malicious content from a website and URLs of the website remain unchanged, then this technique will also classify the URL sequence as malicious even though if it is not.

Abdi et al. [17] proposed a method to classify Malicious URL on the basis of several techniques such as Deep learning techniques [18], machine learning techniques, dynamic attack detection method, and cross layer web detection approach [19]. The URL is classified using three different algorithms CNN, SVM and logistic regression; and then obtained results have been compared. The Dataset was taken from GitHub. The classification model has two layers: first layer uses blacklist method where URL is searched in whole blacklist and if the URL is found, search is stopped otherwise it will be passed on to the second layer of dynamic detection method.

Word2Vec and TF-IDF features are two different techniques used for weighing the features that have been used for detecting malicious URLs. Word2Vec is a model that contains two-layer neural network and is trained for reconstruction of linguistic context of word. It takes a large collection of text as an input and outputs a set of vector space: feature vectors for words in that corpus, it converts semantic text into numerical form that deep neural network models can understand. TF-IDF is a numerical statistic that specifies the importance of a word in a document.

$$\text{TF-IDF}(t, d, D) = \text{tf}(t, d) * \text{idf}(t, D) \quad (1)$$

The SVM learning algorithm classification is based on TF-IDF and is compared with the logistic regression technique and CNN. “Word2vec” feature function is used for weighing features, it converts the texts features into numerical value which can be easily given to CNN. The convolution neural network using Word2vec is similar SVM algorithm based on TF-IDF. 96% accuracy is achieved in detecting malicious URLs using this method. The limitation of this model is that classifying URL through CNN using its textual features, there is not much increase in the efficiency as CNN performs better on image features rather than on textual and semantic features. Instead of using TF-IDF to other better weighing methods could have been used.

The malicious webpages can be detected based on two approaches i.e. static, dynamic. Wang et al. [20] proposed a hybrid approach which has a combination of

static and dynamic analysis. In the proposed approach there are two stages: first stage uses static analysis and second stage uses dynamic analysis; and from going from first stage to second stage a threshold value is determined. The first stage performs static analysis based detection using the static features of webpages. The pre-threshold value determined in static analysis stage that helps in classifying the webpage as benign or malicious. The webpages, which obtained values are below threshold, are referred as unknown in first stage and then are given to the second stage. The second stage uses dynamic analysis, the webpage source code is executed in an emulated environment and if any one of the unknown instance includes malicious shell code, that instance is finally classified to be malicious.

Total 17 features are extracted from URL, Html document, and JavaScript using static method. As there are a lot of features, so in order to get the best possible set of features, correlation-based feature selection method is applied. It helps in selecting the most representative and accurate set of features giving best accuracy and precision among other representative set.

The web pages are classified in three classes: benign, malicious and unknown. The decision tree based algorithm establishes a classifier, predicting a probability distribution over all classes of threats and return the probability of a samples belonging to each class. The class having higher probability related to the sample will be the label of the web page. The ROC(Receiver Operating Characteristics) Curve is used in order to compare different types of features set and based on ROC curve, the URL features performed better than other category of features because of datasets were collected from specialized URL lists that contained evident malicious and benign features of URL.

The results of static, dynamic and hybrid analysis were compared using precision, recall and F-score; and hybrid analysis outperformed other two with Precision = 0.952, Recall = 0.882, F-score = 0.916. The limitation of this model is that the number of features extracted were less and also among supervised learning techniques, SVM shows better performance than others.

Kabanga et al. [21] converted malware images into gray scale image using Deep Learning's CNN showing improvement in the process of classification of malware images as it is easier to use an image as input to a model that uses Deep Learning's Convolution Neural Network rather than using URLs and extracting textual features [24]. The first one to visualize image malware as gray-scale image.

This technique used malware images as input to the classification model. Dataset used are taken from "Maling" Dataset and that used in [22]. The input images were first reshaped to a size of "128*128*1" and vectors of images are generated using the image library from "PIL" package of Python. The classification model is a three-layered deep Convolution Neural Network. The activation function used by them is ReLU (Rectified Linear Unit) as it proves to be faster than hyperbolic tan (tanh) or sigmoid function and also helps in solving the vanishing gradient problem that takes place in lower layer of the convolution neural network model used. Max pooling layer have been used instead of other layers. As they are doing multi-class classification for that purpose they have used Cross entropy function. The output layer consists of 25 neurons that correspond to the 25 different categories of malware present in the input datasets.

The accuracy obtained by the proposed model is about 98%. The “Microsoft Malware Classification Challenge” held in 2015 was won by the technique proposed by [23], and accuracy achieved was above 99%. The limitation of the proposed technique is that the technique used is easily open to different types of attacks and the result produced is somewhat erroneous. [24–26] have shown how a small change in the quality of image not visible to the eyes of users could lead to misclassification of images. Therefore, it can be concluded that use of images features alone for malware classification can be dangerous as a small mistake while extracting images from malware can produce incorrect results.

Liu et al. [27] proposed a MDT (Markov detection tree) scheme for identifying and classifying webpages either as malicious or benign. The decision tree approach has been used as the number of hyper-parameters used for tuning are less [28], and among all the methods that do not use deep neural network, the decision tree achieves best performance computed using information entropy. The information gain value of the web page helps in the selection of best possible set of features from the feature-attribute library of the detected web page. Information gain analyzes the change in information before and after classification.

The combined Markov decision process and decision tree is known as MDT, and the process of webpage detection can be made more efficient make through automatic decision with the help of forward and backward search operations. Multi-webpages detection can be done by simultaneously running both webpage search tree and decision tree in a consolidated way. In Markov Detection Tree each web page are represented through node and correspondingly the decision tree helps identify malicious webpages on cloud. If any one of the node is detected to be malicious then it can be prevented from harming any user. Decision tree classifier is a predictive model and the feature-attribute in them are represented through each node in the tree and value of each distinct features and attributes are represented through the branches of the tree, and all leaf node represents the final output, as there are more than one leaf node so there will be several distinct path from root to leaf and from all such path the path having optimal value will be taken into consideration. The framework deployed by them is operationalized in three phases:

- **Webpage preprocessing:** webpages are first collected from monitor servers stored on cloud platforms.
- **Attribute processing:** pairs of features and attribute are fetched from webpage samples. From each webpage attribute is being extracted.
- **Machine learning:** each node representing a webpage is classified as malicious or benign using detection tree algorithm.

Pearson correlation coefficient has been used for pre-detection. They have used six features to detect and analyze webpages and for feature selection they have used linear method to select attribute. In this method redundant attributes according to the number of occurrence of attribute are removed and only important attributes are kept. For determining the pair of feature and attribute that has to be retained, the frequency of each pair of feature and attribute is calculated and then inserted in feature column of the attribute and after that according to the probability value attribute is filtered. As each feature has a number of redundant attribute information gain ratio is used along

with an optimizing parameter which helps in reducing the volatility of data and helps in obtaining a more reasonable and precise information gain ratio. The results shows that AMD and MVPD algorithm has higher accuracy and efficiency in comparison to C4.5 as the missing values are filled more efficiently by AMD and MVPD than C4.5. The limitation of this model is that they have used a very small data set for training and testing. The limitation is that only 6 features has been used for classification which is very much less, as we know the more the features much better is classification result (Table 1).

Table 1. Comparative Analysis of recent techniques of malicious web page detection

S.No	Research Papers	Datasets	Features	Classifier Model	Accuracy
1	Malicious URL Detection Using Multi-Layer Filtering Model [5]	“ http://www.mwsl.org.cn/ ”; “ http://www.dir001.com/ ”	Seven URL features extracted (as discussed)	Multi-Layer filtering model	Accuracy = 79.55%
2	Comparison of Machine Learning Techniques for detecting Malicious Webpages [8]	1. Safe webpage – 2. Alexa (2013) 3. Malicious-Phishtank(2013)	Semantic, URL, Page links, Visual features (as discussed)	Supervised Learning-KNN,SVM, Naïve Bayes, Unsupervised Learning- K-Means, Affinity Propagation	Supervised Learning-98% Unsupervised Learning-0.963 (silhouette coefficient)
3	Fine-Grained Mining and Classification of Malicious Web Pages [10]	1. Spam webpage- WEBSpAM-UK2007 2. Phishing-PhishTank 3. Normal-Yahoo	30 features from HTML, JavaScript and URL features. (as discussed)	KNN = 5 (Binary Classifier) KNN = 13 (Multiple Classifier)	Accuracy = 88.1%
4	Malicious URL Sequence Detection using Event De-noising Convolution Neural Network [13]	1. “ http://www.malwaredomainlist.com/ ” 2. “ hphosts, https://hosts-file.net/?s=Download ” 3. “ http://www.alexa.com/topsites/ ”	17 RHDN features 4 URL based feature	Event De-noising Convolution Neural Network	Accuracy = 68.09%

(continued)

Table 1. (continued)

S.No	Research Papers	Datasets	Features	Classifier Model	Accuracy
5	Malicious URL Detection Using Convolution Neural Network [17]	Github data set	Images features	CNN	Accuracy = 96%
6	Detection of Malicious webpages Based on Hybrid Analysis [20]	Alexa websites, Malware Domain List: " http://www.malwaredomai-list.com/mdl.php "	Features of URL, HTML document, and JavaScript	Combining both Static and dynamic analysis	Accuracy = 91.3%
7	Malware Image Classification using CNN [21]	9458 grayscale images of 25 malware families from Maling Dataset	Image features such as edges, corners, ridges etc.	CNN Images are reshaped into a size of 128*128*1	Accuracy = 98%
8	A Markov Detection Tree-Based Centralized Scheme to automatically identify Malicious Webpages on Cloud Platforms [27]	61 malicious 80 benign webpages	JavaScript based 6 groups of features extracted: Num, fun, str, gett, sett, acti	Markov Decision Process + Decision Tree = Markov Detection Tree	Accuracy = 91%

3 Research Gap

The most of the papers discussed above while classifying the URL link as benign or malicious have used very less textual and URL features. When classifying the image of a webpage as malicious or benign they have used the URL link of webpage instead of using the image as input and extracting features related to images.

4 Conclusion

The earlier URL classification methods [5, 8, 10, and 22] uses only lexical features. The URL features have been extracted for classifying images also. They have used the URL link of the image to extract the textual and semantic features related to the link. Now-a-days with the help of CNN [17, 21], image classification can be done and even Deep CNN is helpful in identifying images as malware or not. They have used CNN

for classifying malware image, but Deep CNN provides better result than CNN. As a webpage contain both images and texts, it will be difficult using separate techniques for classifying both images and texts. The survey presented in the paper, discussed either texts or images classification techniques but not both at the same time. Thus, a CNN model can be developed for classifying images and texts both at the same time.

References

1. Di Lucca, G.A., Fasolino, A.R.: Testing web-based applications: the state of the art and future trends. *Inf. Softw. Technol.* **48**(12), 1172–1186 (2006)
2. Ma, J., Saul, L.K., Savage, S., Voelkar, G.M.: Learning to detect malicious URLs. *ACM Trans. Intell. Syst. Technol.* **2**(3), 1–24 (2011)
3. Thomas, K., Grier, C., Ma, J., Paxson, V.: Design and evaluation of a real-time URL spam filtering service. In: *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, pp. 447–462, May 2011
4. Whalen, S., Boggs, N., Stolfo, S.J.: Model aggregation for distributed content anomaly detection. In: *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop -AISec 2014*, pp. 61–71, ACM Press, New York, USA (2014)
5. Kumar, R., Zhang, X., Ahmad Tariq, H., Khan, R.U.: Malicious URL detection using multi-layer filtering model. In: *International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, December 2017 <https://doi.org/10.1109/ICCWAMTIP.2017.8301457>
6. Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian cart model search. *J. Am. Stat. Assoc.* **93**(443), 935–948 (1998)
7. Steinberg, D., Colla, P.: CART: classification and regression trees. In: *The Top Ten Algorithms in Data Mining*, pp. 179–201 (2009)
8. Kazemian, H.B., Ahmed, S.: Comparisons of machine learning techniques for detecting malicious webpages. *Expert Syst. Appl.* **42**(3), 1166–1177 (2015)
9. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* **18**(11), 613–620 (1975)
10. Yue, T., Sun, J., Chen, H.: Fine-grained mining and classification of malicious web pages. In: *2013 Fourth International Conference on Digital Manufacturing & Automation* (2013)
11. Choi, H., Zhu, B.B., Lee, F.: Detecting malicious web links and identifying their attack types. In: *2011 WebApps 2011 Proceedings of the 2nd USENIX conference on Web application development*. USENIX Association, Berkeley, CA, USA, p. 11 (2011)
12. Canali, D., Cova, M., Prohiler, G.V.: A fast filter for the large-scale detection of malicious web pages. In: *2011 WWW 2011 Proceedings of the 20th international conference on World Wide Web*. ACM, New York, NY, USA, pp. 197–206 (2011)
13. Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., Ohsita, Y., Murata, M.: Malicious URL sequence detection using event de-noising convolutional neural network. In: *ICC 2017 IEEE International Conference on Communications*, pp. 1–7 (2017)
14. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for DNS. In: *2010 Proceedings of the 19th USENIX Security Symposium*, pp. 273–290 (2010)
15. Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou II, N., Dagon, D.: Detecting malware domains at the upper DNS hierarchy. In: *2011 Proceedings of the 20th USENIX Security Symposium*, pp. 1–16 (2011)
16. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: Exposure: finding malicious domains using passive dns analysis. In: *Proceedings of the 18th Annual Network and Distributed System Security Symposium*, San Diego, California, USA (2011)

17. Abdi, F.D., Wenjuan, L.: Malicious URL detection using convolutional neural network. *Int. J. Comput. Sci. Eng. Inf. Technol.* **7**(6), 01–08 (2017)
18. Wang, Cai, W.D., Wei, P.C.: A deep learning approach for detecting malicious JavaScript code: using a deep learning approach to detect JavaScript-based attacks. *Secur. Commun. Netw.* **9**(11), 1520–1534 (2016)
19. Xu, L., Zhan, Z., Xu, S., Ye, K.: Cross-layer detection of malicious websites. In: 2013 Proceedings of the third ACM conference on Data and application security and privacy. ACM, pp. 141–152 (2013)
20. Wang, R., Zhu, Y., Tan, J., Zhou, B.: Detection of malicious web pages based on hybrid analysis. *J. Inf. Sec. Appl.* **35**, 68–74 (2017)
21. Kabanga, E.K., Kim, C.H.: Malware images classification using convolutional neural network. *J. Comput. Commun.* **06**(01), 153–158 (2018)
22. Nataraj, L., Karthikeyan, S., Jacobs, G., Manjunath, B.S.: Malware Images
23. Wang, X., Liu, J., Chen, X.: First place team: say no to over fitting. In: Winner of Microsoft Malware Classification Challenge (BIG 2015) (2015)
24. Nguyen, A., Yosinki, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: IEEE, Computer Vision and Pattern Recognition (CVPR) (2015)
25. Papernot, N., McDaniel, P., Goodfellow, I.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communication Security, pp. 506–519 (2017)
26. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 2015 International Conference on Learning Representations (2015)
27. Liu, J., Xu, M., Wang, X., Shen, S., Li, M.: A markov detection tree-based centralized scheme to automatically identify malicious webpages on cloud platforms. *IEEE Access* **6**, 74025–74038 (2018)
28. Zhang, Y.-L., et al.: Distributed deep forest and its application to automatic detection of cash-out fraud. *ACM Trans. Intell. Syst. Technol.* **55** (2019)



Ontology-Based Modeling of Cloud Application Using Security Patterns

Ashish Kumar Dwivedi¹(✉), Shashank Mouli Satapathy²,
and Aakanksha Sharaff³

¹ Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering(A), Visakhapatnam 530048, India

² Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

³ Department of Computer Science and Engineering,
National Institute of Technology, Raipur, Raipur, India
`asharaff.cs@nitrr.ac.in`

Abstract. The acceptance of various computing aspects of cloud-based systems gets hampered by the evolution of various security threats. Security and privacy issues are considered as the primary challenges to an adoption of cloud computing. In order to develop a secure cloud, there is a need of proper analysis of security threats and their associated detection as well as prevention techniques. In this study, an attempt has been made to offer an ontology-based analysis and design approach for the cloud security. The security provision of cloud systems is accomplished by using security patterns, which are often specified by using UML (Unified Modeling Language) diagrams. In order to specify cloud pattern notations, a semantic modeling approach i.e., Web Ontology Language (OWL) has been considered. Description Logic (DL) has been considered for analyzing security requirements, which is supported by the OWL editor i.e., Protégé.

Keywords: Cloud security · Ontologies · Security patterns · UML

1 Introduction

Cloud computing has the characteristics of providing by a good number of services, such as network access, resource pooling, rapid elasticity etc. According to the reference architecture of cloud computing, it comprises five actors such as cloud consumer, cloud auditor, cloud provider, cloud broker, and cloud carrier. Cloud provider has the task to support a number of services, such as service deployment, service orchestrations, security management, etc. These services may be influenced by a number of security threats, which affect cloud environments such as traffic handling, virtualization, cloud data processing, access

A. K. Dwivedi — Internet of Things and Connected Technologies.

control, etc. The cloud security threats can be overcome to a certain degree by applying suitable security mechanism. In this study, a composition of security patterns is considered for the use-case i.e., migration of virtual machine in the presence of man-in-the-middle attack [1]. In this scenario Fernandez et al. [1] have presented UML-based reference architecture which is semiformal in nature that leads to ambiguities as well as inconsistencies.

In this study, patterns-based solution [1] is extended by applying some other desirable security patterns and further formalized by using ontology-based approach. The security patterns are applied for the problem occurring in a particular context, when a cloud administrator requests virtual machine migration process from source machine to the destination virtual machine monitor (VMM). In this context, a problem may occur as the attacker that listens to the network during the transmission of virtual machine. For this problem, the composition of security patterns are applied, which protect the network traffic from the security threats. The structural aspect of cloud security patterns for the migration of virtual machine is specified with the help of UML diagram which is further formalized by using Web Ontology Language (OWL) [2]. In order to specify cloud security requirements, Description Logic has been being applied.

Software design patterns are defined as a set of design decisions which are used to solve the problems, which are recurring in nature [3,4]. Security design patterns are proven solution for recurring security problem, occurring in a particular context [5]. A pattern is often described using a pattern template that helps to understand the problem and expressing the solution for the problem. The modeling of a pattern-based software is often carried out by using informal (natural languages) and semi-formal (UML) notations, which lead to ambiguities and inconsistencies. In this study, an ontology-based security framework is presented for specifying cloud security patterns. Ontology-based modeling enables system information in such a way where the information are termed as *things* [6].

2 Related Work

Various literatures are available for the modeling of cloud security, which require formal representation of their notations. In this study, some of the related work are presented [7,8].

Mozzaquatro et al. [9] presented an ontology oriented cyber security framework which performed the knowledge reasoning for the Internet of Things. Authors have claimed that their model helps to analyze design time for security services and run time for the monitoring of threats and vulnerabilities. Dwivedi [10] has analyzed web service secure conversation pattern by using the idea of ontology. Author has presented an ontology which shows semantic description of the conversation pattern. Auxilia and Raja [11] have presented semantic context aware authorization model which controls data access, based on context information and offers semantic technologies for the policy as well as context specification. Authors have presented a set of rules to handle dynamic context changes and its adaptation in policies.

Wen and Katt [12] have proposed an ontology-based methodology for the analysis of security knowledge. They have considered context based analysis for the retrieval of security knowledge. Authors have presented their security ontology by using design concepts and the process of evaluation. Mouliswaran et al. [13] have presented inter-domain access permission of multiple domain environments using ontology. They have presented a set of rules to map the matrix of access permission into the ontology of inter-domain access control. Souag et al. [14] have presented a method that exploits security ontologies and domain ontologies dynamically by using various production rules.

It is observed that most of the existing approaches are not based on security patterns which provide reusable solutions for the reoccurring problem in a particular context. The existing modeling approaches are not based on meta-modeling concepts which help to rigorously analyze the target system. Some of the existing approaches are not even specially used for cloud security domain. In this study, a semantic approach is being considered for providing unambiguous and inconsistent specification of cloud security patterns.

3 Proposed Work

Security aspects in cloud computing are categorized into *security standards, network, access control, cloud infrastructure, and data*. These aspects are further associated with different issues, such as lack of standards, lack of auditing, lack of SLA (Service Level Agreement), lack of network firewalls, IP vulnerabilities, lack of configuration security, lack of QoS, lack of data privacy, etc., which result in the large amount of information. A large number of resources came into existence, because the amount of information growing day-by-day. These resources contains data by using various storage techniques. But the limitations in availability of the resources creates problem to meet the amount of huge information, which are resolved by applying the idea of Semantic Web. Semantic Web offers data that is shared and also reused by an application domain. The components of semantic web such as Resource Description Framework (RDF) and the Web Ontology Language (OWL) provides an integrated framework where information can easily be organized, published, and retrieved. It is a family of knowledge representation language which is characterized by formal semantics and RDF/XML. In this study, an ontology-based security modeling framework is presented for checking consistency, concept satisfiability, instance classification, and refinement of security patterns and their composition.

3.1 Ontology-Based Cloud Security Modeling Framework

Ontology-based cloud security demonstrating structure is an augmentation of Model-Based Security Framework that is based on the idea of security patterns. The proposed methodology depends on the metamodel which empowers examination and plan of rules, imperatives, models, and speculations relevant

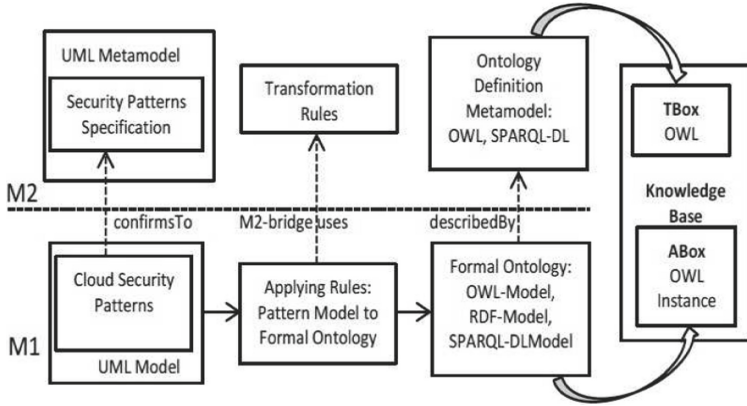


Fig. 1. Framework for the modeling of cloud security using ontology

for displaying a predefined class of issues. The demonstrating structure is displayed as far as Platform Independent Model (PIM) and Platform Specific Model (PSM) [15]. This study includes only two layers i.e., PIM and PSM for bridging the gap between semiformal notation (UML) with the formal notation (OWL) as shown in Fig. 1. PIM settle framework useful prerequisites regarding issue space while PSM offers an answer model that resolves both utilitarian and non-useful necessities of a framework. In the proposed study, static part of cloud security configuration is analyzed by using A-Box ontology. The dynamic part of cloud security configuration model is analyzed by DL-Safe principles. For the confirmation procedure a mechanized thinking can be performed by using a reasoner i.e., Pellet [16] which is supported by Protégé [17].

3.2 Cloud Security Patterns

In the presented approach, security patterns are applied for the usecase i.e., securing virtual machine migration process. The structural aspect of the selected security patterns is presented in Fig. 2. In this scenario, a cloud administrator tries to migrate virtual machine from one server to another. A cloud portal may be used to interact with the cloud services such as SaaS (Software as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service), support services, and infrastructure. Infrastructure includes a cluster which is a collection of physical entities such as hardware components (storage devices, network devices, servers) and a virtual machine monitor. In this structural aspect, three security patterns such as Check point, Authenticator, and Secure Proxy have been taken into consideration for providing protection from malicious user i.e., Attacker. When Cloud Administrator makes a request to Cloud Portal for VM migration, it forward request to Check Point for user authentication. Check Point may call Authenticator for request verification. If request is authenticated, it forward the request to Cloud Controller. Later the requested id is forwarded

to Cluster Controller then to Node Controller, and then to virtual machine monitor. VMM helps to manage various virtual machines and support remote access to different hardware devices. In this scenario, the attacker often monitors network to intercept the virtual machine migration process by modifying some of the essential information. The dynamic aspect of cloud security patterns in the form of sequence diagram is presented in Fig. 3.

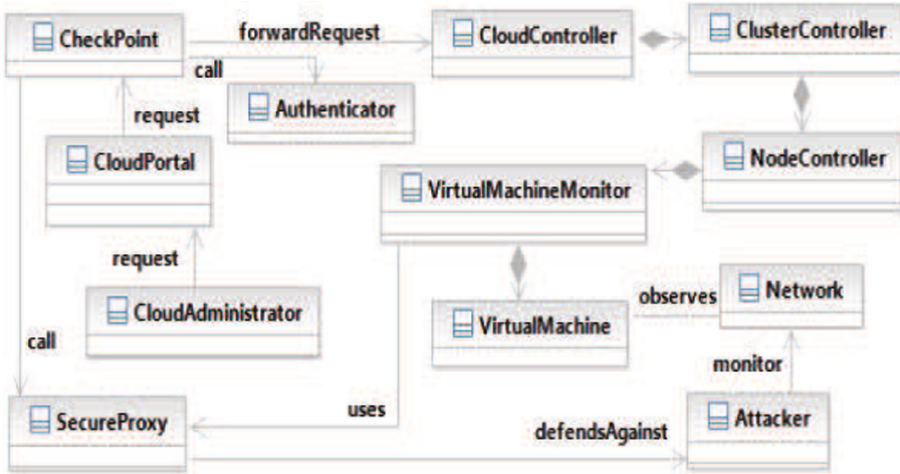


Fig. 2. Structural aspect of cloud security patterns

The composition of cloud security patterns are described by using standard pattern template elements [4]. In this study four template elements such as context, problem, solution, and consequences have been considered for representing patterns composition. In this scenario, context is a situation when a cloud user or administrator wants to migrate VMs. In this context, problem may occur in the form of Man-in-the-Middle attack, where an attacker wants to access or modify essential information during VM migration process. For this problem, solution may be provided by applying suitable security patterns. The usecase i.e., migrateVM requires security provisions which defend against the attacker. In this aspect, Check Point and Secure Proxy are used for providing request authentication and secure virtual machine migration respectively as shown in Fig. 3. The consequences may occur in the form of insecure virtual machine migration which creates problem for other processes e.g., stop the migration. Another consequence may create by the attacker that modifies crucial information during VM migration process.

The behavioral aspect of cloud security patterns is presented in Fig. 3, where Cloud Administrator’s request is validated by using Check Point pattern that contains two participants such as Authenticator and Secure Proxy as security patterns. Check Point calls Authenticator for request validation. The validated

request is forwarded to VMM through Cloud Controller, Cluster Controller, and Node Controller. When request is forwarded from Cloud Controller to VMM an Attacker can listen to the network and forward his request to VMM. This problem is solved in this study by applying Secure Proxy pattern, which aims to validate the request and remove the chance of Man-in-the-Middle attack.

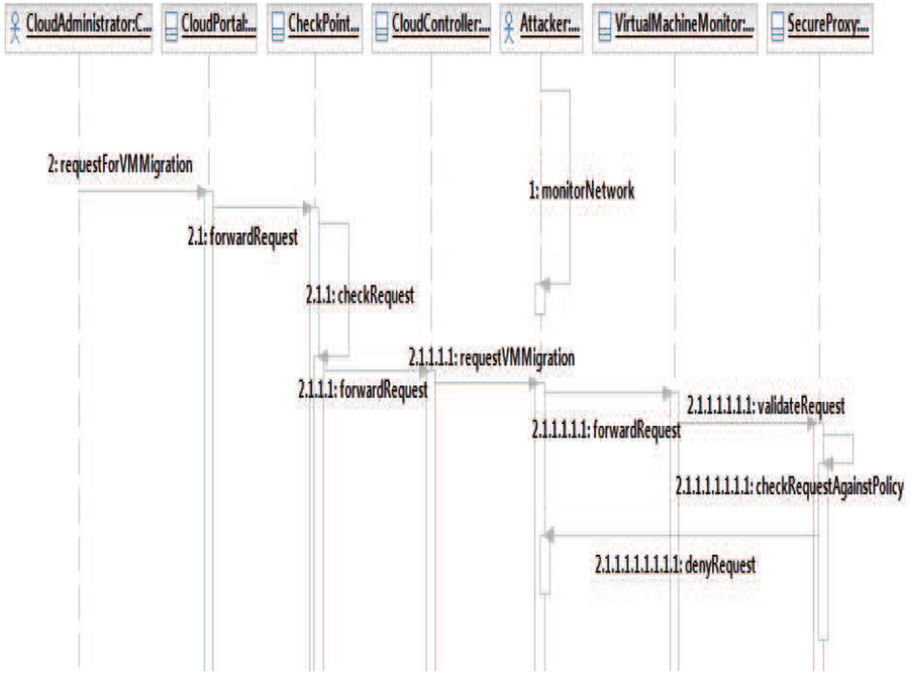


Fig. 3. Dynamic nature of cloud based security patterns

3.3 Ontology for Cloud Security Patterns

The proposed ontology-based cloud security modeling framework offers conceptualization for protecting virtual machine migration process as well as other similar types of processes, because patterns provide solution for recurring design problems. The presented ontology helps to understand the high-level abstraction for the cloud security environment. It also supports the concept of name spaces for specifying consistent data space. Figure 4 presents the classes in ontology and their semantic relationships. In this study, all assertions and properties are not presented in order to preserve the simplicity. The proposed ontology presents classes and subclasses such as CloudSecurityPatterns, Cloud, Attacker, CheckPoint, Authenticator, CloudServices, CloudCluster, etc. Cloud ontology also presents relationships such as isSubPatternOf, isSubClassOf, isAbstract, etc. For example CheckPoint isSubPatternOf CloudSecurityPatterns. This ontology

presents a number of object properties and data properties. Object property shows the relationship between instances of more than one classes. Here a number of object properties are represented such as *isSubPatternOf*, *isParticipantOf*, *controls*, *defends*, *exploits* etc. Data properties of OWL associate instances of a class to RDF literals or XML schema datatypes. The selected data properties are *sharedState*, *unsharedState*, *isElement* etc. The selected cloud security patterns are specified by using description logic which is shown in Table 1.

Description logic (DL) is used to represent knowledge in a formal notation, which analyzes various domains by using concepts and relationships. The concepts and relationships are denoted as classes and relations respectively. In Table 1, first axiom expresses that *CheckPoint*, *Authenticator*, and *SecureProxy* are subclasses of *CloudSecurityPattern*. Second axiom denotes the semantic relationship between OWL classes. For example, *Check Point* is associated with *Authenticator* and *SecureProxy* classes by using call object property where *Check Point* is a domain and *Authenticator* is a range of the call property. Axiom three to six denotes the aggregator relationship. For instance *ClusterController* is a containing class and *NodeController* is a aggregate class. Twelfth articulation demonstrates that all people are subset of all *hasPermission* and all *allowAccess* properties. Articulations thirteenth and fourteenth indicate for precisely one access and precisely one consent individually. Fifteenth articulation says that *Authorized-Consumer* and *Attacker* can't be a similar client. Seventeenth articulation determines about *allowAccess* activity executed by *CheckPoint*.

In order to provide semantic description of patterns, ontologies can be represented as the Terminology-Box (T-Box) and the Assertion-Box (A-Box) as shown in Fig. 1. T-Box helps to map metamodels at M2 level, whereas A-Box maps models at M1 level. Formalization of UML-based cloud security patterns can be done by applying various constraints, such as *isSubPatternOf*, *isSubClassOf*, *uses*, etc. These constraints provide semantic relationship between OWL classes for defining patterns. The OWL-based semantic definitions are based on set of rules which are processed by OWL reasoner i.e., Pellet. The proposed demonstrating structure is developed based on change rules, which incorporate linguistic structure, semantics, and example requirements change.

1. Each pattern participant such as instance of classes is defined by using variable systems, which denotes the name of pattern's participants.
2. A number of other predicates are being used as prefixes to semantically define cloud pattern's notations. For example, *isClass* defines the ontology of a class.
3. Similarly, relationships among the pattern participants can be defined by using a number of predicates *isSubClassOf*, *isSubPatternOf*, *isAbstract*, etc.
4. The constraints of a design pattern can be changed into OWL documentation by using design semantics in UML, which are indicated as UML class. A class includes techniques, qualities, and relationship. For instance, strategy can be spoken to as *isMethodOf* (*Migrate isMethodOf VirtualMachineMonitor*).

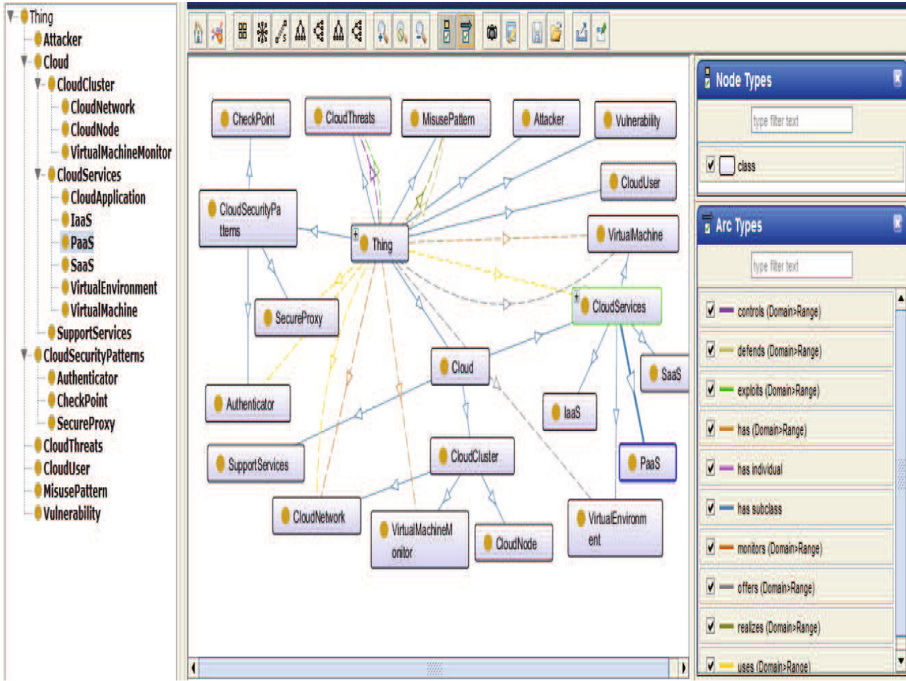


Fig. 4. Ontology for cloud security patterns

Table 1. Description logic specification for cloud security design patterns

S.No.	DL Notation
1.	$(\text{CheckPoint} \sqcap \text{Authenticator} \sqcap \text{SecureProxy}) \sqsubseteq \text{CloudSecurityPatterns}$
2.	$\text{CheckPoint} \equiv \exists \text{call} . (\text{Authenticator} \sqcap \text{SecureProxy})$
3.	$((1 : \text{CloudController}) \sqcap (2 : \text{ClusterController}))$
4.	$((1 : \text{ClusterController}) \sqcap (2 : \text{NodeController}))$
5.	$((1 : \text{NodeController}) \sqcap (2 : \text{VirtualMachineMonitor}))$
6.	$((1 : \text{VirtualMachineMonitor}) \sqcap (2 : \text{VirtualMachine}))$
7.	$\text{VirtualMachineMonitor} \equiv \exists \text{uses} . \text{SecureProxy}$
8.	$\text{Attacker} \equiv \exists \text{monitor} . \text{Network}$
9.	$\text{CloudAdministrator} \equiv \exists \text{request} . \text{CloudPortal}$
10.	$\text{CloudPortal} \equiv \exists \text{request} . \text{CheckPoint}$
11.	$\text{SecureProxy} \equiv \exists \text{defendsAgainst} . \text{Attacker}$
12.	$(\top \sqsubseteq \forall \text{hasPermission} . \text{Permission} \sqcap \text{allowAccess} . \text{Policy})$
13.	$(\text{Policy} \sqsubseteq (= 1 \text{ allowAccess} . \top))$
14.	$(\text{Permission} \sqsubseteq (= 1 \text{ hasPermission} . \top))$
15.	$(\text{AuthorizedConsumer} \sqcap \text{Attacker} \equiv \perp)$
16.	$\exists \text{cp} . \top \sqsubseteq \forall \text{cp} . \text{CheckPoint}$
17.	$\exists \text{operation} . \text{allowAccess} \sqsubseteq \forall \text{cp} . \text{CheckPoint}$

As indicated by change runs, an objective model can be produced from the source model. Right now security demonstrating structure, source model and target model affirms to comparing source metamodel and target metamodel as appeared in Fig. 1.

3.4 Evaluation

While the plan design philosophy takes after the traditional UML meta-model that contains a few highlights which add extra expressiveness expected to speak to configuration designs. In Adapter design pattern, *Adapter* adapts *Adaptee* to *Target* by committing to a concrete adapter; yet Adapter won't work when a class and every one of its subclasses are adjusted. This type of consequences may be specified by using semantic notation. The aforementioned notations may become helpful for designers and developers. They can utilized the actual semantics of classes, objects, and their relationships. OWL is used to model the definition of cloud security patterns and its consequences. The composition of patterns can be analyzed by using description logic which provides the availability of practical reasoning.

The performance evaluation is carried out by considering an OWL reasoner i.e., Pellet. For the experiment process, Windows 7 with Intel i3 processor and 2 GB RAM have been considered. The experiment is performed for the number of classes and measured time in milliseconds which is shown in Table 2.

Table 2. Performance evaluation using Pellet

No. of Classes	2	4	6	8
Pellet	133.2	296.8	380.5	680.4

4 Conclusion

In the presented approach, an ontology oriented cloud security framework is proposed by analyzing and designing of cloud-based security for different applications. This approach helps to assist the reuse and extensible modeling of complex and distributed systems. Formal analysis of cloud security is done by using web ontology language. Ontology-based modeling techniques help for representing the aspects, such as reusability, extendibility, modifiability, refinement, online publishing of a particular domain. The motivation behind using the security patterns is to reuse the solution for similar types of problem occurring recurrently in a particular context, which can be mapped into ontology. The property specification of a usecase i.e., migration of virtual machine is performed by using description logic.

5 Threats to Validity and Future Work

The automated tool support might be helpful for mapping semiformal context into semantic formal notations. The proposed work may be extended by developing a tool that transform graphical representation into formal representation.

References

1. Fernandez, E.B., Monge, R., Hashizume, K.: Building a security reference architecture for cloud systems. *Requirements Eng.* **21**, 1–25 (2015)
2. Dean, M., Schreiber, G.: Web ontology language (2004). <http://www.w3.org/TR/owl-ref/>
3. Dwivedi, A.K., Tirkey, A., Rath, S.K.: Applying learning-based methods for recognizing design patterns. *Innovations Syst. Softw. Eng.* **15**, 87–100 (2019)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston (1995)
5. Dwivedi, A.K., Tirkey, A., Rath, S.K.: Software design pattern mining using classification-based techniques. *Front. Comput. Sci.* **12**, 908–922 (2018)
6. Tao, M., Zuo, J., Liu, Z., Castiglione, A.: Multi-layer cloud architectural model and ontology-based security service framework for IOT-based smart homes. *Future Gener. Comput. Syst.* **78**, 1040–1051 (2018)
7. Liu, H., Ning, H., Mu, Q., Zheng, Y., Zheng, J., Yang, L.T., Huang, R., Ma, J.: A review of the smart world. *Future Gener. Comput. Syst.* **96**, 678–691 (2019)
8. Dwivedi, A.K., Tirkey, A., Rath, S.K.: An ontology based approach for formal modeling of structural design patterns. In: 2016 Ninth International Conference on Contemporary Computing (IC3), pp. 1–6. IEEE (2016)
9. Mozzaquatro, B.A., Agostinho, C., Martins, J., Jardim-Goncalves, R.: An ontology-based cybersecurity framework for the internet of things. *Sensors* **18**, 3053 (2018)
10. Dwivedi, A.K.: Ontology-based modelling of extended web service secure conversation pattern. *Int. J. Metadata Semant. Ontol.* **13**, 285–299 (2019)
11. Auxilia, M., Raja, M.: Semantic context aware authorization for securing data in cloud. *Adv. Nat. Appl. Sci.* **9**, 683–689 (2015)
12. Wen, S.F., Katt, B.: Managing software security knowledge in context: an ontology based approach. *Inf. Multidiscip. Digit. Publ. Inst.* **10**, 216 (2019)
13. Mouliswaran, S.C., Kumar, C.A., Chandrasekar, C.: Inter-domain role based access control using ontology. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2027–2032. IEEE (2015)
14. Souag, A., Salinesi, C., Wattiau, I., Mouratidis, H.: Using security and domain ontologies for security requirements analysis. In: 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 101–107. IEEE (2013)
15. Dwivedi, A.K., Satapathy, S.M.: Ontology-based modeling of IoT design patterns. *J. Inf. Knowl. Manag.* **20**, 1–32 (2020)
16. Clark & Parsia, L.: Pellet (2011). <http://semanticweb.org/wiki/Pellet>
17. Protégé: Description is available from (2010). <http://protege.stanford.edu/products.php#desktop-protege>



Deep-Learning Based Mobile-Traffic Forecasting for Resource Utilization in 5G Network Slicing

Rohit Kumar Gupta, Amit Ranjan^(✉), Md Ashraf Moid, and Rajiv Misra

Department of Computer Science and Engineering, Indian Institute of Technology
Patna, Patna, India
rajivm@iitp.ac.in

Abstract. Network slicing is the key technology in 5G wireless communication, which aims to provide services based on latency, availability, reliability, throughput and more. With the rapid development of mobile networks and new networking applications, it is turning out to be more difficult to meet the Quality of Services (QoS) under the current mobile traffic and mobile-network architecture. Mobile Traffic forecasting is one of the domains that can benefit the mobile companies in optimizing their resources. In this paper, we consider a dataset with Internet usage patterns by users over a period of six days. Based on past time-steps trends we tries to predict the current network slice that would be classified into streaming, messaging, searching, and cloud classes. We compared the four deep learning models namely MLP, Attention-based Encoder Decoder, GRU and LSTM and we evaluated these models on recall, precision and f1 score performance matrices. We found that MLP, Encoder-Decoder models performed average for mobile-traffic forecasting while the GRU, LSTM performs well and out of them LSTM obtained best result.

Keywords: Deep learning · Network slicing · Latency · Availability · Throughput · Reliability · QoS · Mobile traffic · GRU · LSTM

1 Introduction

The quick advancement of 5G systems and IoT innovation is setting off a flood in utilization of cellular traffic network worldwide. The most recent assessment demonstrate that 77.5 exabytes of versatile information will be devoured every month by 2022, which represents 71% of overall IP traffic [11]. The flow of cellular traffic is causing more use of radio spectrum and enormous amount of energy utilization, which requires productive network design and system operations to meet the requirements. In current time, cellular traffic determining models are of high complexity, which results in slow and expensive forecasting process.

Additionally, because of the decentralized nature of cloud computing, virtualization together with SDN advancements, the basic system design for Network Slice (NS) is relied upon to be profoundly adaptable and agile. This carries with it the issue of recognizing the perfect route through NS to meet high standards of quality of service, security and so forth. Recently inspired by the development in advanced parallel computing, neural network based deep learning approach [9] is getting progressively significant right now and becoming increasingly important in this field [20].

One of the main ideas in the architecture of 5G design is that of Network Slicing [1], which is driven by various prerequisites requirements that 5G requires. The most recent need tending towards the requirements of 5G networking frameworks, is to address the issues for a variety of regularly clashing new service necessities, which are bringing another outlook change that is quickly making strides. The thought of a NS has risen as NS provide various explicit services, intended for subset of 5G use cases.

Network Slicing is a developing 5G innovation that permits a new way for service providers to provide “slices” of various resources to network occupants. By this way, a new business model [2] is presented where the service provider are planning to select buyers to get the keys to providing their services. Instinctively, this includes various difficulties that fall in the research field, which requires a point by point comprehension of the underlying mechanism. Network slicing is an initial move towards tackling the difficulties of high throughput, low latency and high power, empowering to intelligently differentiate the business model for each service. It provides the right equipment in one slice but requires exact estimations and actual specifications to suit the needs of portable traffic. Network Slice is a state-of-the-art technology that takes points of interest of SDN and NFV advancements for adapting new services which meets many requirements of various necessities which varies across the physical network. It accommodates each kind of services on a devoted system as needed and each slice consists only of the required features and services. A network slice allows you to provide different services (such as a car) that can be given by various network slices. Every one of these examples comprises of a set of virtual network functions running on the same custom bandwidth infrastructure with a particularly custom fitted coordination. Right now, heterogeneous prerequisites can be met in the similar infrastructure, as various different network slices can be coordinated and arranged to their specific needs. In addition, this can be done economically, as the diverse network slices effectively share the similar framework.

Access control and resource allocation challenges emerge for network slice technology. Our work is associated with this unique situation. In order to obtain accurate mobile traffic prediction to help network slices, in this article, we recommend to go with different deep learning models for sequence learning. The issue of traffic forecasting has been extensively examined in the current literature.

Paper Organisation: In Sect. 2, we review the literature survey on various scenario where deep learning models are used to overcome the challenges based

on mobile traffic forecasting. Section 3 described more about data acquisition and processing. Section 4 discuss various methodology of deep learning models like - MLP, Encoder-Decoder, GRU and LSTM that we used in our work. Section 5 presents the computation setup and experimental study for mobile-traffic forecasting. we consider, and presents the result obtained from our simulation. Finally, we conclude in Sect. 6 followed by future work discussion.

2 Related Work

Upto this, past investigations have analyzed on how to apply AI procedures to 5G wireless networks. In particular, these works addressed some basic issues, for example, portable traffic prediction, remote system advancement, and client conduct forecast. Several algorithms have been proposed in the literature for configuring network slices and virtual resources. The purpose of these instructions varies from algorithm to algorithm. We outlined the principle in writing on the subject of Slicing, lining hypothesis for cloud services and access to network traffic. Our concern is firmly connected to predictions of mobile traffic, some of the related work which have been done related as this are summarized below.

In [17], authors implemented two different forecasting models (MLP and ARIA) to analyze the limit usage of microwave point to point connects in mobile networks. The basic idea was to include statistical features of the mobile network with assistance of linear and nonlinear prediction models.

In [8], the authors discuss a multi-queue-based controller that have been developed for tenants to receive the network slice with given requests frequency and qualities. Their outcomes are consistent with their stated standards and demonstrate that sudden tenant behavior can be adequately addressed by appropriate access control policies.

In [12], authors propose an Overload and Requirement (OvC&R) algorithm that maximizes the QoS of a number of user requests, including their utilization cases prerequisites, including network capabilities based on availability, reliability and duration. Their experimental results show that their algorithm can select the most advantageous slice from the available network slices while limiting the complete over-burdening cost.

In [4], authors set out a framework for delivering a 5G virtual network based on descriptive applications, incorporating QCI and security prerequisites. The principle allows for exchange, selection and integrated networking for various applications in 5G systems.

As earlier discussed the availability and rapid implementation of SDN and NFV arrangements present new network protocols with significant challenges. In [6], the authors address one of the major challenges in this area: placement of virtual functions on the network. The author led an intensive hypothetical investigation in the NFV to identify network features with the aim of reducing the total cost of the network while tracking the small constraints of the nodes.

In [13], authors discuss the utility-based slice deployment algorithm (UBA) approach that provides a QoS framework for network slices including the different

use cases to determine the use of data requirements, including network data based on availability, reliability and latency.

In [3], authors use the deep learning methodology and introduce DeepCog, a new data analysis tool for knowledge management on the dynamics of sliced 5G network. DeepCog incorporates a deep neural network model, a learning-based approach which incorporates a modified loss function that focuses on predictive capabilities as opposed to inheritance mobile traffic expectation.

In [19], authors challenge the prediction problem for the multi-service mobile traffic estimation to help the organisation of network traffic. Their proposed model utilize a sequence-to-sequence learning standard as well as ConvLSTM models, which can complete up to 1-h of traffic for up to 36 services for mobile phones with high precision, given 1-h estimation based perceptions.

In [16], author addressed the CSI estimation issue for 5G communications. To productively secure the CSI well, they applied an online CSI expectation conspire, named OCEAN, for CSI prediction based on historical data in 5G wireless communications. They used learning models for the framework which is a mixture of CNN (Conv neural network) and LSTM (long short-term memory) models.

In [7], the authors propose GANSlicing, a powerful software defined network driven approach for network slicing based on GAN (Generative Adversarial Network) to anticipate the capabilities of IoT applications to enable the ability to use and improve client QoE. GANSlicing is focused to create next-generation cellular networks for better vertical industries performance with better fine-grained cutting with the dynamic asset arrange.

In [10], authors address the allocation problem of resources by designing an admission control algorithm that the server will execute in the monitoring system when getting receipt of the profile request by the tenants in the form of network slice.

In [14], takes a closer look at the hotspot forecast to support high-quality, low-latency virtual small cells (VSCs) operation. In particular, LSTM neural system predicts the number of user equipments (UEs) in each network continuously because it is possible to study long-term addiction. By examining the expected distribution of networks, hotspot can recognize to frame the VSC ahead of time.

3 Data Acquisition and Processing

3.1 Dataset Description

We have considered the Dataset-Unicauca-Version2 [18] dataset for the deep learning models, which contains a collection of six days mobile traffic data having 87 features and 35,77,296 instances. We categorized these dataset into 4 classes named as streaming, messaging, searching and cloud.

Context. The data presented here was collected in a web division from Universidad Del Cauca, Colombia, Popayan, to record a package connection on different days at various hours, during morning and evening, more than six days (April 26, 27, 28 and May 9, 11 and 15), 2017. A total of 3,577,296 samples were gathered.

Content. This dataset contains 87 features. In each example, information about IP traffic generated by network equipment that is IP address of origin and destination, port, arrival time and layer 7 protocol (application) used as a class in this traffic. Most properties are numeric, however there are additionally name and date type because of time constraints.

3.2 Data Cleaning

We removed the instances of intranet such as HTTP and also removed the features which were mainly flags and some of these features remained almost constant throughout the dataset. We classifies all the services in four categories namely streaming, messaging, searching and cloud based applications shown in Fig. 1. These four categories represent the various slices. We sorted the dataset according to time and later formed a sequence out of this to transform it into a time series forecasting problem.

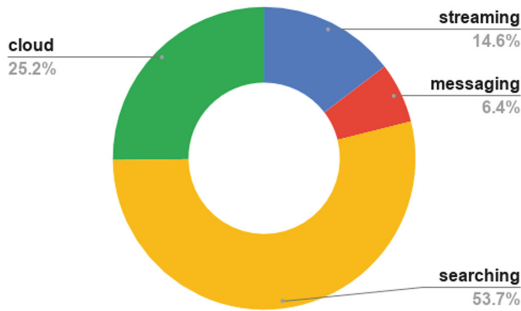


Fig. 1. Fraction of mobile-traffic consumed by service classes

The distribution of data in these four categories are described in details in Fig. 2. which helps in visualizing this distribution clearly on a log scale. The $Y - axis$ represents the number of times a service occurs in the dataset and and the $X - axis$ marks the service that is being referred to.

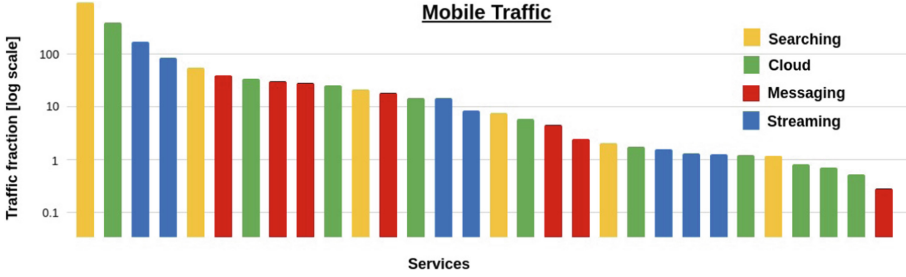


Fig. 2. Fraction of mobile-traffic consumed by each mobile-service

Color scheme is used for each slice in order to visualize it on category basis. Our dataset contains 284839 instances of Streaming category 125526 instances of Messaging services 1045920 instances of Searching services and 490121 instances of cloud based applications category. So after the cleaning of dataset we had in total 1946406 instances.

4 Methodology

4.1 Multi-Layer Perceptron (MLP)

In MLP there are more than one layer. The first layer is the input layer, then there are several hidden layers and at last there is an output layer [17]. The main goal of feed-forward networks is to typically approximate some function $f()$. More formally we try to approximate the classifier by defining $y = f(x; \theta)$ where x is the input vector and θ is the set of parameters that we try to learn. The MLP is composed of many layers and each of these layers is composed of units that perform transformation of the linear sum of inputs. An architecture of basic multi-layer perceptron with input layer, hidden layer and output layer is shown in Fig. 3.

Each layer is represented as $y = F(Wx + b)$ where W is the set of weights that we try to learn and is the input to the current layer or output of previous layer and b is the bias and f is the activation function that's applied to learn the relationship between input and output. Some common activation functions are sigmoid, tanh, relu etc.

Model Summary: In our model we formed a sequence of length 100 and each of these sequences had 78 attributes in them. We transformed it into a vector of 7800 dimension and passed it to our neural network with 2 hidden layers. The

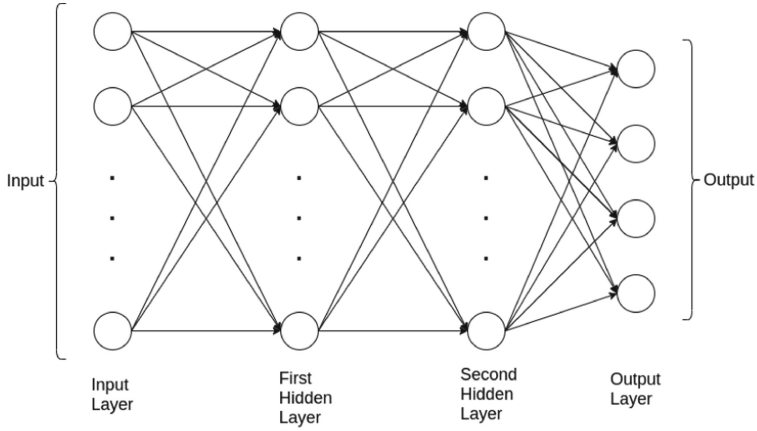


Fig. 3. MLP architecture

first layer has 500 units,second layer has 400 units. Finally we had softmax at the output layer.

```
MLP(
  (layers): Sequential(
    (0): Linear(in_features=7800, out_features=500, bias=True)
    (1): ReLU()
    (2): Linear(in_features=500, out_features=400, bias=True)
    (3): Tanh()
    (4): Linear(in_features=400, out_features=4, bias=True)
    (5): Softmax(dim=1)
  )
)
```

4.2 Encoder-Decoder

Encoder: It acknowledge, processes, collects and distributes one component of input sequence at each step. **Intermediate Vector:** The encoder segment of model generates the last internal state i.e. intermediate vector [15]. Contains data regarding whole input grouping to enable the decoder-part to produce result as per the expectations.

Decoder: Given the entire sequence input, it gives an output at each timestamp. The Encoder as well as Decoder in our architecture uses LSTM cell. In our architecture we also applied attention mechanism to the architecture to capture important attributes.

Attention Mechanism: In most of the series problems some of the attributes influence the output more than others, and in order to give special attention to these attributes we applied attention mechanism. Basically in attention we assign

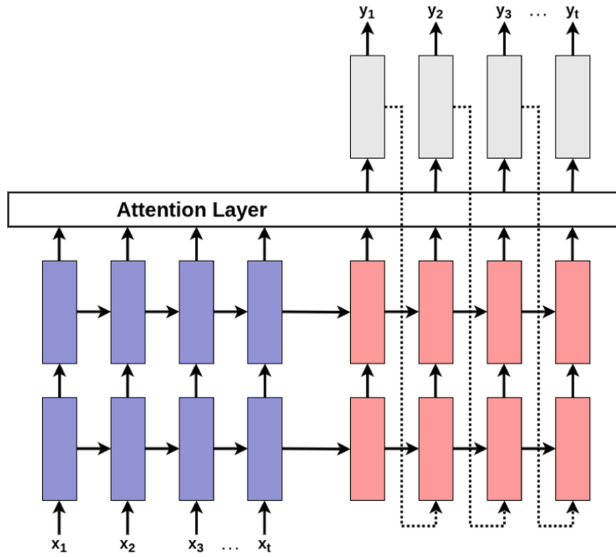


Fig. 4. Encoder-decoder-attention based architecture

weight with each of the attributes and try to capture the dependencies of these attributes. An architecture of Encoder-Decoder-Attention based architecture is shown in Fig. 4.

Model Summary: In our model we formed a sequence of length 100 and each of these sequences had 78 attributes in them. The Encoder layer has single layer which transform the input in 400 dimension and then we apply attention mechanism on this. Finally we pass it through the decoder layer which transforms it into 25 dimension embedding and passes it through a fully connected artificial neural network with softmax as activation function.

```
Encoder_Decoder_Attention_based(
  (encoder): Encoder(
    (lstm): LSTM(41, 400)
  )
  (attentionDecoder): AttentionDecoder(
    (attn): Linear(in_features=425, out_features=1, bias=True)
    (lstm): LSTM(404, 25)
    (final): Sequential(
      (0): Dropout(p=0.1, inplace=False)
      (1): Linear(in_features=25, out_features=4, bias=True)
      (2): Softmax(dim=1)
    )
  )
)
```

4.3 Gated Recurrent Unit (GRU)

RNN (Recurrent Neural Network), a type of artificial neural network and it belongs to one of the amazing algorithms present at the moment due to its ability to store information and manipulate information within its memory unit [5]. Its intrinsic memory capacity permit it to display powerful unique behavior over a time period succession. Unlike feed-forward neural systems, RNNs can utilize their inside state (memory) to process series of inputs.

$$\begin{aligned}
 z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
 r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
 \vec{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\
 h_t &= (1 - z_t) * h_{t-1} + z_t * \vec{h}_t
 \end{aligned}$$

RNNs are comparable yet extraordinary; every layer has certain number of RNN memory units that is fed with the input say $(Y1, Y2, Y3, \dots)$ which represents the input at each time step and each of these memory units output a context vector/state vector which may be of some other dimension say $(A1, A2, A3, \dots)$. The output is created by connecting vector A to the input Y . However, a refreshed vector A is created which is carried to following layer. The representation of single unit of GRU model is shown in Fig. 5.

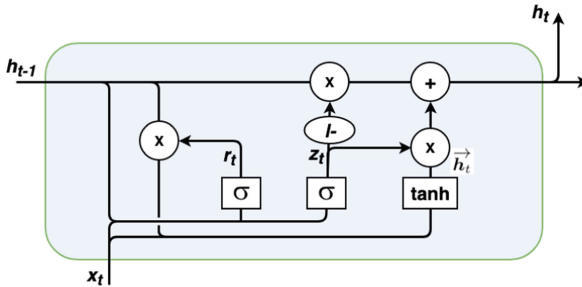


Fig. 5. Single unit representation of GRU model

Model Summary: A GRU as opposed to having a straightforward neural system with four nodes as the RNN had already has a cell that contains many functions. Presently the model that repeats each arrangement is the three models, each of which can be a neural system. GRU utilizes the update gate and reset gate. In our model we used 2 layers of GRU network where each layer outputs a 400 dimension and finally connecting the output of last layer with a feed-forward neural network and softmax as activation function. We also applied dropout with ratio of 0.4.

```
timeNet(
  (lstm): GRU(78, 400, num_layers=2, batch_first=True, dropout=0.4)
  (dropout): Dropout(p=0.4, inplace=False)
  (linear): Linear(in_features=400, out_features=4, bias=True)
)
```

4.4 Long Short-Term Memory (LSTM)

The Long Short Term Memory networks, often called LSTM, is a type of RNN that can learn long-term addiction. They were presented by Hochreiter and Schmidhuber in the year 1997. LSTM are structured to stay away from long-term addiction issue. Rather than having a single NN layer, they contain 4 different sorts of layer that work together in a different manner [16]. LSTM are able to expel and add data to cellular state, which is performed with a very much planned Gates. These Gates provide an approach to alternatively let data through. They consist of a sigmoid NN layer and a clockwise increase activity. The representation of single unit of LSTM model used in our implementation is shown in Fig. 6.

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means nothing should be allowed to pass while a value of one means everything should be allowed to pass.

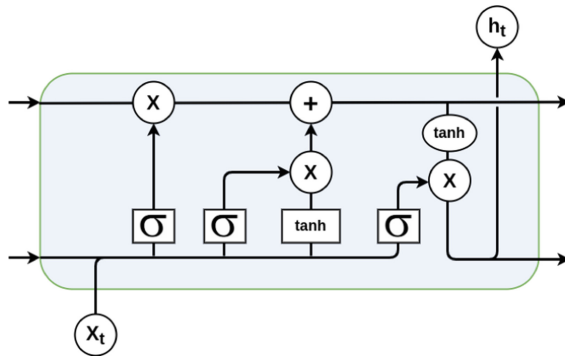


Fig. 6. Single unit representation of LSTM model

Model Summary: In our architecture we used 2 layers of LSTM networks where the output is of dimension 30. Finally we connect the output of the last layer with feed-forward neural network and used softmax as the activation function.

```

timeNet(
  (lstm): LSTM(41, 30, num_layers=2)
  (fc): Sequential(
    (0): Dropout(p=0.1, inplace=False)
    (1): Linear(in_features=30, out_features=4, bias=True)
    (2): Softmax(dim=1)
  )
)

```

5 Result and Discussion

5.1 Experimental Setup

We transformed the problem as discussed above into a time-series forecasting problem. For this we made a sequence of length 100 from previous time-steps and tried to predict the output of the next time-step. We kept 78 attributes out of the 87 attributes and partitioned the dataset in the 70:30 ratio for training and testing. We took 0.1 fraction of the training dataset for validation purpose. In all the Deep-learning models that we applied we used weighted categorical cross Entropy as the loss function. The weight of each of these class was calculated based on the number of their instances in the training set.

5.2 Result and Comparison

In the simple MLP model we converted our input vector to single dimension vector of size 7800 since it takes 1d vector as input. For the GRU, LSTM and Encoder-Decoder model we passed in the input as a matrix of size [100,78] where 100 represents the sequence-length and 78 represents the number of attributes. Since this is a classification task therefore we used softmax as an activation function on the output layer. The classification probabilities improved a bit with a reduction in sequence size from 100 to 10. We trained all the models for 50 epochs on the GPU. All the Architectures were made using pytorch library. Finally, we also tried the encoder-decoder architecture with attention mechanism. The basic intuition behind this was not all the attributes contribute same for the classification of these services So, in order to capture this dependence we applied the attention mechanism. We did a comparison of our models on the parameters of

Table 1. Recall values of different class

Recall values				
Model name	Streaming	Messaging	Searching	Cloud
MLP	1	0	0	0
Encoder-Decoder	0	0	1	0
GRU	0.08	0.00095	0.52	0.396
LSTM	0.037	2.60E-05	7.84E-01	1.77E-01

Table 2. Precision values of different class

Precision values				
Model name	Streaming	Messaging	Searching	Cloud
MLP	0.148	0	0	0
Encoder-Decoder	0	0	0.65745133	0
GRU	0.144	0.055	0.5393	0.249
LSTM	0.145	0.045	0.538	0.251

Table 3. F1-score values of different class

F1-score values				
Model name	Streaming	Messaging	Searching	Cloud
MLP	0.258	0	0	0
Encoder-Decoder	0	0	0.79332807	0
GRU	0.103	0.0018	0.529	0.3
LSTM	5.90E-02	5.32E-05	6.38E-01	2.08E-01

Recall, Precision and F1-score values. The results obtained from our experiment are shown in Table 1, 2 and 3.

As we can see from the above tables MLP, Encoder-Decoder models performed average for mobile-traffic forecasting, while the GRU, LSTM performs comparatively well and are able to differentiate among different mobile service classes. LSTM and GRU has almost similar values for precision, but on recall matrix we can see LSTM has better result than GRU.

In the above results shown in Fig. 7, describe the results obtained from service categories for mobile-traffic classes i.e. streaming, messaging, searching and cloud that we have contemplated in this work. In Fig. 7(a), shows the result obtained for recall values considering the four classes which indicates that GRU and LSTM performs better but it is somewhat biased for MLP and Encoder-Decoder models. In Fig. 7(b), shows the obtained precision value that goes well for GRU, LSTM and encoder-decoder-attention based models which results in a high precision value for searching class. Further in Fig. 7(c), the f1-score performance matrix describes that f1 levels goes well for GRU, LSTM models and Encoder-Decoder is biased for messaging class. The f1-score provides average performance for MLP model.

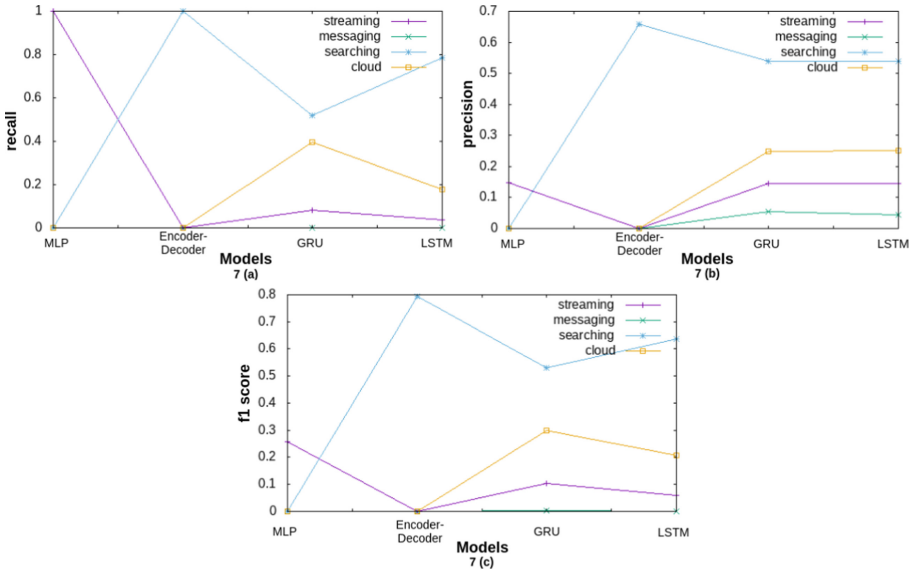


Fig. 7. Results obtained from MLP, Encoder-Decoder, GRU and LSTM model (a) shows the recall (b) shows the precision and (c) shows the f1 score values.

6 Conclusion

In this paper, we investigated the MLP, Attention-based Encoder Decoder, GRU and LSTM deep learning models on Dataset-Unicauca-Version2 mobile-traffic dataset. We have clean the dataset for our work, mainly we considered four classes of mobile-services namely streaming, messaging, searching and cloud. In particular for resource utilization in 5G network slicing, we have applied recall, precision and f1-score performance matrices for mobile-traffic forecasting. We found that MLP, Encoder-Decoder models performed average for mobile-traffic forecasting while the GRU, LSTM performs well and out of them LSTM obtained best result. In future, we plan to work on other time-prediction approaches for resources. We are also interested to work in Multi-access Edge Computing (MEC) in IIoT applications to support industry 4.0.

References

1. Alliance, N.: Description of network slicing concept. NGMN 5G P 1, 1 (2016)
2. Association, G.S., et al.: 5g network slicing for cross industry digitization. 5G Slicing Association, Tech. Rep (2018)
3. Bega, D., Gramaglia, M., Fiore, M., Banchs, A., Costa-Perez, X.: Deepcog: Cognitive network management in sliced 5g networks with deep learning. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications. pp. 280–288. IEEE (2019)

4. Choyi, V.K., Abdel-Hamid, A., Shah, Y., Ferdi, S., Brusilovsky, A.: Network slice selection, assignment and routing within 5g networks. In: 2016 IEEE Conference on Standards for Communications and Networking (CSCN). pp. 1–7. IEEE (2016)
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555) (2014)
6. Cohen, R., Lewin-Eytan, L., Naor, J.S., Raz, D.: Near optimal placement of virtual network functions. In: 2015 IEEE Conference on Computer Communications (INFOCOM). pp. 1346–1354. IEEE (2015)
7. Gu, R., Zhang, J.: Ganslicing: A gan-based software defined mobile network slicing scheme for iot applications. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). pp. 1–7. IEEE (2019)
8. Han, B., Sciancalepore, V., Feng, D., Costa-Perez, X., Schotten, H.D.: A utility-driven multi-queue admission control solution for network slicing. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications. pp. 55–63. IEEE (2019)
9. Hu, H., Liu, Z., An, J.: Mining mobile intelligence for wireless systems: A deep neural network approach. *IEEE Computational Intelligence Magazine* **15**(1), 24–31 (2020)
10. Hua, Y., Li, R., Zhao, Z., Zhang, H., Chen, X.: Gan-based deep distributional reinforcement learning for resource management in network slicing. arXiv preprint [arXiv:1905.03929](https://arxiv.org/abs/1905.03929) (2019)
11. Index, C.V.N.: Forecast and methodology 2017–2022. San Jose, CA, USA, Cisco (2019)
12. Kammoun, A., Tabbane, N., Diaz, G., Achir, N.: Admission control algorithm for network slicing management in sdn-nfv environment. In: 2018 6th International Conference on Multimedia Computing and Systems (ICMCS). pp. 1–6. IEEE (2018)
13. Kammoun, A., Tabbane, N., Diaz, G., Dandoush, A., Achir, N.: End-to-end efficient heuristic algorithm for 5g network slicing. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA). pp. 386–392. IEEE (2018)
14. Liu, Y., Wang, X., Boudreau, G., Sediq, A.B., Abou-zeid, H.: Deep learning based hotspot prediction and beam management for adaptive virtual small cell in 5g networks. *IEEE Transactions on Emerging Topics in Computational Intelligence* **4**, 83–94 (2020)
15. Loyola, P., Liu, C., Hirate, Y.: Modeling user session and intent with an attention-based encoder-decoder architecture. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. pp. 147–151 (2017)
16. Luo, C., Ji, J., Wang, Q., Chen, X., Li, P.: Channel state information prediction for 5g wireless communications: A deep learning approach. *IEEE Transactions on Network Science and Engineering* (2018)
17. Mahmood, A., et al.: Capacity and frequency optimization of wireless backhaul network using traffic forecasting. *IEEE Access* (2020)
18. Rojas, J.S.: Ip network traffic flows labeled with 75 apps - labeled ip flows with their application protocol, (2017). <https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps>

19. Zhang, C., Fiore, M., Patras, P.: Multi-service mobile traffic forecasting via convolutional long short-term memories. In: 2019 IEEE International Symposium on Measurements & Networking (M&N). pp. 1–6. IEEE (2019)
20. Zhang, C., Patras, P., Haddadi, H.: Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials* **21**(3), 2224–2287 (2019)



Software Defined Radio Based Multi-band Audio Broadcasting System for Drone Based Communications

Yaswanth Chalamalasetti^(✉), Sudhir Kumar Sahoo, Barathram Ramkumar,
and M. Sabarimalai Manikandan

Wireless Communication Lab, School of Electrical Sciences, Indian Institute
of Technology Bhubaneswar, Jatani, Khordha 752050, Odisha, India
{cy10,sks14,barathram,msm}@iitbbs.ac.in

Abstract. Traditional light weight wireless communication modules like ZigBee, Bluetooth, LoRa have fixed frequency band and fixed protocol stack and hence lack flexibility of changing frequency, modulation techniques, waveforms etc. and may also prone to interference, jamming. Due to obstacles in between the transmitter and receiver, the communication coverage range is not large. But in emergency situations like war field or any disaster, there is need for robust communication in which the soldiers or public have to receive crucial information from base station. Software defined Radios like Universal Radio Software Peripheral (USRP), WARP boards needs to interface with computer for signal processing. Hence it is not suitable for drone based communication. So we proposed a solution to overcome bulkier system and limited communication range problem. In this work, a light-weight communication platform was developed which acts as a repeater for broadcasting voice commands that were transmitted from base station. The repeater is mounted on the drone and it is hovered at a certain altitude to make line of sight communication possible with both transmitter and receiver. This system was designed by interfacing Radio Frequency (RF) board present in USRP 2900 with Raspberry Pi 4, a credit card sized computer. The RF board overcomes the difficulty of standard wireless communication modules mentioned above, could easily mounted on drone, capable of communicating in different modulation schemes at multiple frequency bands and also one could implement cognitive radio features like spectrum sensing and taking automatic decisions based on data which is obtained from various sensors that are connected to Raspberry Pi.

Keywords: Drone · Repeater · USRP · Raspberry Pi · Software defined radio · Python · Spectrum sensing

1 Introduction

In war field or in the case of disaster, information should be broadcasted to soldiers or public. But due to many obstacles like buildings, hills, trees, vehicles etc.

This research work is supported by IMPRINT-II and MHRD Grant, Govt. of India.

That are present in between base station and receivers, line of sight communication may not be possible. Also due to limited coverage range, people who are far away might not receive the broadcasted message from base station. To address these problems, a line of sight communication from base station to drone (that is hovering at certain height) and drone to receiver was proposed, so that coverage range could be increased as shown in Fig. 1. A brief literature overview of audio broadcasting using various lightweight wireless modules are summarised below. In [1], A. Meliones et al. proved that the Zigbee's radio channel has enough bandwidth to support a full duplex communication by using narrow band voice codecs. In [2], D. Brunelli et al. investigated the feasibility of voice transmission over ZigBee platform. A custom hardware using ZigBee module and an application to analyze the ability of ZigBee protocol for streaming voice at low-rates were developed. In [3], A. Rowe et al. designed a module for streaming voice over wireless sensor networks. A sensor node was developed which adopts a TDMA-based network scheduling to meet the requirements audio timing. In [4], T. Facchinetti et al. developed an embedded device for the wireless broadcasting of voice signals over IEEE 802.15.4 with low-power 16-bit microcontroller under low-cost. A real time operating system (RTOS), a psychoacoustic model based on FFT signal decomposition and the Haar wavelet transform are used to develop a new audio compression algorithm. In [5], L. Meiqin et al. developed a stand-alone module that utilizes an IEEE 802.15.4-compliant RF transceiver. Audio input and output ports are integrated on the board to connect microphone and speaker. The disadvantage of the above works is that, the wireless communication modules supports only fixed frequency band and protocols. This is not suitable for emergency or defence applications where flexibility is required. In [6], J. Modares et al. designed a communications testbed using SDR transceiver. This testbed is used for quick testing and prototyping of various drone based communication protocols and is based on SDR devices like USRP E310 that runs on Linux operating system. In [7], S.G.E. Brucal et al. have designed Public Addressing system by building it on USRP using LabVIEW platform to obtain bandwidth efficiency, audio clarity, and long-range transmission. To test maximum possible non line of sight distance, different materials were placed midway as obstructions and the receiver was able to receive signals up to 67 m. The disadvantage of the above two works is that the SDR platforms are expensive due its high processing capability and requires bulkier software like LabView which are not required for simple audio broadcasting.

From the above literature review it can be seen that ZigBee based audio broadcasting systems lack flexibility whereas SDR based audio broadcasting system requires expensive hardware and software modules. Also traditional radio systems can communicate only at certain bands of frequencies. An attacker can interfere in particular band and can sniff or jam the communication happening in that band. Protocols like ZigBee, LoRa, Bluetooth, Wifi are already reverse engineered which have a high probability of tapping the communication. These wireless modules communicates at fixed frequency band and also there is no scope to change frequency, modulation techniques, etc. so these systems fail to

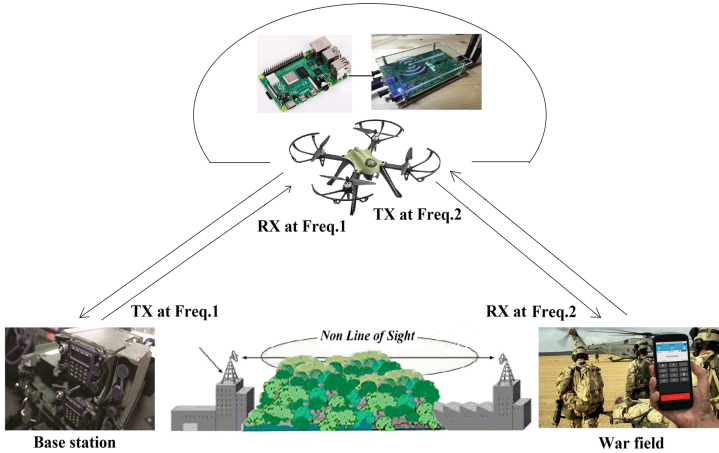


Fig. 1. Typical audio broadcasting scenario

provide security. That's why there is a need for a flexible radio which can provide secured encrypted communication and could also change frequency automatically without manual operator when that band is being jammed. So a lightweight transceiver system for audio transmission and reception is developed in which RF board from USRP 2900 is interfaced to Raspberry Pi 4 platform. The proposed system can act as a transmitter, receiver and repeater which can easily be mounted on a drone. Baseband processing is done on Raspberry Pi 4 using simple python programming language instead of bulkier packages like LabView. Unlike traditional wireless communication modules, the proposed system supports multiple frequency bands, modulation schemes and data rates. The proposed system has non-LoS distance of 80m and LoS distance of 630m which is better than the range achieved by [7].

This paper is organized as follows. In Sect. 2, the proposed system is explained. In Sect. 3, the Range calculations are explained. The results are summarized in Sect. 4 followed by conclusion and future work.

2 Proposed System

In this section, details of proposed system are discussed. Since USRP and laptop setup cannot be mounted on drone as repeater, RF board is interfaced with Raspberry Pi 4 so that it can be mounted on drone. Two antennas are connected to respective SubMiniature version A (SMA) ports of RF board in which one acts as transmitting antenna and other acts as receiving antenna. The RF board that was used in this setup has operating frequency range from 70 MHz to 6 GHz, max. output power 20 dBm, max. input power -15 dBm, transmitter gain 89.75 dB, receiver gain 76 dB. RF hardware board is connected to Raspberry Pi using USRP Hardware Driver(UHD) which acts as a pipe in between python programming language and RF hardware board. Raspberry Pi is a small credit card

sized computer running on Raspbian operating system which is used as signal processing unit. The proposed transceiver setup shown in Fig. 2 is designed to act as transmitter, receiver, and repeater for audio transmission over a long distance in different frequency bands. The transceiver system must be light weight to mount on drone and requires proper selection of antennas, baseband algorithms, programming language and interfacing drivers. These are explained in following sections.

2.1 Antenna Selection

The antennas that were chosen in this setup are vertical dipole antenna (VERT 900) which supports two frequency bands 824 MHz–945 MHz and 1710 MHz–2180 MHz. Before using antenna, frequency for communication need to be fixed. For that return loss plot has been analyzed to find the frequency where reflection coefficient S_{11} is less. Less return loss at a particular frequency signifies that there won't be any lose due to reflection of power while transmission and reception from RF board at that particular frequency. For testing the antenna, ENA Series Network Analyzer, Model: E5071C from Agilent Technologies was used. It was found that less value of reflection coefficient S_{11} at 5 different frequencies below -10 dB impedance bandwidth. Among them, 862MHz and 1990 MHz are not interfering with other communication protocols. Hence 862 MHz was used as uplink frequency where S_{11} is -14 dB and 1990 MHz was used as downlink frequency at which S_{11} is -22 dB. Two frequencies were considered so that there won't be any interference while transmitting by base station and drone transceiver.

2.2 Baseband Algorithms

As mentioned earlier the proposed system will act as transmitter, repeater (that can be mounted on drone) and receiver. The baseband algorithm for each is explained below.

Base station (TX). In this work, both live audio and recorded audio are considered. The audio is sampled at sample rate of 48 KHz by the ADC present in Raspberry Pi. The bit stream is then modulated using Wide Band Frequency Modulation in python code that is running on Raspberry Pi. Maximum deviation considered is 75 KHz. The baseband signal is upconverted to 862 MHz by RF board and transmitted on air from VERT900 antenna connected to SMA port at sample rate of 240 KSps. The designed transmitter is shown in Fig. 2(a). The flow graph of the transmitter is shown in Fig. 3.

Drone (RX+TX). The transceiver mounted on drone receives the signal from base station at sample rate of 240KSps on uplink frequency 862 MHz and re-transmits the same received data by resampling it at 256 KSps from drone to receiver on downlink frequency 1990 MHz by increasing signal power. The modulation scheme used is same as that of base station. The designed repeater mounted on the drone is shown in Fig. 2(b).

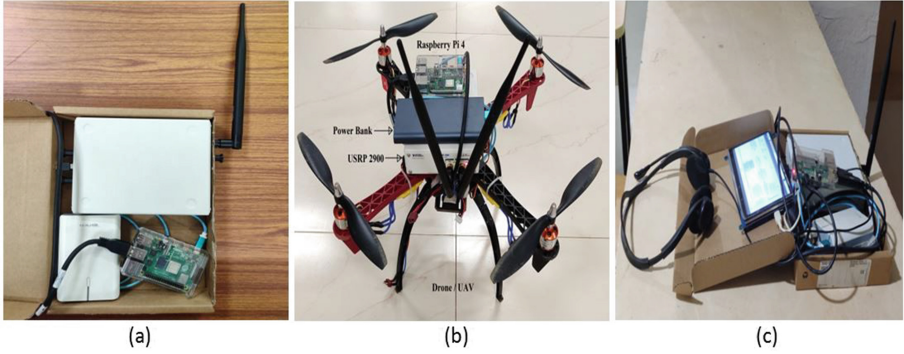


Fig. 2. (a) Base station, (b) Repeater on Drone, (c) Receiver

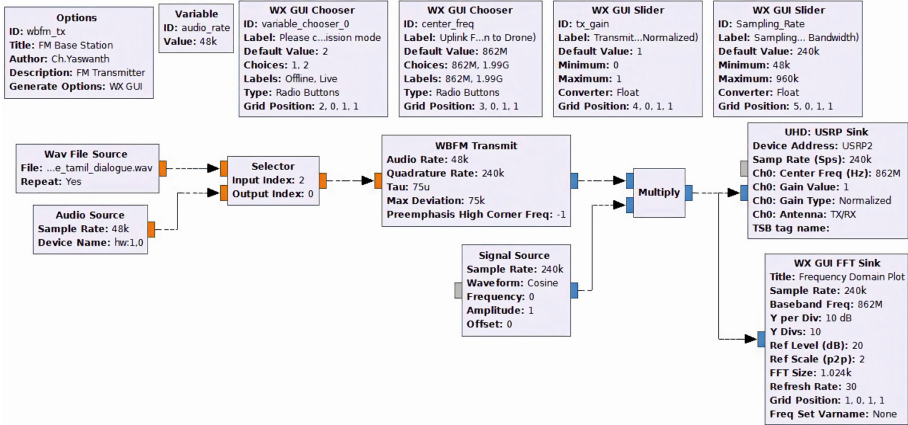


Fig. 3. Transmitter flow graph

Receiver (RX). The receiver receives the signal at sample rate of 256 KSPs on downlink frequency 1990 MHz and then it is downconverted to baseband signal. This signal is sent to Low pass filter with cut-off frequency 30 KHz and demodulation is done by python code on Raspberry Pi. The demodulated signal is resampled to 48 KHz so that it can be listened in speaker. The designed receiver is shown in Fig. 2(c). The flow graph of the receiver is shown in Fig. 4.

2.3 Graphical User Interface

Raspberry Pi is connected to an Internet access point with inbuilt Wi-Fi chip present on it and Virtual Network Computing (VNC) Server is started. By connecting client/mobile to same access point, signal processing parameters can be

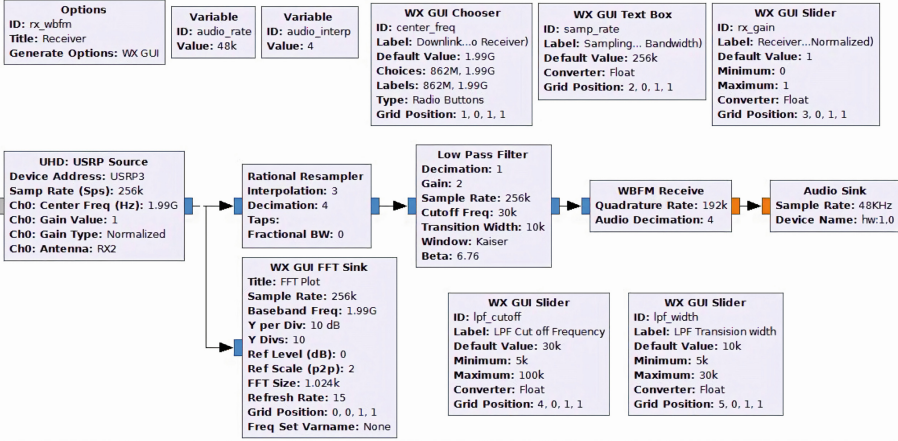


Fig. 4. Receiver flow graph

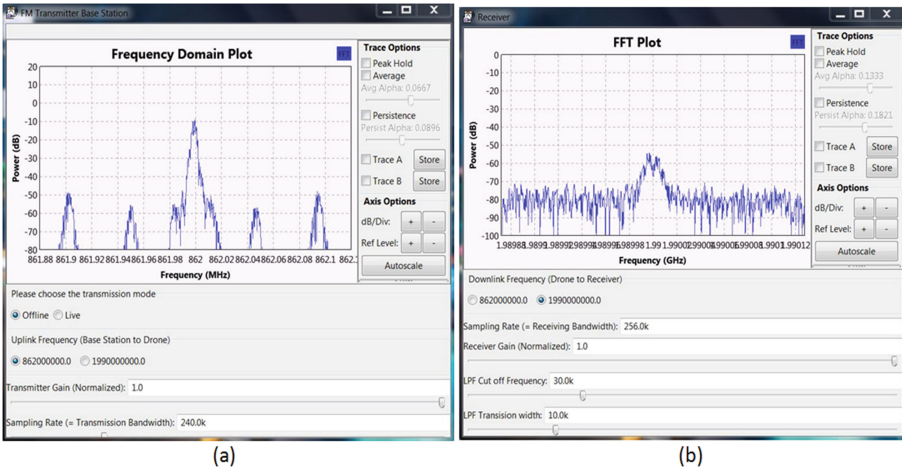


Fig. 5. (a) Transmitter GUI (b) Receiver GUI

controlled from VNC Client using Graphical user interface that are shown in Fig. 5 which are made with python programming language.

3 Range Calculations

The performance of proposed system was analyzed for different distances in two cases, non line of sight and line of sight. In first case, as shown in Fig. 6(a), transmitter (base station) is in front of School of electrical sciences (SES) building at IIT Bhubaneswar and receiver is exactly behind SES building at a horizontal distance of 80 m in non-LoS and it is receiving voice commands with acceptable



Fig. 6. (a) Non Line of Sight Distance (b) Line of Sight Distance

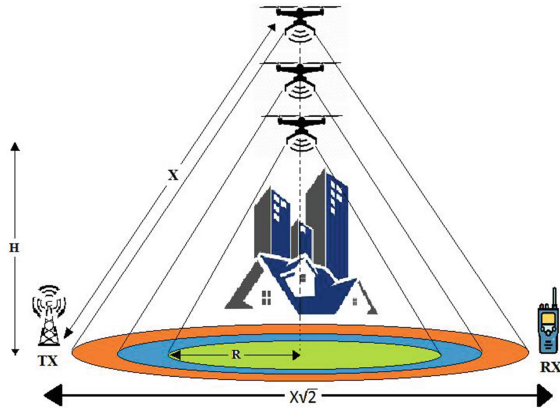


Fig. 7. Distance calculations

quality despite the presence of multiple obstacles. When receiver is moved more than 80 m., signal from transmitter is not being received. So its concluded that in non-LoS mode the max. Distance between transmitter and receiver can be upto 80 m. In second case, as shown in Fig. 6(b), transmitter is placed at point ‘A’ and receiver is placed at point ‘B’ the signal is received with acceptable quality, the distance between points ‘A’ and ‘B’ is 630 m. For calculating the maximum coverage area for drone based audio broadcasting, the method is discussed below according to Fig. 7. The results are summarized in Table 1.

Table 1. Coverage range

Method	Coverage
Tx. directly from base station to Receiver (Non-LoS)	80 m horizontal length
Tx. directly from base station to Receiver (LoS)	630 m horizontal length
Tx. from drone which is hovering (proposed method)	180,000 m ² circular area

Now calculations are done to find the maximum possible distance that a receiver can be taken from transmitter in non line of sight without failure of communication link.

Let 'X' be the maximum possible LoS communication distance. If drone's height (H) = Horizontal distance (R) = $X/\sqrt{2}$, then maximum non-LoS communication is possible up to a distance of $X\sqrt{2}$. Therefore maximum circular coverage area on ground as seen from drone is $X^2/2$.

Due to presence of obstacles, command may not reach soldier's walkie-talkie in war field. But they will reach the transreceiver present on drone since it is hovering at 630m slant distance (X) in LoS from base station. The repeater which was designed is placed on top of drone and hovered at 445 m height (H) from ground level and taken 445 m horizontal distance (R) towards receiver from transmitter as shown in Fig. 7. So by looking the ground from drone's point of view, circular area of 180,000 m² will be covered at 1.84 sr. Now base station will send voice commands to drone over uplink frequency 862 MHz. As drone's receiver is tuned to 862 MHz, it will receive the voice commands and these are re-transmitted by transmitter at downlink frequency 1990 MHz by increasing the signal power. Since the receiver is tuned to 1990 MHz, it receives voice commands that were sent from base station through repeater present on drone.

4 Results

Quality of voice signal was evaluated at receiver end at 3 different frequencies and at 7 different distances both in LoS and Non – LoS by using 4 metrics PESQ, SIG scale, BAK scale, and OVRL. The results of these quality metrics are summarized in Table 2. A brief summary of above mentioned metrics are discussed below.

4.1 Objective Evaluation Test

PESQ (Perceptual Evaluation of Speech Quality) is a standard which comprises a methodology for evaluating speech quality as experienced by user. PESQ algorithm analyzes the speech signal sample-by-sample after a temporal alignment of reference and test signal. PESQ combines the excellent psycho-acoustic and cognitive model with a time alignment algorithm that handles varying delays perfectly. The most eminent result of PESQ is Mean Opinion Score (MOS). It directly tells the voice quality. The PESQ MOS as defined by the ITU recommendation P.862 ranges from 1.0 (worst) up to 4.5 (best).

4.2 Subjective Evaluation Tests

The subjective listening tests are designed to evaluate the speech quality along three dimensions: signal distortion, noise distortion, and overall quality. This method instructs the listener to successively attend to rate the degraded speech signal by comparing with original speech signal on a five-point scale of signal distortion (SIG), a five-point scale of background intrusiveness (BAK). The overall quality uses the scale of the mean opinion score – [1 = Bad, 2 = Poor, 3 = Fair, 4 = Good, 5 = Excellent]. A total of 20 listeners were recruited for the listening tests and the average values are mentioned in Table 2.

Table 2. Quality metrics of voice signal

Mode	Distance	Frequency	PESQ	SIG Scale	BAK Scale	OVRL
Non - Line of Sight	30 m	1990 MHz	3.71	4.2	3.9	Excellent
		960 MHz	2.79	3.3	4.0	Good
		862 MHz	3.45	4.6	3.8	Excellent
	50 m	1990 MHz	2.80	4.4	4.3	Good
		960 MHz	2.52	3.9	3.8	Good
		862 MHz	2.92	4.5	4.2	Excellent
	80 m	1990 MHz	2.63	2.2	1.9	Fair
		960 MHz	2.37	2.8	3.0	Poor
		862 MHz	2.73	3.5	2.9	Fair
	100 m	1990 MHz	2.11	1.5	1.8	Bad
		960 MHz	1.97	2.0	1.6	Bad
		862 MHz	2.45	2.2	2.1	Poor
Line of Sight	250 m	1990 MHz	4.23	3.9	4.0	Excellent
		960 MHz	3.57	3.2	3.5	Good
		862 MHz	3.45	4.2	4.1	Excellent
	400 m	1990 MHz	3.98	4.1	3.8	Fair
		960 MHz	2.94	3.7	3.1	Fair
		862 MHz	3.25	3.9	4.2	Good
	630 m	1990 MHz	2.86	4.2	2.9	Bad
		960 MHz	2.38	3.7	3.1	Fair
		862 MHz	3.37	3.6	3.5	Good

4.3 Observation

In Non-LoS case, at all 3 frequencies, when the distance is increasing between base station and receiver, PESQ value is decreasing. From SIG value, BAK

value and OVRL it was evident that audio quality is decreasing as the distance is increasing between base station and receiver. In LoS case, the audio quality is Excellent at 862 MHz and 1990 MHz. Hence we considered 862 MHz as uplink frequency and 1990 MHz as downlink frequency.

5 Conclusion and Future Work

In this work, drone based light weight audio broadcasting transceiver is designed. The system is build using Raspberry Pi 4 and RF components from USRP 2900. The performance is evaluated for various distance (LoS and non LoS). The audio transmission was achieved for non LoS range 80m and LoS range 630m with acceptable quality.

In initial phase, for testing purpose analog FM modulation technique was used, next it is going to be extended for digital modulation technique to have encrypted communication. Here operator is manually tuning to some particular uplink and downlink frequencies. In real time, the frequency should change automatically when attacker is trying to jam the band. Receiver which is connected to Raspberry Pi will sweep given range of frequencies and sends an image of spectrogram plot to base station for knowing the vacant bands and occupied bands in spectrum. A camera module that is connected to Raspberry Pi will transmit live video of war field or emergency situation to base station in the vacant bands (white spaces). Some parameters of channel like temperature, humidity, dust, gas, etc. can be known from the values obtained from respective sensors that are connected to Raspberry Pi. So based on these channel conditions and availability of white spaces, operating frequency should automatically change. By using a custom protocol, the frequency which got changed should be informed to the base station and to the receiver so that there won't be any loss of information.

References

1. Meliones, A., Touloupis, E., Perello, J., Serrat, A.: Z-Phone: design and implementation of embedded voice over ZigBee applications. In: IEEE Symposium on Computers and Communications (ISCC), pp. 1–7. IEEE (2014)
2. Brunelli, D., Teodorani, L.: Improving audio streaming over multi-hop ZigBee networks. In: IEEE Symposium on Computers and Communications, pp. 31–36. IEEE (2008)
3. Mangharam, R., Rowe, A., Rajkumar, R., Suzuki, R.: Voice over sensor networks. In: IEEE International Real-Time Systems Symposium (RTSS), pp. 291–302. IEEE (2006)
4. Facchinetti, T., Ghibaudi, M., Goldoni, E., Savioli, A.: Real-time voice streaming over IEEE 802.15.4. In: IEEE Symposium on Computers and Communications, pp. 985–990. IEEE (2010)
5. Meiqin, L., Yuxuan, W., Zhen, F., Senlin, Z.: Voice communication based on ZigBee wireless sensor networks. In: IEEE Proceedings of Chinese Control Conference, pp. 389–394. IEEE (2014)

6. Modares, J., Mastronade, N.: UB-ANC Drone: A Flexible Airborne Networking and Communications Testbed. arXiv preprint [arXiv:1509.08346](https://arxiv.org/abs/1509.08346) (2015)
7. Brucal, S.G.E., Bautista, G.M., Cruz, K.J.D.D., Lauguico, S.C., Soriano, J.A.D.: Design of an audio transmitter with variable frequency modulation parameters using national instruments LabVIEW 2011 and USRP-2920 for public address system improvement of Asia pacific college. In: IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), pp. 1–6. IEEE (2018)



Genetic Algorithm Based Feature Selection for Software Reliability Prediction Using Multi-layer Perceptron

Priyanka Kumari, Kulamala Vinod Kumar^(✉), and Durga Prasad Mohapatra

NIT Rourkela, Orisha, India
durga@nitrkl.ac.in

Abstract. Software reliability depends on the number of faults present in the software system (more number of faults in the software means reliability is less and vice versa). In this paper, two software fault prediction models are developed using Support Vector Machines (SVM) and Multi-layer Perceptron (MLP). Genetic Algorithm (GA) is applied to select the best features. Then these reduced features are considered as input to the two classifier algorithms such as SVM and MLP. The developed models are applied on 6 different datasets collected from Github Repository (Nasa Defect Dataset). From the results, it is observed that MLP with GA model is performing better than other models.

Keywords: Support vector machine · Multi-layer perceptron · Genetic algorithm · Software reliability

1 Introduction

The motive of this paper is to propose models to improve the software quality attributes such as software reliability using machine learning techniques. Each technique has its own advantages and benefits but for good results need good implementation. In this paper, three well liked techniques called Support Vector Machines, Multi-Layer Perceptron and Genetic Algorithms are implemented by us. Software Reliability is main factor to become system dependent. For data processing system, we use machine learning with some set of rules and sample and the system train the incoming data. Machine learning algorithms produce generic solution they do not care about the problem you are solving. In this paper, we apply support vector machine, multi-layer perceptron and genetic algorithm and estimate the performance measure of software system in term of accuracy, F-measure and AUC. Software reliability is much more important in safety critical systems like airplanes. We extract software metrics for each module of the software. Then, these software metrics will be input to the machine learning model and the output is whether the module is faulty or non-faulty.

The main goals of our research work are as follows:

- To develop machine learning based models to estimate and predict the reliability of software systems.

- To use hybrid machine learning techniques to predict reliability of software systems.
- To compare the performance of the proposed approaches with that of the existing approaches [6].

The rest of this paper is organized as follows: Sect. 2 describes the basic concepts such as pre-processing data techniques and some machine learning techniques namely SVM, MLP and GA. Section 3 briefly includes literature survey. In Sect. 4, we explain the proposed methodology namely software reliability prediction using SVM/MLP and software reliability prediction using SVM/MLP with features selection. Section 5 introduces implementation and results of our proposed models. Section 6 presents comparisons with existing models. Finally, conclusions and future work is described in Sect. 7.

2 Basic Concepts

In this section, we briefly discuss the basic concepts which are required to understand our proposed approach.

2.1 Preprocessing Data

To compare performance of the model across datasets with different ranges and distributions, it is often helpful to normalize the data within some limits. We use the feature scaling normalization technique as given in Eq. 1.

$$\text{Normalize}(x_i) = \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} \quad (1)$$

2.2 Support Vector Machine (SVM)

SVM is a supervised model for classification and regression problems. SVM is used for both purpose linear and non-linear problems. The idea of SVM is: the algorithm draws a hyperplane which separates the data points into classes. The basic ideas of SVM are Max-Margin and Kernels. Max-Margin Classifier is used to find a plane that has the maximum distance between data points of both classes. To solve a non-linear problem in svm a "kernel" method is using as linear classifier. The possible hyperplane and maximum margin are shown in Fig. 1.

2.3 Multi-Layer Perceptron (MLP)

MLP is a class of feed-forward artificial neural network. The term MLP is used ambiguously, sometimes it is just refer to any feed-forward artificial neural network, sometimes it is strictly to refer to networks which are composed of multiple layers of perceptrons which having threshold activation. An MLP model has at

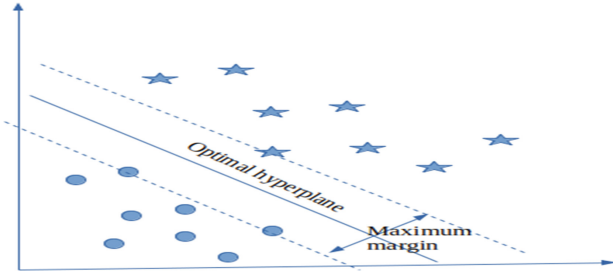


Fig. 1. Support vector machine.

least three layers of nodes: an input layer, a hidden layer and an output layer. Each node is a neuron excluding the input nodes that uses a non-linear activation function. For training purpose, MLP uses backpropagation supervised learning technique. It has non-linear activation function to differentiate from a linear perceptron. It can also differentiate non-linearly separable data.

2.4 Genetic Algorithm (GA)

GA is a search heuristic optimization technique based on the function of Natural Selection and Genetics. GA is mostly used in optimization problems. GA is commonly used to generate better solutions to search problems and optimization by using bio-inspired operators such as mutation, crossover and selection. The flow chart of GA is shown in Fig. 2.

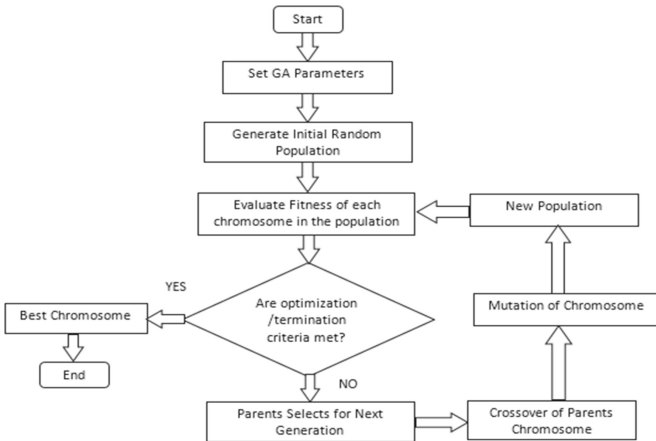


Fig. 2. Flow chart for genetic algorithm.

3 Literature Survey

In this section, we briefly review some of the literature available on software reliability prediction.

Jin and Jin [6] introduced a model hybrid of artificial neural network and Quantum Particle Swarm Optimization in software fault-proneness prediction. For classification, they used artificial neural network such that software modules are classified into faulty or non-faulty categories and for dimensionality reduction, Quantum Particle Swarm Optimization is applied. The results indicated that the presented model minimizes the effort and cost of software maintenance.

Hammouri et al. [3] introduced a model based on machine learning techniques for software bug prediction. They used three supervised machine learning algorithms on historical data to predict future software faults. The classifiers, they used namely Artificial neural network, Decision Trees and Naïve Bayes. As a result, they compared their models with other techniques and found their model gives better performance.

Twala [10] presented software faults prediction using an ensemble of classifiers. The ensemble classification model improves the classification performance. The result showed that 2 NASA public datasets give higher accuracy in comparison of individual classifiers. In other hand, boosting with associate rules and decision trees as constituent of an ensemble performance is better in software faults prediction.

Kamei et al. [7] investigated the prediction performance of an support vector machine model, and compared the performance of SVM with other most frequently used models including logistic regression, a classification tree, neural networks and linear discriminant analysis. The evaluation process indicated that the SVM model demonstrated the best performance among all other models.

Kumar and Singh [8] presented models using various ML techniques such as ANN, SVM, cascade correlation NN, DT, fuzzy interface system for prediction of software reliability. Using these techniques they estimated precision, correlation coefficient, mean absolute error and root mean squared error. The results showed that their models having better capability of generalization.

Chunhong and Licheng [1] introduced a model for automatic parameters selection by using GA for SVM. The estimated results of practice data model revealed the effectiveness and better efficiency of the proposed model.

Islam et al. [5] introduced a model which is implemented on a custom set of test data using SVM. It concentrated on a specific dataset behavior in predicting reliability. Concentrating on a particular dataset behavior, not on the development of an accurate model. They also built a SVM model to find the relationship between reliability and error rate.

Thant et al. [9] suggested a hybrid method, which combine Support Vector Machine-Radial Basis Function as base learner for Adaptive Boost, with the use of Minimum- Redundancy-Maximum-Relevance (MRMR) feature selection. The experimental results showed that hybrid approach with MRMR give better accuracy compared to SVM.

Iqbal and Aftab [4] develop a classification framework which uses Multi-Filter feature selection technique and Multi-Layer Perceptron (MLP) to predict defect prone software modules. According to results the proposed framework with class balancing technique performed well in all twelve NASA MDP datasets.

4 Proposed Methodology

In this section, we discuss our proposed methodologies. In Methodoly-I, we design the model without features selection using SVM and MLP techniques. In Methodoly-II, we construct another model with feature selection by GA using SVM and MLP methods.

4.1 Methodology-I: Software Reliability Prediction Using SVM/MLP

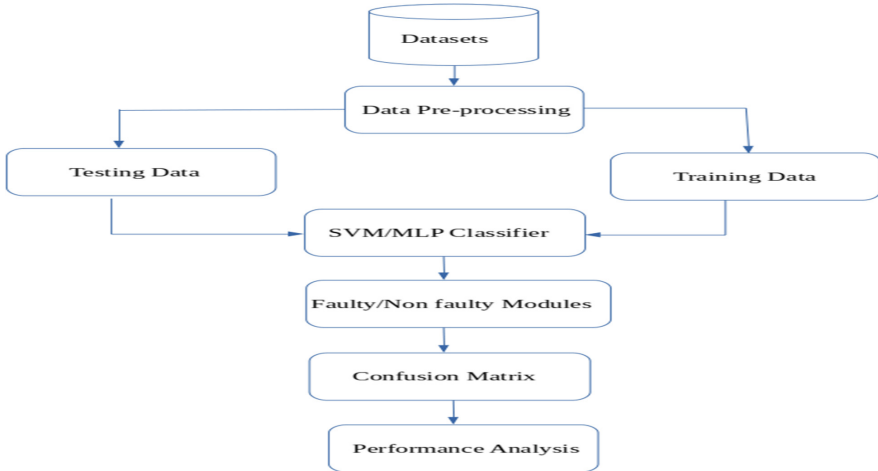


Fig. 3. Flow of proposed method1 (without features selection).

In this section, we proposed a method for software fault prediction using SVM/MLP. First, the input datasets are normalized using min-max normalization as shown in Eq. 1. Then, these datasets are given as input to the SVM. In this work, each dataset is splitted into two parts, training and testing data with 80% and 20% portion respectively. Then, we have trained the SVM model with RBF kernel and also we have trained MLP model with ten hidden layers using training part of dataset, and tested the SVM model with testing part of dataset. As a result, a confusion matrix is generated. From the confusion matrix, the performance measures such as accuracy, F-measure and AUC are calculated. The detailed process is shown in Fig. 3.

4.2 Methodology-II: Software Reliability Prediction Using SVM/MLP with Feature Selection by GA

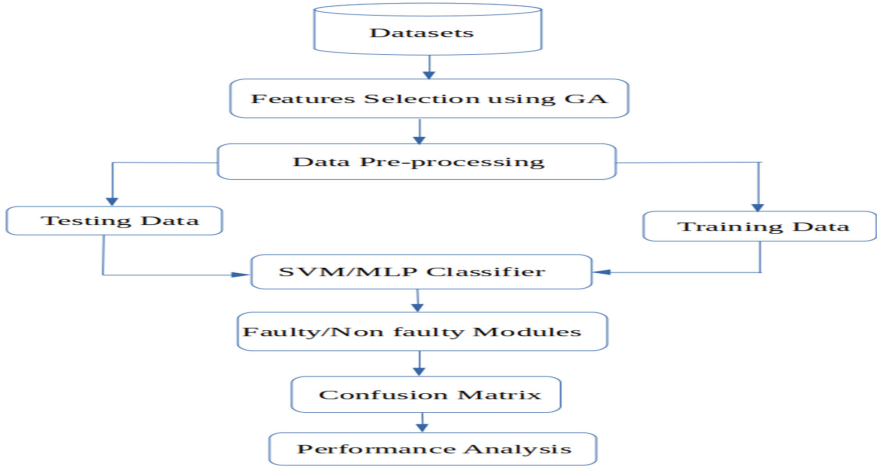


Fig. 4. Flow of proposed method2 (with features selection).

This method is similar to the method proposed in Sect. 4.1. But, here we predict the reliability of software after selecting the features, such as unique operand, halstead content etc. In this method, Genetic Algorithm is used to select the optimized features for classifying the faulty modules. The Genetic Algorithm procedure is explained in Fig. 2. The selected features are shown in Table 2. Then, the selected features from datasets are pre-processed using min-max normalization. After data normalization, we train the SVM/MLP techniques and get faulty and non-faulty modules. By the help of faulty and non-faulty modules, the results are generated in the form of confusion matrix and from the confusion matrix, the performance measures are calculated. The detailed process is shown in Fig. 4.

5 Implementation and Results

In this section, we briefly discuss the features of the dataset and what are the features selected by GA. Also, explain the evaluation metrics and results of our proposed models.

5.1 Datasets

We fetched the fault prediction datasets from Github Repository (Nasa Defect Dataset) [2]. These datasets are shown in Table 1.

Table 1. Software fault datasets

Sl. No.	Datasets	Number of Features	Total Number of Modules	Number of Faulty Modules	Number of Non-Faulty Modules
1	KC1	22	2109	326	1783
2	KC2	22	522	107	415
3	KC3	40	458	415	43
4	PC1	22	1109	77	1032
5	PC3	38	1077	943	134
6	JM1	22	10885	8779	2106

Table 2. Features selected by GA

Sl. No.	Datasets	Total no. of Selected Attributes	Selected attributes name
1	KC1	9	v(g), v, d, i, IOCode, IOComment, IOBlank, uniq_Opnd, branchCount
2	KC2	3	ev(g), b, uniq_Opnd
3.	KC3	11	LOC_BLANK, BRANCH_COUNT, LOC_CODE_AND_COMMENT, CYCLOMATIC_DENSITY, ESSENTIAL_COMPLEXITY, HALSTEAD_CONTENT, HALSTEAD_PROG_TIME, HALSTEAD_VOLUME, NUM_UNIQUE_OPERANDS, PERCENT_COMMENTS, LOC_TOTAL
4	PC1	6	v(g), I, IOComment, locCodeAndComment, IOBlank, uniq_Opnd
5	PC3	9	LOC_BLANK, LOC_COMMENTS, HALSTEAD_CONTENT, NORMALIZED_CYLOMATIC_COMPLEXITY, NUM_OPERANDS, NUM_UNIQUE_OPERANDS, NUMBER_OF_LINES, PERCENT_COMMENTS, LOC_TOTAL
6	JM1	8	loc, v(g), ev(g), iv(g), i, IOComment, IOBlank, locCodeAndComment

5.2 Evaluation Metrics

To measure performance, we use following metrics: confusion matrix, F-measure, Accuracy and Area under the curve (AUC). Confusion matrix shows the actual

Table 3. Accuracy(%) on different datasets

Sl. No.	Datasets	SVM	GA+SVM	MLP	GA+MLP
1	KC1	85.7	85.5	84.7	85.6
2	KC2	84.5	85.0	81.9	79.1
3	KC3	90.3	90.4	90.6	78.8
4	PC1	93.2	93.0	92.9	93.1
5	PC3	86.9	97.4	87.8	87.0
6	JM1	81.4	81.4	80.8	81.0

and predicted labels from a classification problem. Test’s accuracy is measured by F-Measure and is calculated as per Eq. 2

$$F - Measure = \frac{(2 * a * b)}{(a + b)} \tag{2}$$

Where, a is precision and b is Recall. Accuracy is measured as the ratio of correctly predicted data by the total data.

$$Accuracy = \frac{(TP + TN)}{TP + FP + TN + FN} \tag{3}$$

AUC is estimated as the overall performance of a classification model based on area under the ROC curve.

5.3 Results

The results of our proposed model are discussed in this section. The accuracy of the models for different datasets shown in Table 3. From Table 3, we can observe that, out of 6 datasets, for one dataset (KC3), MLP shows better performance and for rest of all datasets, GA+SVM and SVM, both have similar performance. The same thing is represented by bar chart in Fig. 5.

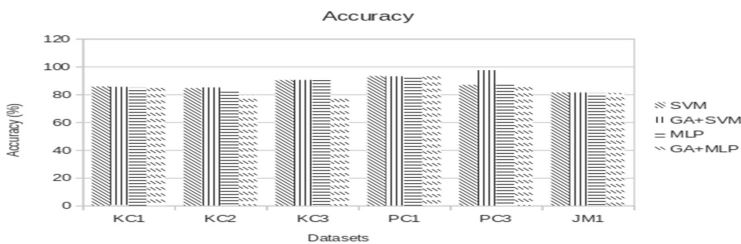


Fig. 5. Accuracy comparisons of our proposed models.

Table 4. F-Measure on different datasets

Sl. No.	Datasets	SVM	GA+SVM	MLP	GA+MLP
1	KC1	0.906	0.912	0.912	0.908
2	KC2	0.928	0.907	0.911	0.906
3	KC3	0.955	0.960	0.948	0.882
4	PC1	0.965	0.963	0.977	0.963
5	PC3	0.92	0.922	0.941	0.930
6	JM1	0.888	0.903	0.889	0.896

The F-Measure of the models for different datasets is shown in Table 4. From Table 4, it may be observed that, out of 6 datasets, for two datasets (JM1 and KC3), GA+SVM shows better performance, for KC2 dataset, SVM shows better performance, for KC1 dataset, GA+SVM and MLP, both have similar performance and for rest of all datasets, MLP shows better. The same thing is represented by a bar chart in Fig. 6. The AUC comparison for our models is shown in Table 5. From Table 5, we can observe that, for all datasets except PC1 and JM1, GA+MLP shows better performance, for PC1 dataset, GA+SVM shows better performance and for JM1 dataset, MLP shows better performance. The same thing is represented by a bar chart in Fig. 7.

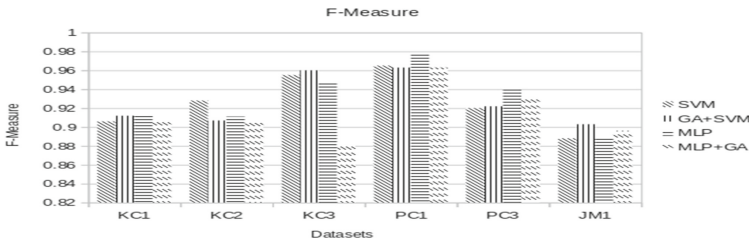


Fig. 6. F-Measure comparisons of our proposed models.

Table 5. AUC on different datasets

Sl. No.	Datasets	SVM	GA+SVM	MLP	GA+MLP
1	KC1	0.572	0.698	0.797	0.805
2	KC2	0.832	0.793	0.854	0.875
3	KC3	0.6823	0.709	0.776	0.821
4	PC1	0.73	0.932	0.843	0.798
5	PC3	0.75	0.689	0.824	0.741
6	JM1	0.631	0.618	0.729	0.705

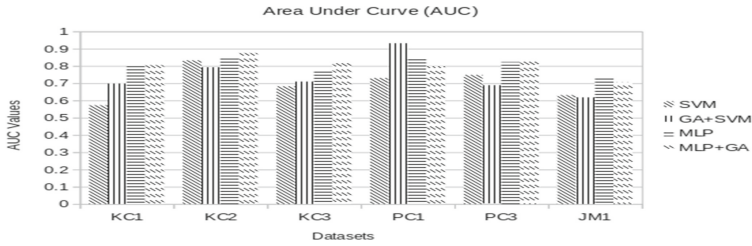


Fig. 7. AUC comparisons of proposed models.

Finally, we can conclude that, MLP classifier outperforms than SVM classifier. We also used feature selection technique, which has various characteristics such that to save the maintenance cost of the system, reduce the training time, reduces the complexity of the system and reduces overfitting problems.

6 Comparisons with Existing Model

In this section, we compare the performance of our proposed model with the performance of the model proposed by Jin and Jin [6]. We compare the performance by considering the AUC metric, on datasets. The comparison is shown in Table 6. From Table 6, we conclude that our proposed models show better performance for PC1 and KC1 datasets than the model proposed by Jin and Jin [6]. For KC3 and JM1 datasets, Jin and Jin [6]’s model performs better result.

Table 6. Comparisons of AUC

Sl. No.	Datasets	Jin and Jin model	Our proposed model
1	KC1	0.791	0.8048
2	KC3	0.862	0.8206
3	PC1	0.90	0.9321
4	JM1	0.777	0.729

7 Conclusion and Future Work

Software fault prediction models predict the fault-prone modules and hence increase the software reliability. The major importance of feature selection technique is to reduce the maintenance cost of the system and reduce the training time. In this paper, two software fault prediction models GA with SVM and GA with MLP are proposed. GA is applied to select the best features for fault prediction. After the selection of features by GA, SVM and MLP are applied to those selected features. Based on AUC metric, most of times, MLP with GA model outperforms the other models. Our future work will be to improve the performance using Random Forest and PSO, and test on more datasets.

References

1. Chunhong, Z., Licheng, J.: Automatic parameters selection for SVM based on GA. In: Fifth world congress on intelligent control and automation (IEEE Cat. No. 04EX788), vol. 2, pp. 1869–1872. IEEE (2004)
2. NASA Defect Dataset. <https://github.com/klainfo/NASADefectDataset/tree/master/OriginalData/MDP>
3. Hammouri, A., Hammad, M., Alnabhan, M., Alsarayrah, F.: Software bug prediction using machine learning approach. *Int. J. Adv. Comput. Sci. Appl.* **9**(2), 78–83 (2018)
4. Iqbal, A., Aftab, S.: A classification framework for software defect prediction using multi-filter feature selection technique and MLP. *Int. J. Mod. Educ. Comput. Sci.* **12**(1) (2020)
5. Islam, M.B., Islam, A.S., Hasan, M.K., Rahman, Z.: Predict software reliability by support vector machine. vol. 2, pp. 46–52 (2013)
6. Jin, C., Jin, S.-W.: Prediction approach of software fault-proneness based on hybrid artificial neural network and quantum particle swarm optimization. *Appl. Soft Comput.* **35**, 717–725 (2015)
7. Kamei, Y., Monden, A., Matsumoto, K.I.: Empirical evaluation of svm-based software reliability model. In: Proceedings 5th ACM-IEEE Int'l Symposium on Empirical Software Engineering (ISESE2006), vol. 2, pp. 39–41 (2006)
8. Kumar, P., Singh, Y.: An empirical study of software reliability prediction using machine learning techniques. *Int. J. Syst. Assur. Eng. Manag.* **3**(3), 194–208 (2012)
9. Thant, M.W., Aung, N.T.T.: Software defect prediction using hybrid approach. In: 2019 International Conference on Advanced Information Technologies (ICAIT), pp. 262–267. IEEE (2019)
10. Twala, B.: Software faults prediction using multiple classifiers. In: 2011 3rd International Conference on Computer Research and Development, vol. 4, pp. 504–510. IEEE (2011)



Fault Prediction Using Deep Neural Network

Avishikta Chatterjee^(✉), Kulamala Vinod Kumar,
and Durga Prasad Mohapatra

NIT Rourkela, Rourkela, Orissa, India

Abstract. Fault prediction is the major step in large industries where the complexity of the software is rising at an exponential rate. Assigning the proper damaging level of faults encountered in complex and large software, would help the developers to plan for fixing the faults. Traditional fault prediction studies mainly concentrate on designing hand-crafted features, which are input into machine learning classifiers to identify faulty modules. In this paper, we have developed a model based on deep learning techniques. Some of the extracted features from programs are used to train the Deep Neural Network and then other features are passed for testing. The proposed model is validated using open source Promise data repository. It is clear from the results that the performance of predicting faults by the proposed model is better than the existing models.

Keywords: Software fault prediction (SFP) · Deep Neural Network (DNN) · Fault-proneness

1 Introduction

Now-a-days reliability is an important issue, since software are most failure prone and complicated. As the faults in modules at the time of implementation of software is the main reasons of failures, we need to review the modules to improve the reliability and to perform testing to find bugs in modules [1]. Reviewing the modules manually is quite laborious and illogical. Though, the budget of software project is fixed, it is better to check the faulty modules first.

Software fault prediction (SFP) techniques [2–6] were proposed to predict the faults and decrease the software improvement cost [7]. Fault prediction techniques construct models to predict the faults in new instances of modules. SFP is the process of predicating the faulty modules in software. Many machine learning techniques had been applied for SFP, including Naive Bayes (NB) [8], Multilayer Perceptron (MLP) [9], Decision Tree (DT) [11] and K-Nearest Neighbors (KNN) [10].

Programs' semantic information can be efficiently used for bug detection. Recently, one of the most powerful techniques namely deep learning which is used to capture the complicated non-linear features. In this paper, we propose

an approach to predict software faults by using deep learning techniques. First, the features are selected using Principal Component Analysis (PCA) [12] to reduce the complexity of the model. Then, the selected features are normalized by using max-min normalization [13] method. Finally, the normalized features are passed through DNN-based model to predict the faults in the modules and compares the effectiveness of existing fault prediction approaches. We use F1-score, precision and recall to evaluate the performance of this method.

The rest of this paper is arranged as follows. Section 2 provides the contribution made by researchers for software fault prediction. Section 3 provides basic concepts on fault prediction and DNN. Section 4 describes our proposed approach to normalized features and leverage these learned features to detect faults. Section 5 shows the experimental setup and results. Section 6 compares the proposed approach with exiting approaches. We conclude this paper in Sect. 7.

2 Literature Review

One of the most important task in software industry is predicting faulty software modules by researchers and professionals. As a result of this, various efforts have been made for software fault prediction using varied methods.

Sankar et al. [8] applied Naïve Bayes and support vector machine classifiers to predict faults in software system. This model used the history of software data and have the potentiality to overcome the insufficiency in accuracy.

Gayathri et al. [9] discussed an enhanced multilayer perceptron neural network with fuzzy logic based bell-shaped function to detect fault-proneness prediction in software systems. They have applied this approach for NASA's Metrics Data Program (MDP).

Goyal et al. [10] developed a software fault prediction model based on KNN regression. The performance of the model is become unaffected with increasing number of interacting predictors. It has the capability to learn the complex target easily and quickly without losing any information.

Rathore et al. [11] applied decision tree regression to develop intra-releases prediction and inter-releases prediction for software system. For calculating the accuracy of the model, absolute error and relative error are used.

Wang et al. [7] developed a model based on deep belief network to automatically learn feature from token vectors and extracted the abstract syntax tree from the program. To pass the features into the network, the token vectors were converted into integers vectors. Finally, the performance of the model is evaluated by using precision, recall and F1-score.

3 Basic Concepts

This section describes the basic concepts of software fault prediction and deep neural network.

3.1 Fault Prediction

It is the process of predicting faults in a program, which can help developers to develop programs by checking the potentially faulty modules. In Fig. 1, shows a typical fault prediction process. In this figure we can see, the first step is to collect instances from software database and label as faulty or non-faulty. Fault prediction ensures reliability in large software. A file is considered as faulty if it has at least one bug otherwise clean. The next step is to normalized features of each file. The instances with the selected features are subsequently trained using DNN. Lastly, new instances are put to the trained classifier, where those are predicted to be faulty or not.

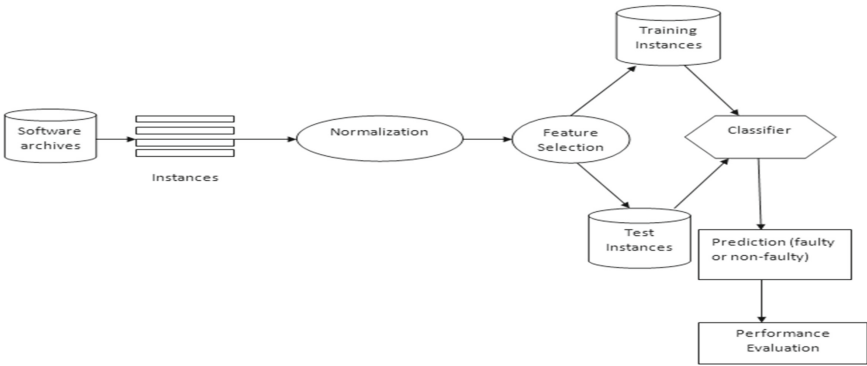


Fig. 1. Framework for fault prediction process

3.2 Deep Neural Network (DNN)

DNN is a neural network, having multilayer between one input and output layer. The topmost layer is the output layer and each layer is formed by several nodes. The number of hidden layers and nodes in the layers depend on users’ demand. It extracts higher level features from the input. The number of nodes in the topmost layer is based on the size of learned features.

The joint probability distribution between input layer and the hidden layers in the proposed model is as follows:

$$P(m, h^1, \dots, h^l) = P(m|h^1) \left(\prod_{x=1}^l P(h^x|h^{x+1}) \right) \tag{1}$$

where m is the number of nodes in input layer, l defines the number of hidden layers, and h^x is the data vector of node in x^{th} layer. $P(h^x|h^{x+1})$ is a conditional

probability distribution for the adjacent x and $(x + 1)$ layer, which is given as follows:

$$P(h^x|h^{x+1}) = \prod_{j=1}^{n_k} P(h_j^x|h^{x+1}) \quad (2)$$

where n_k is the number of node in layer k .

b is a bias matrix and W is the weight matrix. DNN learns W and b matrices automatically using an iteration process. W and b are updated by using the following equations:

$$W_{ij}(y + 1) = W_{ij}(y) + \eta \frac{(P(v|h))}{\delta W_{ij}} \quad (3)$$

$$b_x(y + 1) = b_x(y) + \eta \frac{(P(v|h))}{\delta b_x} \quad (4)$$

where y is the number of iterations and η is the learning rate of the model.

Initialize W via Restricted Boltzmann Machine and b is set to zero. W and b can be tuned with respect to some criterion such as error rate between original input data and reconstructed input data, training iterations. With these tuned weight and bias, DNN generates semantic features for both the test data and training data.

4 Proposed Approach

We have proposed an approach for predicting faults in software modules using DNN. In Fig. 2, we have described the workflow of our approach. Our approach takes datasets from the software archives as input, and then builds and evaluates the model for predicting the faults. Specifically, our approach extracts token vectors from the source modules of each file in both the test set and training set. We normalize the range of features and perform the feature selection using PCA to reduce the complexity of model. Finally, we build the model from the training datasets and evaluate the performance on the test datasets.

It consists of five major steps: 1) labeling, 2) normalization, 3) feature selection, 4) building prediction model, 5) performance evaluation.

4.1 Labeling

A group of data that is tagged with one or more labels is called labeled data. A set of unlabeled data is taken and embedded with significant tags those are informative. This process is called labeling. A primary step in enhancing computer vision model is to set a training algorithm and validate these models using high-quality training data. Data labeling is important because the machine learning algorithms have to understand the data. Based on this only, we will be able to train the model. After getting a labeled datasets, machine learning models are able to use these datasets for prediction.

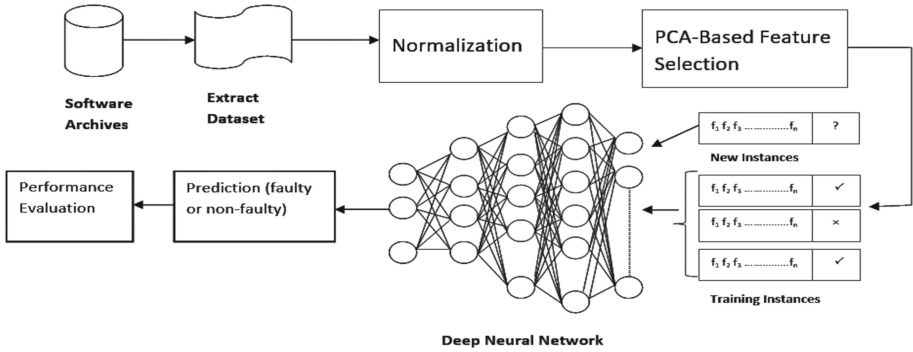


Fig. 2. Workflow of our proposed DNN-based fault prediction

4.2 Normalization

One or more attributes of the dataset are re-scaled within a small-specified range, such as 0.0 to 1.0, this process is called normalization. There are various kind of data normalization process such as max-min normalization, decimal scaling normalization and z-score normalization. In our proposed approach, we have used max-min normalization method. So, below we briefly explain the process of max-min normalization:

Max-Min Normalization: Max-min normalization [13] is carried out a linear transformation on the original data. It maps a value d of attribute A to d^h in the range $[nmin(a), nmax(a)]$. It is calculated by the following formula:

$$d^h = \frac{[d - min(a)] * [nmax(a) - nmin(a)]}{[max(a) - min(a)]} + nmin(a) \tag{5}$$

where $min(a)$ is the minimum value of attribute and $max(a)$ is the maximum value of attribute. In our case, the value d of A is to d^h in the range $[0,1]$, so $nmin(a) = 0$ and $nmax(a) = 1$. This normalization technique preserves the relationship among the original data values.

4.3 Feature Selection

The process of selecting features manually or automatically which helps in predicting faults more efficiently is called feature selection. It helps in filtering the redundant and irrelevant features from the datasets. One of the feature selection techniques, namely PCA-based feature selection [12] uses linear algebra to compress the dataset. This approach is applicable to both supervised and unsupervised classification.

This technique converts n vectors (Y_1, Y_2, \dots, Y_n) from a d -dimensional space to n vectors $(Y_1^h, Y_2^h, \dots, Y_n^h)$ in a new, d^h -dimensional space using the following equation:

$$Y_i^h = \sum_{k=1}^{d^h} a_k, e_k, d^h \leq d \quad (6)$$

where (e_k) is the eigen vector and (a_k) is the projections of the original vectors (Y_i) on the eigen vectors (e_k) . d and (d^h) must be positive integers, and the dimension (d^h) is always smaller than d . the scatter matrix S is defined using the following equation:

$$S = E[Y_i Y_i^T] \quad (7)$$

where $(E[Y_i Y_i^T])$ is the statistical expectation operator applied on the outer product of (Y_i) and its transpose. The variance of component (a_k) is given by

$$\sigma^2(e_k) = E[a_k^2] = e_k^T S e_k \quad (8)$$

It is clear that the variance of the component is the function of the value of the component of (e_k) .

4.4 Training DNN

To distinguish the faulty and clean modules, first we need to generate accurate semantic features. Then we train the DNN model by using the set of training data. To train DNN model we need to tune three parameters, these are: 1) total number of hidden layers, 2) total count of training iterations, 3) total number of nodes present in each hidden layer.

To make our model simple, the number of nodes are set to be same in each hidden layers. DNN is able to capture semantic differences of features through hidden layers and nodes. The input data are reconstructed by DNN through backpropagation validation by adjusting the weight among nodes. DNN learns probabilities to traverse from nodes of bottom layer to nodes of top layer.

We need to normalize data by max-min normalization [13] to get the range from 0 to 1 so that DNN is able to take the value as input data. In mapping process, the values of different tokens are used as identifiers. If two tokens have different mapping values that means two nodes are independent and different.

4.5 Performing Fault Prediction

After getting the features of each file in both the testing data and training data, we build a model and use it to train the features. Finally, we test the data to evaluate the performance.

5 Experimental Setup and Results

Several experiment is performed to study the semantic features and predict the faults in the modules. Then compare the outcomes with the existing models.

5.1 Evaluation Metrics

We have used four metrics such as accuracy, recall, precision and F1-score to estimate fault prediction results.

Accuracy: It is defined as the ratio of the number of correctly predicted files to the total number of files.

$$Accuracy = \frac{truepositive + truenegative}{truepositive + falsepositive + truenegative + falsenegative} \quad (9)$$

Precision: It is defined as the ratio of the number of correctly predicted files as faulty to the number of files predicted as faulty.

$$Precision : P = \frac{truepositive}{truepositive + falsepositive} \quad (10)$$

Recall: It is defined as the ratio of the number of correctly predicted files as faulty to the total number of files that are actually faulty.

$$Recall : R = \frac{truepositive}{truepositive + falsenegative} \quad (11)$$

F1-Score: F1-score depends on both recall and precision.

$$F1 - score = \frac{2 * P * R}{P + R} \quad (12)$$

5.2 Datasets

The datasets are taken from the PROMISE [14] data repository. We have taken 8 datasets such as ar1, ar3, ar4, ar5, ar6, mw1, mc2, jedit. We need the information about the number of average modules to feed the model. Each dataset contains a number of features, from these we have selected some effective features which are highly correlated to get the best result. We have given a brief description of each dataset in Table 1.

Table 1. Dataset description

Sl. No	Dataset	Description	Avg. modules	Faulty rate (%)
1	Ar1	“Embedded software in a while-goods product”	121	7.44
2	Ar3	“Embedded software in a while-goods product”	73	12.70
3	Ar4	“Embedded software in a while-goods product”	107	18.69
4	Ar5	“Embedded software in a while-goods product”	36	22.22
5	Ar6	“Embedded software in a while-goods product”	101	14.85
6	Jedit	“Text editor designed for programmers”	284	46.67
7	Mc2	“A video guidance system”	125	35.20
8	Mw1	“A zero gravity experiment related to combustion”	253	10.67

5.3 Result

In this section, we are going to discuss the results of the our experiment. We use tables to express the result and the comparison of effectiveness of our proposed DNN approach and the existing approaches for different programs.

We are going to evaluate the performances of our proposed model that is shown in Table 2. Here, we have taken four performance measures, such as: accuracy, precision, recall, F1-score.

Table 2. Performance evaluation

Sl. No	Dataset	Accuracy (%)	Precision	Recall	F1-score
1	ar1	92.56	0.92	0.93	0.92
2	ar3	87.30	0.92	0.92	0.89
3	ar4	81.31	0.81	0.91	0.85
4	ar5	77.78	0.85	0.83	0.86
5	ar6	87.13	0.86	0.94	0.90
6	jedit	97.77	0.96	0.97	0.96
7	mc2	70.07	0.72	0.85	0.78
8	mw1	89.33	0.89	0.92	0.91

6 Comparison with Related Work

Predicting faults in software modules is very important work in developing software for industries and organizations. There are various methods such as NB [8], MLP [9], KNN [10] and DT [11] are used for software faults prediction.

We compare the performance of our proposed DNN model for fault prediction with some baseline methods, such as NB, MLP, KNN and DT.

We present the comparison for precision, recall and F1-score of DNN based fault prediction model with above mentioned methods through the graphs shown in the following graphs. It is observed that DNN based model performs best among all the methods.

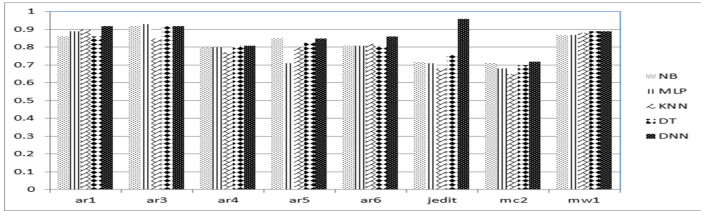


Fig. 3. Comparison of precision between DNN and other baseline methods

In Fig. 3, our approach has compared with NB, MLP, KNN and DT based models on the basis of precision. In average our DNN based proposed model performs better.

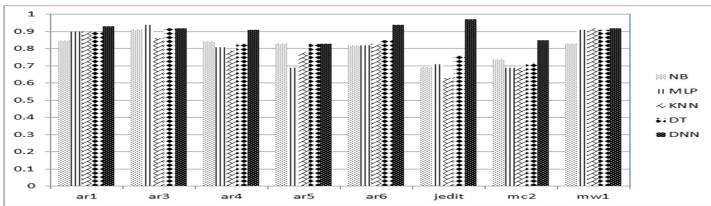


Fig. 4. Comparison of Recall between DNN and other baseline methods

In Fig. 4, our approach has compared with NB, MLP, KNN and DT based models on the basis of recall. In average our DNN based proposed model performs better.

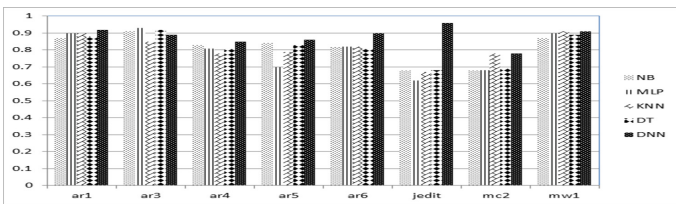


Fig. 5. Comparison of F1-Score between DNN and other baseline methods

In Fig. 5, our approach has compared with NB, MLP, KNN and DT based models on the basis of F1-score. In average our DNN based proposed model performs better.

7 Conclusion and Future Scope

Now-a-days the complexity of modern software is increasing so the reliability assurance becomes a great challenge. In this paper we have focused on predicting faults to enhance the reliability. It helps to maintain the software.

This approach presents deep learning algorithm to learn features from source modules to predict the faults. We deploy DNN model to predict the faults in software modules. We also evaluate the performance of our model using different matrices and compare by taking different datasets.

In future, we will use DNN on more projects for predicting faults. We will also apply other deep learning techniques for predicting software faults.

References

1. Li, J., He, P., Zhu, J., Lyu, M.R.: Software defect prediction via convolutional neural network. In: IEEE, Proceedings of the International Conference on Software Quality, Reliability and Security (2017)
2. Erturka, E., Sezer, E.A.: Iterative software fault prediction with a hybrid approach. *Appl. Soft Comput.* **49**, 1020–1033 (2016)
3. Singh, P., Pal, N.R., Verma, S., Vyas, O.P.: Fuzzy rule-based approach for software fault prediction. *IEEE Trans. Syst. Man Cybern.* **47**(5), 826–837 (2017)
4. Jing, X.Y., Ying, S., Zhang, Z.W., Wu, S.S., Liu, J.: Dictionary learning based software defect prediction. In: International Conference on Software Engineering, pp. 414–423 (2014)
5. Choudhary, G.R., Kumar, S., Kumar, K., Mishra, A., Catal, C.: Empirical analysis of change metrics for software fault prediction. *Comput. Electr. Eng.* **67**, 15–24 (2018)
6. Owhadi-Kareshk, M., Sedaghat, Y., Akbarzadeh, M.: Pre-training of an artificial neural network for software fault prediction. In: 7th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 223–228 (2017)
7. Wang, S., Liu, T., Tan, L.: Automatically learning semantic features for defect prediction. In: ICSE: Proceedings of the International Conference on Software Engineering (2016)
8. Sankar, K., Kannan, S., Jennifer, P.: Prediction of code fault using naive bayes and SVM classifiers. *Middle-East J. Sci. Res. ISSN* **1990–9233**, 108–113 (2014)
9. Gayathri, M., Sudha, A.: Software defect prediction system using multilayer perceptron neural network with data mining. *Int. J. Recent Technol. Eng. ISSN: 2277-3878*, **3**(2) (2014)
10. Goyal, R., Chandra, P., Singh, Y.: Suitability of KNN regression in the development of interaction based software fault prediction models. *Int. Conf. Future Softw. Eng. Multimedia Eng.* **6**, 15–21 (2013)
11. Rathore, S.S., Kumar, S.: A decision tree regression based approach for the number of software faults prediction. *ACM SIGSOFT Softw. Eng.* **41**, 1–6 (2016)

12. Malhi, A., Gao, X.R.: PCA-based feature selection scheme for machine defect classification. *IEEE Trans. Softw. Eng.* ISSN: 2277-3878, **3**(2) (2014)
13. Jain, Y.K., Bhandare, S.K.: Min-Max normalization based data perturbation method for privacy protection. In: *IJCCT*, ISSN: 0975-7449, **3**(4) (2014)
14. Fault Prediction Dataset. <http://tunedit.org/repo/PROMISE/DefectPrediction>. Accessed 2013



Head Pose Classification Based on Deep Convolution Networks

Sadia Afroze and Mohammed Moshiul Hoque^(✉)

Department of Computer Science and Engineering, Chittagong University
of Engineering and Technology, Chittagong 4349, Bangladesh
moshiul_240@cuet.ac.bd

Abstract. Recently, the classification of the head pose has gained incremented attention due to the rapid development of HCI/HRI interfaces. The resoluteness of head pose plays a considerable part in interpreting the person's focus of attention in human-robot or human-human inter-communications since it provides explicit information of his/her attentional target. This paper proposes a geometrical feature-based human head pose classification using deep convolution networks. An MTCNN framework is implemented to identify the human face and a ResNet50 layered architecture built to classify nine head poses. The system is trained with 2,85,000 and tested by 1,15,500 head pose images. The proposed system achieved 90.00% precision for nine head pose classes.

Keywords: Human-computer interaction · Computer vision · Head pose classification · Deep networks · Evaluation

1 Introduction

People have commonly been apt to deliver or collect information using a combination of verbal and nonverbal communication modes. Humans use various kinds of non-verbal cues (such as head motions, head orientation, eye gaze, head pose, hand gesture, and facial expression) during the interaction. Among these cues, understanding of head pose is vital to interpreting people's focus of visual attention, intention, and direction of interest. The goal of the head pose determination is to deduce the human's head direction/orientation from the face images. In recent years, it comes with a diversity of uses such as human behaviour understanding, motorist guidance systems, assisted technologies, gaze estimation systems, attention monitoring systems [12]. Application of head pose is an excellent form of communication that has considered a crucial link in human-computer and human-robot interactions. In recent times, in different marketing strategies and architectural designs are used, the head poses as a substitute to a personal interest in goods [6].

The head poses commonly represented by 03 rotation angles (pitch, yaw, and roll) those narrate the direction of head concerning the head-centred frame. Pitch and yaw are the most dominating angles in estimating the person's head

pose [1]. Our main objective is to develop an HCI framework that may estimate the focus of attention of a person while s/he is attending at multiple objects scenario [8]. Although the visual focus of attention may measure by eye gaze, head pose, body orientation cues or combination of them in our approach, we intend to use head pose due to larger areas of head covered by the camera from long distance. This paper deals with determining the head pose by utilizing a DCNN (deep convolutional neural network) technique. The proposed technique can classify the head pose into 09 classes namely, front frontal (FF), the front upside (FU), left frontal (LF), front down (FD), left down (LD), left up (LU), right up (RU), right frontal (RF), and right down (RD) respectively.

2 Related Work

Existing techniques of head pose estimation may classify broadly in two kinds: geometry-based and appearance-based [2]. To classify the head pose a multi-task learning framework is proposed by Yan et al. [4]. For classification of head pose, they used HOG descriptors to outline the appearance of the face. However, only the yaw angle considered for head orientation. Saeed et al. [5] proposed a frame-based technique to classify head poses based on Viola-Jones Haar-like features. SVM used to classify the head poses, which requires much training time on large data sets. In order to classify the head pose from the single image, a deep CNN based system is developed which used a multi-task framework for training [9]. Some approaches estimated the head pose based on depth data [10,11], and few are used both RGB and depth data [5]. Expected head poses value, and point-based geometric analysis developed for head pose tracking [7]. Vatahska et al. [12] proposed a feature-based head pose detection architecture that can identify face landmarks. Initially, they detect the head. Then they extract features using Haar-like feature extractor. Finally, the system feeds the extracted features into a neural network to classify the head into frontal, left and right poses respectively. In recent work, a combined method including CNN and deep learning, is used to classify the head pose [13]. Orozco et al. [14] extracted head pose features using the supervised descent method (SDM). SDM extracted facial appearance descriptors from the input image. The multi-class segmentation algorithm is developed to determine head pose [15]. The regression-based technique is proposed for head direction estimation [16], which is flexible in determining for the head pose. The MTCNN based face detector used for head rectangle cropping purpose [3] and classifier model trained with ResNet50 layer architecture for *nine* different head pose images [17].

3 Proposed Head Pose Classification Framework

The primary concern of the current research is to design a technique that can compute the head pose while the person is looking in different directions and classifies them into nine different classes. The head pose classification framework comprises of two key modules: face detection and head pose classification.

The face detection module detects the human face using a frontal camera. The detected face region use for the classification task. Figure 1 illustrates the abstract view of the proposed framework.

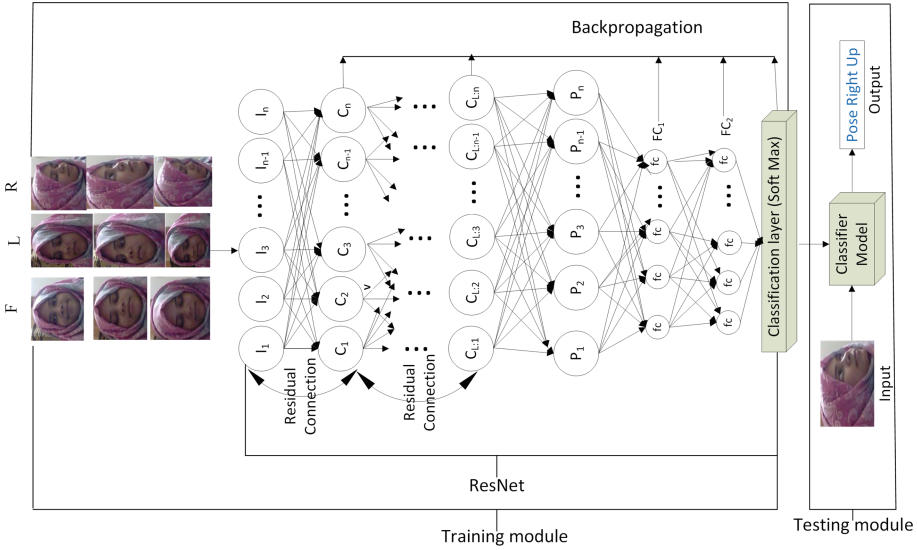


Fig. 1. Head pose classification framework with training and testing phases where the symbol F means front, L means left and R means right head poses respectively

3.1 Face Detection

A camera is fixed in the PC to capture the face of the human. An MTCNN face detection algorithm is used to select the head rectangle. MTCNN framework is consisting of 03 stages: P-net, R-net, and O-net [3]. P-net produces some candidate facial windows and bounding boxes with regression vectors which used to figure out the windows. This network merges the highly overlap windows using the non-maximum suppression (NMS) technique. The marked candidates are feed to the second stage (R-net). R-net also performs the same task as P-net and discarded a significant amount of false candidates. In the 3rd stage, O-net also performed the same task as R-net and identify face region with five facial points. Figure 2 represents the three stages of MTCNN face detection algorithm.

Intersection over union (IoU) operation is predicted the bounding box w. r. to the ground truth [Eq. 1].

$$IoU = \frac{B_r \cap O_t}{B_r \cup O_t} \tag{1}$$

Where B_r means the candidate bounding box and O_t represents ground truth bounding box.

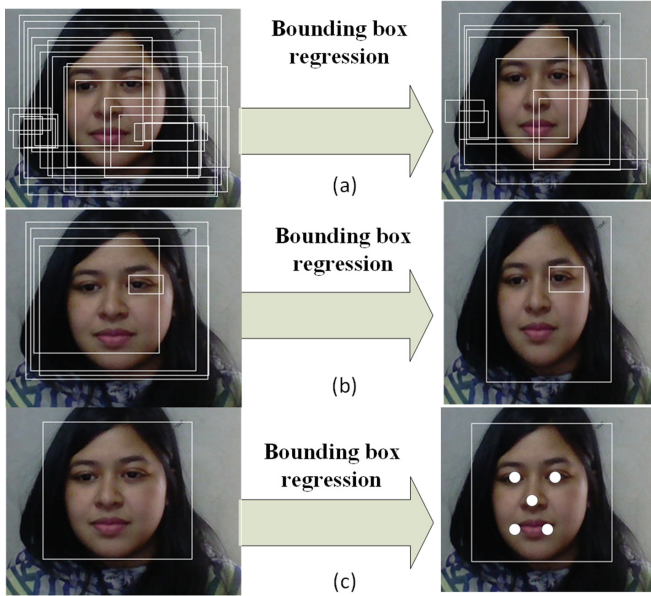


Fig. 2. MTCNN face detection composes of (a) P-Net (b) R-Net and (c) O-Net respectively

The MTCNN framework is performed the classification task as bounding box regression, non-face or face, and localization of facial landmark. During training, the bounding box regression model loss value is calculated by the Eq. 2.

$$Loss_i = -(q_i \times \log(p_i) + (1 - q_i) \times (1 - \log(p_i))) \tag{2}$$

Here, $Loss_i$ indicates the binary classification loss of face and non face. q_i means the ground truth label value. p_i means the probability value which produces by the network.

In the regression model generation time, the network used the euclidean distance for each sample f_i using Eq. 3.

$$Loss_i = \|\hat{q}_i - q_i\|_2^2 \tag{3}$$

Here, \hat{q}_i represents the estimated bounding box from the network and q_i represents the ground truth bounding box. MTCNN pre-trained model is used for face detection and the threshold value is modified based on the system environment. $\theta(N^3)$ simulation is needed to generate the threshold value. The simulation process is depicted in Algorithm 1. Initially, the accuracy is to zero and for each triplet, the accuracy updates according to the P-Net, R-Net and O-Net values. Function $evalMTCNN(i, j, k)$ is evaluated the face detection and head rectangle selection tasks. 1000 triplets are produced after the execution of Algorithm 1 and best triplet is considered to classify maximum face and non-face threshold.

Finally, this face rectangle is re-size to get the head rectangle which feeds to the classification module.

Algorithm 1: Thresholds generation for P-Net, R-Net and O-Net to classify face and non-face.

Result: To generate the best threshold value for P-Net, R-Net and O-Net
accuracy = 0.0, *Pthreshold* = 0.0, *Rthreshold* = 0.0, *Othreshold* = 0.0

```

for i ← 0 to 1 do
  for j ← 0 to 1 do
    for k ← 0 to 1 do
      ac = evalMTCNN(i, j, k)
      if accuracy ≥ ac then
        | Pthreshold = i, Rthreshold = j, Othreshold = k,
        | accuracy = ac
      end
      k ← k + 0.1
    end
    j ← j + 0.1
  end
  i ← i + 0.1
end

```

3.2 Head Pose Classification

The function of the classification module is to classify the head pose into nine classes. The head rectangle used as the input and the head pose is classified according to the head classifier model. The suggested technique used the ResNet50 [17] layer architecture for classifier model generation. The head pose classification technique is split into two stages: training and testing.

Training Phase. To train the classifier model, the proposed framework used nine different head pose images. These head pose images are build based on yaw (α) and pitch (β) angles of the head rotation. Nine different rules are designed for nine head poses by a set of empirical observations. For example, if α angle is $80 \leq (\alpha) \leq 110$ then front frontal (FF). The values of the angles were set by several empirical observations.

The ResNet50 architecture consists of five blocks in which each block contains batch normalization function (BN), convolution layer (CL), activation function (ReLU) and pooling function respectively. In each input image, the convolution computation is performed. The convolution layer uses a matrix filter and extracts the feature from the input image. Convolution is a mathematical operation that happens between an image matrix and filter to form a third matrix which is called feature map matrix or convoluted matrix. In a current input location $x(i, j)$ the convolution is performed as Eq. 4.

$$x(i, j) = \sum_{k=i, p=0}^{R=i+F_h, C=j+F_w} \sum_{l=j, q=0} x(k, l) \times W(p, q) \quad (4)$$

Equation 4, $x(i, j)$ indicates present location of input, R represents row, C represents columns of the weight matrix respectively. The parameters F , F_h and F_w indicates the weight filter, height, and width of the weight matrix respectively. 7×7 kernel size, 3 padding and 64 feature map (i.e., number of filters) are used in the convolution operation. The local features are reduced by convolution operation from inputs and feature map kernel is fine-tuned which produces outputs of 112×112 matrix. In BN process, the input layer is normalized to improve the internal covariate shift problem. It speeds up the training process and also reduces the computation cost. In the training time, the batch normalization layer performs mean (μ_b) and variance (σ_b) of the input layer by the Eqs. 5–6.

$$\mu_b = \frac{1}{Z} \sum_{i=1}^Z x_i \tag{5}$$

$$\sigma_b^2 = \frac{1}{Z} \sum_{i=1}^Z (x_i - \mu_b) \times (x_i - \mu_b) \tag{6}$$

where, Z means the batch size and x_i is input feature value. Finally BN layer normalize the input using the Eq. 7.

$$\hat{x}_p^i = \frac{x_p^i - \mu_b^i}{\sqrt{\sigma_b^2 + \epsilon}} \tag{7}$$

Thus, in every training step BN layer makes the input of every layer to have roughly the same distribution. Activation function ReLU is mapped in between 0 to 1 of the input layer at position $x(i, j)$ using the Eq. 8

$$x(i, j) = \max(x(i, j), 0) \tag{8}$$

After the activation layer, the pooling layer is conducted to reduce the high volume of data. Convolutional layer output is considered as an input to the pooling layer. An average pooling layer performs the reduce-sampling and split the input into a rectangular pooling area, and it computed the average values of the individual region $x(i, j)$ using the Eq. 9.

$$x(i, j) = \frac{1}{K_h \times K_w} \sum_{k=i}^{i+K_h} \sum_{l=j}^{j+K_w} x(k, l) \tag{9}$$

The output of pooling is smoothening into a value of a single vector which is represented with a probability while a certain feature belongs to a head pose class. The fully connected layer reduces the 1024 features. Every fully connected layer is densely concatenated to the next layer/output layer. A weight matrix consists of nine rows as nine different head poses and 1024 columns as a feature is generated in the output layer. This matrix is projected with the fully connected layer feature vector to produce the desired score. Finally, the network

calculates the deviation and backpropagate through the network to adjust the weight matrix.

The main reason to adopt ResNet 50 is that it removes vanishing gradient problem by skip connection. Moreover, batch normalization operation conducted in ResNet and this is absent in other network (such as VGG or Alexnet [18]). In the current approach, ResNet50 used in spite of ResNet18, ResNet101 and ResNet151 because ResNet50 provides optimal training and testing accuracy. ResNet101 and ResNet151 are good to produce higher training accuracy, but their testing accuracy is lower due to the overfitting problem. In the proposed system, the ResNet50 has gained the best accuracy (97.00%) concerning testing data sets.

Testing Phase. The input takes a video frame and passes through ResNet50 to classify head poses. The trained kernel is initialized by the ResNet50. The input frame is propagated through a layered architecture. The output layer (softmax) contains a weight matrix that has nine rows and 1024 feature columns. The fully connected layer produces a 1024 feature vector which projected by the output layer weight matrix.

Classification. The softmax layer generates a score for every class and the maximum score defines the corresponding head pose. The softmax layer calculates the p^{th} sample score using Eq. 10.

$$H_p = \frac{e^{Wm_p \times Fv_p}}{\sum_{i=1}^9 e^{Wm_i \times Fv_p}} \quad (10)$$

In the Eq. 10, H_p is desired score of p^{th} head pose. Wm_p is the weight vector of the softmax layer and Fv_p is the feature vector that got from a fully connected layer. The softmax layer gives the probability of each head pose class.

4 Experiments

The proposed framework is implemented using Python 3.5, seaborn, sklearn Pytorch 0.4.0, TensorFlow 1.2, OpenCV and CUDA 8.0 in our deployed environment. Core i7 processor with GTX 1070 GPU, 32 GB RAM is used for training. Approximately, about seventy two hours are needed to converge the classifier model with a learning rate of 0.0001 and 100 epoch. A frontal webcam (C525, Logitech) is captured the head motions of the participant. The webcam is placed at a distance of 5 m from the human.

4.1 Datasets

Head pose training samples, validation samples, and testing samples are collected from the university level students. The hand-annotated data sets are prepared

from the real-time video stream. Nine classes of head poses are considered: FF, FU, FD, LF, LU, LD, RF, RU and RD respectively. Based on the experimental setup, we have collected the video streams with 25 fps. Table 1 illustrates the summary of the hand-annotated head pose dataset. We also trained and tested

Table 1. Summary of the head pose classification dataset.

Head pose	Video duration (mins)	#Training frames	#Validation frames
FF (Front frontal)	40	40000	20000
FU (Front up)	30	30000	15000
FD (Front down)	25	25000	12500
LF (Left frontal)	30	35000	10000
LU (Left up)	25	30000	7500
LD (Left down)	20	25000	5000
RF (Right frontal)	40	40000	20000
RU (Right up)	30	30000	15000
RD (Right down)	27	30000	10500
Total	267	285000	115500

the proposed framework using the Pointing 04 dataset [19] which consists of 1953 training and 837 testing images. We annotate this dataset into nine classes to serve our purpose.

4.2 Evaluation Measures

The proposed system is evaluated during the training and testing phases. Losses and accuracy measures are used in training and validation phases. Precision (P), recall (R), F_1 -score and accuracy measures are used during testing phase. The training loss for p^{th} iterations are calculated using the Eq. 11.

$$loss_p = -\frac{1}{M} \sum_{i=1}^F \log\left(\frac{e^{W_i^T \times V_i}}{\sum_{j=1}^F e^{W_j^T \times V_j}}\right) \quad (11)$$

Where, $loss_p$ indicates p^{th} iteration loss, F indicates number of pose categories, W_i means weights and V_i means p^{th} category score value. The p^{th} iteration accuracy is measured by the Eq. 12. Where, $Accuracy_p$ indicates the p^{th} batch accuracy and $\#total_pose$ indicates the total number of head pose frames in the system.

$$Accuracy_p = \frac{\#correctly_predict}{\#total_pose} \quad (12)$$

5 Results

Figure 3(a) shows the graphical representation of validation and training losses. Due to the huge amount of data distribution, the initial loss of the training set is 45. At epoch number 1–75, the loss value decreases gradually. The training loss almost stable at epoch number 90–100 and the system is converged at 100 epoch. Validation loss starts from 11, and after some epoch, it decreases gradually. At epoch number 80–100 the loss value almost steady state. In Fig. 3(b) shows the validation and training accuracy. The training time validation accuracy represent by the red colour and training accuracy represent by the green colour. Training and validation accuracy almost steady at epoch 85–100, and it's stable for the proposed system.

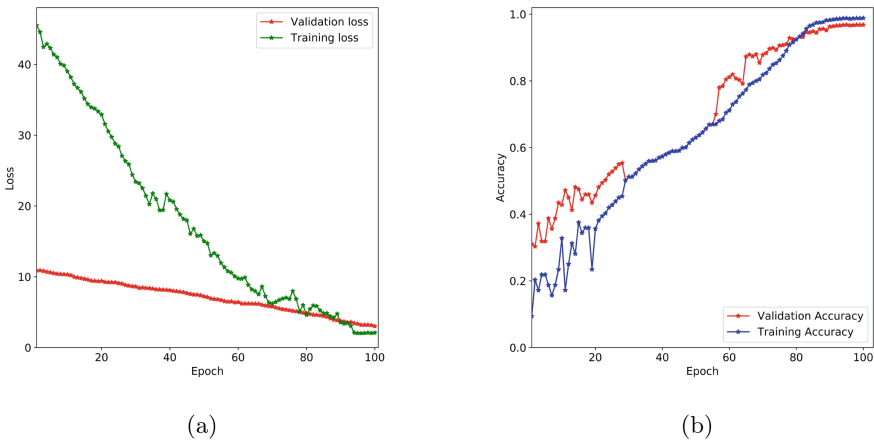


Fig. 3. Validation and training phase evaluation: (a) losses (b) accuracy

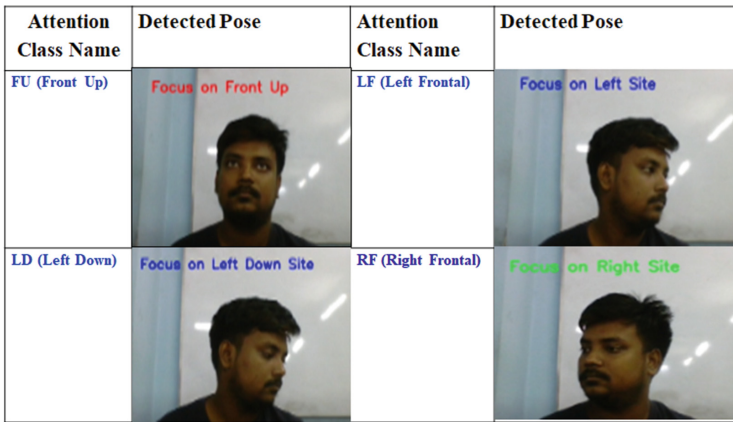
The evaluation results of the testing phase are represented in terms of statistical analysis. Table 2 summarizes the statistical evaluation in terms of P, R and F_1 measures.

Table 2 indicates that the front frontal pose gained more accuracy than other poses. Maximum accuracy gained by front frontal pose (93.00%) and minimum accuracy gained from left down and front down (88.00%). Figure 4 represents few snapshots of head pose classification. The overall accuracy of the suggested technique is 90.00% by using our hand-crafted dataset whereas the pointing'04 dataset provides 89.86% accuracy.

Comparison with Existing Approaches. To investigate the effectiveness of the suggested framework, a comparison with the existing approaches are performed. Table 3 indicates that random projection forest (RPF) based classification considered 6 different head poses which are achieved 72.00% accuracy. Head

Table 2. Evaluation statistics of the head pose classification system.

Pose category	# of Frames	Precision (P)	Recall (R)	F_1 - score
FF (Front frontal)	7000	0.94	0.92	0.93
FU (Front up)	6000	0.92	0.89	0.90
FD (Front down)	5350	0.91	0.86	0.88
LF (Left frontal)	5200	0.93	0.90	0.91
LU (Left up)	6797	0.91	0.92	0.91
LD (Left down)	6200	0.90	0.86	0.88
RF (Right frontal)	6150	0.92	0.93	0.92
RU (Right up)	7320	0.91	0.89	0.90
RD (Right down)	5250	0.91	0.89	0.89
Avg.	6141	0.92	0.89	0.90


Fig. 4. Sample recognition snapshots for the poses FU, LF, LD, and RF.

pose network (HPN) considered 8 different head poses and gained an accuracy of 74.00%. On the other hand, the proposed system is gained 90% accuracy with nine poses which are higher than the previous approaches.

Table 3. Performance comparison.

Methods	#Training frame	#Testing frame	#class	Classification accuracy (%)
RPF [20]	15000	2000	6	72.00
HPN [21]	4500	100	8	74.00
Proposed	285000	51767	9	90.00

6 Conclusion

Our primary purpose is to develop an HRI framework to assess the visual focus/direction of attention of the target person. To determine the attentional direction of the intended person, we concerned in using the head poses. This work represents a head pose classification framework based on CNN which can classify nine different head poses with 90.00% accuracy. It produces quite better results compared to the existing geometrical point-based classification methods. The performance of the proposed system can be enhanced by adding more head poses frames in the training dataset. Body orientation also included estimating the attentional focus where the detection head rectangle failed. These issues left for future improvements.

Acknowledgement. This work was supported by ICT Division, Ministry of Posts, Telecommunications and Information Technology, Bangladesh.

References

1. Geng, X., Xia, Y.: Head pose estimation based on multivariate label distribution. In: 2014 IEEE Conference on Computer Vision & Pattern Recognition, pp. 1837–1842. IEEE Press (2014)
2. Wu, S., Liang, J., Ho, J.: Head pose estimation and its application in TV viewers' behavior analysis. In: 2016 IEEE Canadian Conference on Electrical & Computer Engineering, pp. 1–6. IEEE Press (2016)
3. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **23**(10), 1499–1503 (2016)
4. Yan, Y., Ricci, E., Subramanian, R., Liu, G., Lanz, O., Sebe, N.: A multi-task learning framework for head pose estimation under target motion. *IEEE Tran. Pattern Anal. Mach. Intell.* **38**(6), 1070–1083 (2016)
5. Saeed, A., Al-Hamadi, A., Ghoneim, A.: Head pose estimation on top of Haar-like face detection: a study using the Kinect sensor. *Sensors* **15**(9), 20945–20966 (2015)
6. Djeraba, C., Lablack, A., Benabbas, Y.: Abnormal event detection. In: Djeraba, C., Lablack, A., Benabbas, Y. (eds.) *Multi-Modal User Interactions in Controlled Environments*, pp. 11–58. Springer, Boston (2010)
7. Afroze, S., Hoque, M.M.: Detection of human's focus of attention using head pose. In: *International Conference on Advanced Information and Communication Technology* (2016)
8. Afroze, S., Hoque, M.M.: Classification of attentional focus based on head pose in multi-object scenario. In: Vasant, P., Zelinka, I., Weber, G.W. (eds.) *Intelligent Computing & Optimization*, vol. 1072, pp. 349–360. Springer, Cham (2019)
9. Li, S., Chan, A.B.: 3D human pose estimation from monocular images with deep convolutional neural network. In: *Asian Conference on Computer Vision*, pp. 332–347. Springer (2015)
10. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Gool, L.V.: Random forests for real time 3D face analysis. *Int. J. Comput. Vis.* **101**(3), 437–458 (2013)
11. Borghi, G., Venturelli, M., Vezzani, R., Cucchiara, R.: Poseidon: face-from-depth for driver pose estimation. *Computer Vision & Pattern Recognition*. arxiv.org/abs/1611.10195 (2017)

12. Vatahska, T., Bennewitz, M., Behnke, S.: Feature-based head pose estimation from images. In: 7th IEEE-RAS International Conference on Humanoid Robots, pp. 330–335. IEEE Press (2007)
13. Patacchiola, M., Cangelosi, A.: Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pat. Rec.* **71**, 132–143 (2017)
14. Orozco, J., Gong, S., Xiang, T.: Head pose classification in crowded scenes. In: British Machine Vision Conference, pp. 1–11 (2009)
15. Khan, K., Mauro, M., Migliorati, P., Leonardi, R.: Head pose estimation through multi-class face segmentation. In: IEEE International Conference on Multimedia & Expo, pp. 175–180. IEEE Press (2017)
16. Hara, K., Chellappa, R.: Growing regression forests by classification: applications to object pose estimation. In: European Conference on Computer Vision, pp. 552–567. Springer (2014)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 770–778. IEEE Press (2016)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *Computer Vision and Pattern Recognition*. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) [cs.CV] (2015)
19. Gourier, N., Hall, D., Crowley, J.L.: Estimating face orientation from robust detection of salient facial structures. In: FG Net Workshop on Visual Observation of Deictic Gestures (2004)
20. Lee, D., Yang, M., Oh, S.: Fast and accurate head pose estimation via random projection forests. In: International Conference on Computer Vision, pp. 1958–1966. IEEE Press (2015)
21. Hasan, I., Tsesmelis, T., Galasso, F., Cristani, M., Del Bue, A., Cristani, M.: Tiny head pose classification by bodily cues. In: IEEE International Conference on Image Processing, pp. 2662–2666. IEEE Press (2017)



A Robust Multi-Server Two Factor Remote User Authentication Scheme Using Smartphone and Biometric

Hasan Muhammad Kafi¹ (✉), Md. Al-Hasan¹, Mohammad Hasan^{1,2},
and Md Mamunur Rashid¹

¹ Department of CSE, Bangladesh Army University of Science and Technology (BAUST),
Saidpur, Bangladesh

kafi@baust.edu.bd

² Department of CSE, Chittagong University of Engineering and Technology (CUET),
Chattogram, Bangladesh

Abstract. Recently, many multifactor remote user authentication schemes have emerged to cover-up the security weaknesses of single factor user authentication systems. Among these schemes, the two-factor multi-server user authentication scheme has drawn a considerable amount of attentions of researchers. Subsequently, many two-factor user authentication schemes for multi-server architecture have been introduced in recent past. However, they were unable to prevent the security vulnerabilities like password guessing attack, user impersonation attack, privileged insider attack, server masquerading attack, denial of service attack, replay attack, etc. To address these security deficiencies, we propose a robust two-factor remote user authentication scheme for multi-server architecture using smartphone and biometric that can prevent all the major security vulnerabilities shown by existing schemes. We use ProVerif to demonstrate that our scheme fulfills all the required security properties. Through a comprehensive heuristic security and performance analysis, we show that our proposed scheme can overcome drawbacks of existing systems.

Keywords: Two-factor · Multi-server · User authentication · Biometric · Smartphone

1 Introduction

The most popular and widely used password-based single factor user authentication system has drawn a considerable amount of attention of the security researchers as well as guys like hackers and crackers due to the lack of its ability to provide adequate security. As it is a matter of secrecy, a numerous attempt [1–4] have been taken by security researchers to protect passwords from being compromised. However, all those attempts go in vain due to the habit and practice of users. Most of them reuse same password in several accounts which causes domino effect [5]. According to [6], users have habits of choosing easy passwords as well as forgetting their passwords. These

habits of users give the attackers perfect opportunity to guess their passwords [7]. To encounter these issues, several methods like one-time passwords, password management software, phone aided schemes, graphical password schemes, cognitive authentication scheme, biometrics, hardware tokens, etc., have been proposed [8]. But, none of them meet up to the expectation of researchers.

In the following decade, many multi-factor remote user authentication schemes [9–23] have emerged to replace single factor remote user authentication systems. Among them, smartcard and password based two-factor remote user authentication protocols for multi-server architecture have drawn considerable amount of attentions of researchers [10–16]. However, all of them have several weaknesses.

In 2013, Pippel et al. [10] presented a robust authentication scheme based on smart-card for multi-server architecture. They showed that their scheme shows resistant against various known attacks like offline dictionary attack, replay attack, server masquerading attack, user impersonation attack, smartcard loss attack as well as provides session key security. In 2015, Li et al. [11] conducted a detail analysis on protocol presented in [10] and identified that it is unable to provide protection against user impersonation attack, insider attack as well as cannot ensure mutual authentication. They proposed an enhanced smartcard-based authentication scheme [11] for multi-server architecture and proved that their scheme can provide perfect forward secrecy, prevent stolen smartcard attack as well as offline password guessing attack and so on.

Srinivas et al. [12] have shown that Li et al.'s scheme [11] is unable to withstand perfect forward secrecy. Moreover, it is vulnerable against insider attack, denial-of-service attack, stolen smartcard attack, etc. They proposed an improved password and smartcard-based multi-server authentication scheme [12] which is self-verifiable and capable of providing security against vulnerabilities addressed by them.

In 2018, Xu et al. [14] showed that the scheme presented in [11] is unable to provide user un-traceability as well as insecure against user impersonation attack and password guessing attack. They also conducted cryptanalysis of Amin et al.'s scheme [13] and discovered that it is unable to provide perfect forward secrecy and user un-traceability. Furthermore, they claimed that it is vulnerable to offline password guessing attack and user impersonation attack. They presented an elliptic curve cryptography-based multi-server two-factor user authentication scheme [14] and proved that their protocol can prevent replay attack, man-in-the-middle attack and stolen smart card attack. They also claimed that their scheme can prevent offline password guessing attack, server impersonation attack, user impersonation attack, etc. They also showed that their scheme can provide perfect forward secrecy, user un-traceability as well as mutual authentication.

In this article, we present a two-factor user authentication protocol using smart phone and biometric that can resist all the above-mentioned attacks as well as has an efficient recovery system. Moreover, it is capable of providing mutual authentication as well as session key agreement.

In this work, we choose biometric key to exploit the following benefits: biometric keys cannot be copied, shared, lost or forgotten easily. Moreover, they cannot be guessed easily, prevent non-repudiation and are extremely hard to distribute or forge. Additionally, we use smartphone instead of smart card. Because, now-a-days, all most everyone uses smartphone and it is more powerful than smart card in terms of computation. Many

smartphones have built-in biometric module and can be utilized to imprint biometric template and generate biometric key.

The remainder of this paper is arranged as follows: Sect. 2 discusses proposed protocol briefly and Sect. 3 discusses security analysis of our proposed system. Section 4 presents performance evaluation of our protocol. Finally, we conclude in Sect. 5.

2 Proposed Scheme

Our proposed scheme consists of five phases. They are server registration phase, user registration phase, login and authentication phase, biometric key update phase and recovery phase.

Table 1. The notations used throughout this paper

Notations	Descriptions
U_i	i^{th} User
S_j	j^{th} Server
RC	Trusted registration center
A	The attacker or malicious adversary
Id_i	Identification of user U_i
SId_j	Identification of server S_j
$m_1 - m_6$	Random numbers
SC_i	Smart card of U_i
SP_i	Smart phone of U_i
Pw_i, Pw_i^*	Password and new password of U_i
BK_i	Biometric key of U_i
SK_{ij}	Session key belonging to U_i and S_j
$H(.)$	One-way collision resistance hash function
\parallel	The string concatenation operator
\oplus	The bitwise XOR operator
K_{rc}	AES key of RC
K_{sj}	AES key of S_j
$ENC(.)$	Encryption function for AES
$DEC(.)$	Decryption function for AES
MSK_{ij}	Master secret key for U_i
$RCONT$	Recovery contact of U_i
$T_1 - T_5$	Timestamps
ΔT	Time delay threshold

2.1 Server Registration

The S_j and RC perform following steps during server registration phase:

- S_j chooses a unique identity Sid_j . Then S_j sends it to the RC through a secure channel.
- If Sid_j is not registered, RC generates random numbers rn_1 , G_j and calculates $L_j = H(Sid_j || rn_1 || T_1)$. Here, T_1 is the current timestamp. RC sends $\{Sid_j, L_j, G_j\}$ to S_j through secure channel.
- S_j computes $EL_j = ENC(L_j, K_{sj})$ and $EG_j = ENC(G_j, K_{sj})$ respectively and stores them into the database. Afterwards, S_j sends an acknowledgement to RC via a secure channel.
- RC computes $EL_j^{rc} = ENC(L_j, K_{rc})$ and $EG_j^{rc} = ENC(G_j, K_{rc})$. It stores $\{Sid_j, EL_j^{rc}, EG_j^{rc}\}$ into its database.

2.2 User Registration

During this phase, U_i , S_j and RC perform the following steps:

- U_i chooses his/her unique identity Id_i , imprints his/her biometric template B_i into fuzzy extractor [24] to generate his/her biometric key BK_i , chooses a random number rn_2 as well as his/her recovery contact $RCONT_i$. He/she calculates $b_i = H(BK_i || rn_2)$ and sends $\{Id_i, S_j, b_i, RCONT_i\}$ to the RC via a secure channel.
- If Id_i is not registered, then, RC randomly generates rn_3 and computes $v_{sj} = rn_3 || b_i$. RC transmits $\{Id_i, Sid_j, v_{sj}\}$ to S_j through secure channel.
- S_j computes $L_j = DEC(EL_j, K_{sj})$, master secret key $MSK_{ij} = H(L_j || v_{sj})$ and $EMSK_{ij} = ENC(MSK_{ij}, K_{sj})$ and stores $\{Id_i, EMSK_{ij}\}$ into its database. Finally, S_j sends acknowledgement to RC via secure channel.
- When the acknowledgement arrives, RC sends $\{Sid_j, b_i, v_{ui}, G_j\}$ to U_i through a secure channel upon computing $L_j = DEC(EL_j^{rc}, K_{rc})$, $G_j = DEC(EG_j^{rc}, K_{rc})$ and $v_{ui} = L_j || rn_3$ respectively.
- U_i generates his/her biometric key BK_i and calculates $b_i^* = H(BK_i || rn_2)$. If $b_i^* = b_i$, then he/she computes $Ev_{ui} = ENC(v_{ui}, BK_i)$, $EG_j^{ui} = ENC(G_j, BK_i)$ and $ERN2_{ui} = ENC(rn_2, BK_i)$ respectively. Afterwards, U_i stores $\{Sid_j, Ev_{ui}, ERN2_{ui}, EG_j^{ui}\}$ into his/her smart phone's (SP_i) database. Then, he/she sends an acknowledgement to RC through secure channel.
- RC computes $ERCONT_i = ENC(RCONT_i, K_{rc})$, $Ev_{sj}^{rc} = ENC(v_{sj}, K_{rc})$, $Ev_{ui}^{rc} = ENC(v_{ui}, K_{rc})$ and $ERN3_{rc} = ENC(rn_3, K_{rc})$ respectively. Finally, RC stores $\{Id_i, Sid_j, ERCONT_i, Ev_{sj}^{rc}, Ev_{ui}^{rc}, ERN3_{rc}\}$ into its database.
- When any party terminates session or rejects any message, then all parties need to roll back into their previous state.

2.3 Login and Mutual Authentication

During this phase, U_i and S_j need to perform the following steps:

- U_i connects his/her smart phone with his/her computer. Then, he/she provides his/her Id_i and imprints his/her biometric template B_i at fuzzy extractor [24] to generate biometric key BK_i . After that, smart phone SP_i computes $G_j = DEC(EG_j^{ui}, BK_i)$, $v_{ui} = DEC(Ev_{ui}, BK_i)$, $rn_2 = DEC(ERN_{2ui}, BK_i)$, $b_i = H(BK_i || rn_2)$ and $MSK_{ij} = H(v_{ui} || b_i)$ respectively. He/she also provides current timestamp T_2 , generates a random number rn_4 and computes $UID_i = Id_i \oplus H(T_2 || rn_4)$, $MSG_1 = H(Id_i || SID_j || T_2 || rn_4 || MSK_{ij})$, $MSG_2 = H(SID_j || G_j) \oplus T_2$ and $MSG_3 = H(G_j || T_2) \oplus rn_4$ respectively. After completing the computation, SP_i sends $\{UID_i, MSG_1, MSG_2, MSG_3\}$ to S_j via public channel.
- At first S_j calculates $G_j = DEC(EG_j, K_{sj})$ and $T_2 = MSG_2 \oplus H(SID_j || G_j)$ respectively. Afterwards, S_j takes current timestamp T_3 and checks whether $|T_3 - T_2| \leq \Delta T$ holds or not. If it does not hold, then S_j terminates the session. Otherwise, it computes $rn_4 = MSG_3 \oplus H(G_j || T_2)$, $Id_i = UID_i \oplus H(T_2 || rn_4)$, $MSK_{ij} = DEC(EMSK_{ij}, K_{sj})$ and $MSG_1^* = H(Id_i || SID_j || T_2 || rn_4 || MSK_{ij})$. Then, it compares whether $MSG_1^* = MSG_1$ or not. If they are not equal, then S_j terminates the session. In case of equality, S_j chooses a random number rn_5 and calculates $MSG_4 = H(Id_i || SID_j || T_3 || rn_4 || rn_5 || MSK_{ij})$, $MSG_5 = H(SID_j || G_j || rn_4) \oplus T_3$ and $MSG_6 = H(G_j || T_3 || rn_4) \oplus rn_5$ respectively. Afterwards, S_j sends $\{MSG_4, MSG_5, MSG_6\}$ to U_i via public channel.
- SP_i of U_i computes $T_3 = MSG_5 \oplus H(SID_j || G_j || rn_4)$ and checks whether $|T_4 - T_3| \leq \Delta T$ is valid or not. Here, T_4 is user's current timestamp. If it is invalid, then SP_i terminates the session. Otherwise, it computes $rn_5 = MSG_6 \oplus H(G_j || T_3 || rn_4)$ and $MSG_4^* = H(Id_i || SID_j || T_3 || rn_4 || rn_5 || MSK_{ij})$ respectively. Afterwards, it checks whether $MSG_4^* = MSG_4$ or not. If they are not equal, then SP_i terminates the session. Else, S_j is authenticated by U_i . SP_i computes $MSG_7 = H(MSK_{ij} || T_4 || rn_4 || rn_5)$ and $MSG_8 = H(G_j || rn_4 || rn_5) \oplus T_4$ respectively and transmits $\{MSG_7, MSG_8\}$ to S_j .
- After receiving $\{MSG_7, MSG_8\}$, S_j calculates $T_4 = MSG_8 \oplus H(G_j || rn_4 || rn_5)$. It picks current timestamp T_5 and checks whether $|T_5 - T_4| \leq \Delta T$ is valid or not. If it is invalid, then S_j terminates the session. Afterwards, S_j computes $MSG_7^* = H(MSK_{ij} || T_4 || rn_4 || rn_5)$ and compares with MSG_7 . If it is not equal, then S_j terminates the session. Otherwise, U_i is authenticated by S_j .
- U_i and S_j both compute session key $SK_{ij} = H(rn_4 || rn_5 || T_2 || T_3 || T_4)$ for further secret communication.
- Every time, when S_j has to terminate session, it sets a counter to count number of login attempts made by U_i . If the counter crosses over a certain threshold, then S_j denies any login attempt from U_i and blocks the user until he/she recover his/her account.

2.4 Biometric Key Update

The following steps are performed by U_i , S_j and RC during biometric key update phase:

- U_i connects his/her smart phone to the computer, provides his/her unique identity Id_i , generates his/her old biometric key BK_i , new biometric key BK_i^{new} and

computes $rn_2 = DEC(ERN2_{ui}, BK_i)$, $b_i = H(BK_i || rn_2)$, $b_i^{new} = H(BK_i^{new} || rn_2)$, $v_{ui} = DEC(Ev_{ui}, BK_i)$, $G_j = DEC(EG_j^{ui}, BK_i)$ and $GV_{ij} = H(G_j || v_{ui})$ respectively. Afterwards, SP_i sends $\{Id_i, SId_i, GV_{ij}, b_i^{new}\}$ to RC .

- RC computes $G_j = DEC(EG_j^{rc}, K_{rc})$, $v_{ui} = DEC(Ev_{ui}^{rc}, K_{rc})$ and $GV_{ij}^* = H(G_j || v_{ui})$ respectively. If $GV_{ij}^* = GV_{ij}$, then it computes $rn_3 = DEC(ERN3_{rc}, K_{rc})$ and $v_{sj}^{new} = rn_3 || b_i^{new}$ respectively. RC transmits $\{Id_i, SId_j, v_{sj}^{new}\}$ to S_j through secure channel.
- S_j computes $L_j = DEC(EL_j, K_{sj})$, master secret key $MSK_{ij}^{new} = H(L_j || v_{sj}^{new})$ and $EMSK_{ij}^{new} = ENC(MSK_{ij}^{new}, K_{sj})$. Then, it replaces $EMSK_{ij}$ by $EMSK_{ij}^{new}$ into its database. Finally, S_j sends acknowledgement to RC via secure channel.
- When the acknowledgement of S_j arrives, RC sends a confirmation message $\{SId_j, b_i^{new}\}$ to U_i through a secure channel.
- U_i generates his/her biometric key BK_i^{new} and calculates $b_i^* = H(BK_i^{new} || rn_2)$. If $b_i^* = b_i^{new}$, then he/she computes $Ev_{ui}^{new} = ENC(v_{ui}, BK_i^{new})$, $EG_j^{ui-new} = ENC(G_j, BK_i^{new})$ and $ERN2_{ui}^{new} = ENC(rn_2, BK_i^{new})$ respectively. Afterwards, U_i deletes $\{SId_j, Ev_{ui}, ERN2_{ui}, EG_j^{ui}\}$ from his/her smart phone's (SP_i) database and stores $\{SId_j, Ev_{ui}^{new}, ERN2_{ui}^{new}, EG_j^{ui-new}\}$ into the database. Then, he/she sends an acknowledgement to RC through secure channel.
- RC computes $Ev_{sj}^{rc-new} = ENC(v_{sj}^{new}, K_{rc})$ and replaces Ev_{sj}^{rc} by Ev_{sj}^{rc-new} into its database.
- When any party terminates session or rejects any message, then all parties need to roll back into their previous state.

2.5 Recovery

When a user loses his/her smartphone or database of his/her smart phone is erased then he/she needs to go through recovery phase. During this phase, U_i , S_j and RC will perform the following steps:

- U_i provides his/her Id_i and sends a recovery request along with Id_i and SId_j to RC .
- If Id_i is registered with SId_j , RC generates a random number rn_6 and computes $RCONT_i = DEC(ERCONT_i, K_{rc})$. Then, it sends $\{SId_i, rn_6\}$ to the recovery contact $RCONT_i$ of U_i through secure channel.
- U_i imprints his/her biometric template B_i into fuzzy extractor [24] to generate his/her biometric key BK_i . He/she chooses a random number rn_7 and computes $b_i = H(BK_i || rn_7)$. After that, U_i sends $\{Id_i, S_j, b_i, rn_6\}$ to the RC via a secure channel.
- If rn_6 is valid, then RC confirms the validity of U_i . Afterwards, U_i , S_j and RC will follow the similar steps as user registration phase to complete the recovery.

3 Security Analysis of the Proposed Scheme

In this section we are going to demonstrate comprehensive heuristic security analysis of our proposed scheme. Moreover, the proof of authentication and secrecy properties

of the proposed protocol is demonstrated using ProVerif [25] which is a well-known cryptographic protocol verification tool.

3.1 Provides User Un-traceability and Anonymity

During login and authentication phase, Id_i is sent in terms of UID_i instead of sending directly. Moreover, the attacker “A” can compute Id_i if and only if he/she knows T_2 , rn_4 and G_j . However, G_j is protected by means of AES at both S_j and RC . Hence, our proposed scheme satisfies user un-traceability and anonymity requirement.

3.2 Prevents Stolen Smart-Phone Attack

The information from smartphone can be extracted by stealing or without stealing it. Nevertheless, “A” cannot use this information to conduct any attack because Ev_{ui} , EG_j^{ui} and $ERN_{2,ui}$ are encrypted by biometric key. Hence, our proposed scheme can prevent stolen smartphone attack.

3.3 Prevents Privileged Insider Attack

An attacker from inside may eavesdrop and collect registration message $\{Id_i, S_j, b_i, RCONT_i\}$. Yet, he/she unable get the biometric key BK_i because BK_i in b_i is protected by random number rn_2 as well as one-way collision resistance hash function $H(\cdot)$. Therefore, our proposed scheme is resistant to privileged insider attacker.

3.4 Prevents User Impersonation Attack

It is implausible for an attacker “A” to impersonate as a valid user because, to send login message $\{UID_i, MSG_1, MSG_2, MSG_3\}$, he/she needs to generate BK_i as well as decrypt EG_j^{ui} , Ev_{ui} and $ERN_{2,ui}$ which is implausible. Hence, proposed scheme is immune to user impersonation attack.

3.5 Resists Server Masquerading Attack

The attacker “A” is unable to launch server masquerading attack because, to generate message $\{MSG_4, MSG_5, MSG_6\}$, he/she needs knowledge about MSK_{ij} and G_j . But, MSK_{ij} , components need to compute MSK_{ij} as well as G_j is kept encrypted. Hence, proposed scheme is able to prevent server masquerading attack.

3.6 Prevents Replay Attack

An attacker “A” can use old messages to conduct replay attack. However, we use timestamp mechanism to prevent this. The difference between old timestamp and current timestamp will way over threshold ΔT . Hence, old messages cannot pass the timestamp verifications in server side. Moreover, random number rn_5 in old mutual authentication message $\{MSG_7, MSG_8\}$ is different than the random number rn_5 currently used by S_j in reply message $\{MSG_4, MSG_5, MSG_6\}$. Subsequently, “A” will fail to authenticate by S_j . So, we can claim that our proposed scheme can resist replay attack.

3.7 Provides Known Key Security

It is worth to mention that the security of SK_{ij} is ensured by one-way collision resistance hash function $H(\cdot)$, random numbers and timestamps. Also, every SK_{ij} is unique. Hence, if somehow an attacker “A” manages to collect previously used session key, he/she cannot use it in communication or generating new session key.

3.8 Prevents Denial of Service Attack

If any scheme uses biometric template directly, there exists a possibility that it may fail matching with stored template. This is due to different orientation in imprinting biometric by user, presence of noise, etc. In our proposed scheme, we use an efficient way to generate biometric key from reasonably close biometric template [24]. In this way our scheme prevents denial of service attack.

3.9 Formal Verification with ProVerif

ProVerif [25] is an automatic cryptographic protocol verifier to prove secrecy and authentication properties of security protocols. The result of executing our proposed scheme in ProVerif 2.0 is shown in Fig. 1. The result shows that our proposed scheme confirms reachability property, session key secrecy as well as mutual authentication.

<pre>-- Query inj- event(userAuthenticate(server_43)) ==> inj- event(serverAuthenticate(user_44)) ok, secrecy assumption verified: fact unreachable attacker(BKi[]) ok, secrecy assumption verified: fact unreachable attacker(Ks[]) Starting query inj- event(userAuthenticate(server_43)) ==> inj- event(serverAuthenticate(user_44)) RESULT inj- event(userAuthenticate(server_43)) ==> inj- event(serverAuthenticate(user_44)) is true. -- Query not attacker(USKij[])</pre>	<pre>ok, secrecy assumption verified: fact unreachable attacker(BKi[]) ok, secrecy assumption verified: fact unreachable attacker(Ks[]) Starting query not attacker(USKij[]) RESULT not attacker(USKij[]) is true. -- Query not attacker(SSKij[]) ok, secrecy assumption verified: fact unreachable attacker(BKi[]) ok, secrecy assumption verified: fact unreachable attacker(Ks[]) Starting query not attacker(SSKij[]) RESULT not attacker(SSKij[]) is true.</pre>
---	---

Fig. 1. The result of execution of the proposed scheme in ProVerif 2.0.

4 Performance Evaluation

In this section we are going to demonstrate comparison of our proposed scheme with few other similar schemes in terms of security features and functionalities theoretically. We also present the efficiency comparison in terms of computational cost.

Table 2. Comparison of security properties and functionalities

Properties	S ₁	S ₂	S ₃	S ₄	S ₅
Provides User Un-Traceability and Anonymity	0	0	0	1	1
Prevents Stolen Smart-Phone Attack	-	-	-	-	1
Prevents Stolen Smart-Card Attack	0	0	0	0	-
Prevents Password Guessing Attack	0	0	0	0	-
Security of Server Database	0	0	0	0	1
Prevents Privileged Insider Attack	0	1	1	1	1
Resists User Impersonation Attack	0	0	0	0	1
Resists Server Masquerading Attack	0	1	1	0	1
Prevents Replay Attack	1	1	1	1	1
Provides Known Key Security	1	1	1	1	1
Provides Mutual Authentication	0	0	0	0	1
Resists Man-in-the-Middle Attack	0	1	1	0	1
Prevents Denial of Service Attack	-	-	-	-	1
Perfect Forward Secrecy	1	1	0	0	1
Recovery Phase	0	0	0	0	1

S₁ = Li et al.'s scheme [11], S₂ = Srinivas et al.'s scheme [12], S₃ = Amin et al.'s scheme [13], S₄ = Xu et al.'s scheme [14], S₅ = Our proposed scheme. 1 = Yes, 0 = No, - = N/A.

4.1 Comparison of Security Features and Functionalities

Comparison among the proposed scheme and few of the recently proposed relevant schemes in terms of security features and functionalities are summarized in Table 2.

4.2 Comparison of Computational Cost

Table 3 demonstrates the comparison among our proposed scheme and few of the similar schemes in terms of computational cost. Our focus is only on login and authentication phase. The XOR operation and string concatenation operation can be ignored because the computation cost of these two are trivial when comparing with other operations. However, the computation cost of point multiplication operation, modular exponential operation, hash function and AES decryption operation must be considered. It is worth to pointing that AES decryption operation is as efficient as hash function in terms of computational cost. Moreover, execution time of point multiplication operation and modular exponential operation is way greater than the execution time of AES decryption operation and hash function. Therefore, from Table 3, we can say that our proposed scheme is more time efficient than other existing schemes.

Table 3. Comparison of computational cost

Schemes	User-side computation	Server-side computation
S ₁	$T_{me} + 5T_h$	$3T_{me} + 8T_h$
S ₂	$2T_{me} + 8T_h$	$2T_{me} + 4T_h$
S ₃	$T_{me} + 6T_h$	$T_{me} + 4T_h$
S ₄	$3T_{mp} + 9T_h$	$3T_{mp} + 6T_h$
S ₅	$3T_d + 11T_h$	$2T_d + 10T_h$

S₁ = Li et al.'s scheme [11], S₂ = Srinivas et al.'s scheme [12], S₃ = Amin et al.'s scheme [13], S₄ = Xu et al.'s scheme [14], S₅ = Our proposed scheme. T_{me} = Execution time of modular exponential operation, T_{mp} = Execution time of point multiplication operation, T_h = Execution time of hash function, T_d = Execution time of AES decryption operation.

5 Conclusion

In this article, we have presented a robust multi-server based two-factor user authentication scheme using biometric and smartphone. Through formal analysis using ProVerif, we have demonstrated that the scheme proposed by us satisfies all the security requirements. Furthermore, we have depicted a comprehensive heuristic security analysis and proved that our proposed scheme prevents all the major security attacks. It uses the strength of AES encryption algorithm to ensure the security of valuable data. Moreover, it uses efficient and secure biometric key release algorithm to generate biometric key which is immune to biometric template matching issues. Our scheme also has an efficient recovery phase to take care of smart phone lost and database corruption issues. Finally, through security features and functionalities comparison as well as efficiency comparison in terms of computational cost, we have shown that our presented scheme outperforms all other existing schemes and clearly has the security advantages over those schemes.

References

1. Lamport, L.: Password authentication with insecure communication. *Commun. ACM* **24**, 770–772 (1981)
2. Morris, R., Thompson, K.: Password security: a case history. *Commun. ACM* **22**, 594–597 (1979)
3. Bellare, S.M., Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: *Proceedings of the Symposium on Security and Privacy*, pp. 72–84. IEEE (1992). <https://doi.org/10.1109/risp.1992.213269>
4. Spafford, E.H.: OPUS: preventing weak password choices. *Comput. Secur.* **11**, 273–278 (1992)
5. Ives, B., Walsh, K.R., Schneider, H.: The domino effect of password reuse. *Commun. ACM* **47**, 75–78 (2004)

6. Florencio, D., Herley, C.: A large-scale study of web password habits. In: 16th International World Wide Web Conference, WWW 2007, pp. 657–666 (2007). <https://doi.org/10.1145/1242572.1242661>
7. Han, W., Li, Z., Ni, M., Gu, G., Xu, W.: Shadow attacks based on password reuses: a quantitative empirical analysis. *IEEE Trans. Dependable Secure Comput.* **15**, 309–320 (2018)
8. Bonneau, J., Herley, C., Van Oorschot, P.C., Stajano, F.: The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In: Proceedings - IEEE Symposium on Security and Privacy, pp. 553–567. Institute of Electrical and Electronics Engineers Inc. (2012). <https://doi.org/10.1109/SP.2012.44>
9. Chang, I.P., Lee, T.F., Lin, T.H., Liu, C.M.: Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. *Sensors (Switzerland)* **15**, 29841–29854 (2015)
10. Pippal, R.S., Jaidhar, C.D., Tapaswi, S.: Robust smart card authentication scheme for multi-server architecture. *Wirel. Pers. Commun.* **72**, 729–745 (2013)
11. Li, X., Niu, J., Kumari, S., Liao, J., Liang, W.: An enhancement of a smart card authentication scheme for multi-server architecture. *Wirel. Pers. Commun.* **80**(1), 175–192 (2015). <https://doi.org/10.1007/s11277-014-2002-x>
12. Srinivas, J., Mukhopadhyay, S., Mishra, D.: A self-verifiable password based authentication scheme for multi-server architecture using smart card. *Wirel. Pers. Commun.* **96**, 6273–6297 (2017)
13. Amin, R., Maitra, T., Giri, D., Srivastava, P.D.: Cryptanalysis and improvement of an RSA based remote user authentication scheme using smart card. *Wirel. Pers. Commun.* **96**, 4629–4659 (2017)
14. Xu, G., et al.: A multi-server two-factor authentication scheme with un-traceability using elliptic curve cryptography. *Sensors* **18**(7), 2394 (2018). <https://doi.org/10.3390/s18072394>
15. Yu, H., Wang, L.: A security-enhanced mutual authentication scheme with privacy protected in wireless sensor networks. *Cluster Comput.* **22**, 7389–7399 (2019)
16. Li, W., Wang, P.: Two-factor authentication in industrial Internet-of-Things: attacks, evaluation and new construction. *Future Gener. Comput. Syst.* **101**, 694–708 (2019)
17. Banerjee, S., Chunka, C., Sen, S., et al.: An enhanced and secure biometric based user authentication scheme in wireless sensor networks using smart cards. *Wirel. Pers. Commun.* **107**, 243–270 (2019)
18. Li, C.T., Hwang, M.S.: An efficient biometrics-based remote user authentication scheme using smart cards. *J. Netw. Comput. Appl.* **33**, 1–5 (2010)
19. Khan, M.K., Zhang, J., Wang, X.: Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices. *Chaos Solitons Fractals* **35**, 519–524 (2008)
20. Lin, C.H., Lai, Y.Y.: A flexible biometrics remote user authentication scheme. *Comput. Stan. Interfaces* **27**, 19–23 (2004)
21. Das, A.K.: Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. *IET Inf. Secur.* **5**, 145–151 (2011)
22. An, Y.: Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards. *J. Biomed. Biotechnol.* **2012**, 1–6 (2012). <https://doi.org/10.1155/2012/519723>
23. Renuka, K., Kumari, S., Li, X.: Design of a secure three-factor authentication scheme for smart healthcare. *J. Med. Syst.* **43**, 133 (2019)
24. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **38**, 97–139 (2008)
25. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: IEEE Computer Society Foundation Workshop (CSFW), pp. 82–96 (2001)



Packet Error Probability Model for IEEE 802.15.6 MAC Protocol in Wireless Body Area Network

M. Ambigavathi^(✉)  and D. Sridharan

Anna University, CEG Campus, Chennai, India
sridhar@annauniv.edu

Abstract. Packet error in IEEE 802.15.6 MAC protocol is one of the main sources that degrades the performance and its inconsistency. This protocol does not have the ability to control the packet errors which is caused by the transmission failures or collisions. To alleviate the issues presented in the traditional protocol, this paper aims to develop an analytical model (p_r) to study the effects of packet errors and validate the performance of the IEEE 802.15.6 MAC protocol using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. The p_r -model decreases the packet error rate by incorporating the optimum number of Retry limit (R_l) and Contention Window (CW) size for all User Priority (UP) nodes. Therefore, this model can be deliberated as the first-ever analytical model for optimizing the UP and R_l assignment in order to improve the overall performance of the Wireless Body Area Network system. The simulation results suggest that minimizing transmission failure probability has more effect on saturation throughput than multiplying the CW size after even number of failures.

Keywords: Wireless body area network · Packet error · IEEE 802.15.6 MAC protocol · User priority · Retry limit

1 Introduction

Digital and sensor technologies are constantly evolving and reshaping the healthcare industry in several ways. Wireless Body Sensor Network is a branch of sensor technology that plays an integral role in providing remote healthcare monitoring applications to monitor the vital parameters of the human body through smart sensors and transfer the sensed data to the hub using a wireless medium [1, 2]. In such a network, efficient transmission of time-critical data with less packet error is the most significant requirement due to strict delay constraints. Several analytical studies have been conducted [4, 5] to evaluate the performance of the IEEE 802.15.6 MAC protocol using the CSMA/CA technique. This CSMA/CA technique uses a random backoff mechanism to avoid the packet collisions among the sensor nodes. Whenever a sensor node receives a data packet, then immediately picks its backoff counter value over the interval

$[1, CW]$, where, $CW \in [CW_{max}, CW_{min}]$. If the medium is idle, then the sensor node decrements its backoff counter value by one for each idle slot, when it reaches zero, the data frame is successfully transmitted. Initially, the size is set to each User Priority (UP) node and it does not change the value for each successful frame transmission. If the medium is still busy due to current transmission, then each node doubles its value for even number of channel attempt failures and remains unchanged its value for an odd number of channel attempt failures. Once the value exceeds the value, then node sets it's to in the ongoing transmission. Furthermore, sensor node will start to freeze the current backoff counter value, if there is any transmission on the channel during pCCATime and unfreeze its backoff value, when the channel is sensed to be idle for each pSIFS period. Sensor nodes can also freeze their backoff values when they are not allowed to access the channel in a given access period or the current access length is not long enough for completing the frame transmission process [3].

Most of the research studies considered infinite retry limits (R_l) for all UPs during the data transmission phase. Since there is no specific R_l fixed for UP based on the urgency of the data traffic. Even though the sensor node can successfully transmits its data packet based on the predefined UPs as it may drop several data packets when it experiences with the same size and R_l in the ongoing transmission. Therefore, the assignment of CW size and R_l is a key factor for successful frame transmission. If the assignment of these values is not perfect, sensor nodes cannot able to transmit their data packets on time even if they receive the time-critical data packets. Many research studies have been validated the performance of IEEE 802.15.6 MAC protocol using CSMA/CA mechanism but they failed to consider the probability of packet errors experienced by the sensor nodes during the backoff phases. This study inspects the effect of packet errors on the performance of the IEEE 802.15.6 MAC protocol which is due to other than packet collisions. Table 1 indicates the minimum and maximum CW values of different UP nodes.

The rest of this paper is organized into the following sections. Section 2 briefly examines the existing analytical models with respect to packet errors. The motivation and key contribution of this analytical study are summarized in Sect. 3. p_r -model is described in detail in Sect. 4. Simulation results of p_r -model is presented in Sect. 5 and the paper is concluded in Sect. 6.

Table 1. CW_{min} and CW_{max} Values for Different UPs with B_{cf}

UP	Traffic class	Data type	R_l	CW boundary values		Backoff value conflicts	
				CW_{min}	CW_{max}	Inter B_{cf}	Intra B_{cf}
UP_7	T1	Emergency	2	1	1	No	No
UP_6	T2	Critical	3	2	4	No	YES
UP_5	T3	Normal	4	4	8	No	YES

2 Motivations and Key Contributions

By analyzing the causes and effects of packet errors in the existing IEEE 802.15.6 MAC protocol, we observe that many retransmissions occur due to packet errors and collisions. If the value of R_l is infinite that can increase the probability of the data transmissions but it reflects the variability of packet errors. In case, if R_l value is finite for all UPs, then all nodes have an equal chance of retransmitting the collided packets until the transmission succeeds. Besides, each UP node increases its minimum size value for even number of channel attempt failures that may result in higher backoff delay, backoff value conflicts and packet errors. Thus, both R_l and CW values are the key contributor to reduce the probability of packet error rate during the data transmission phase. The aforementioned research issues are highly motivated us to design an accurate packet error probability model for validating the performance of IEEE 802.15.6 MAC protocol. So the proposed analytical model assumes that assigning the fixed or infinite R_l value for all UP nodes and expanding the CW size after every unsuccessful channel attempts will greatly affect the network throughput under saturation network traffic regime. Moreover, an increase in both R_l and CW leads to high risk of backoff conflicts, collisions and packet errors.

The main contributions of this paper are summarized as follows: (1) Fixed R_l values are assigned to different UP nodes to minimize the backoff delay and investigate the effects of packet errors in the ongoing transmission. (2) An analytical model is designed to evaluate the packet errors experienced by the sensor nodes owing to R_l instead of collisions and transmission failures. (3) The accuracy of analytical model is verified by the simulations under different metrics in order to evaluate the performance of IEEE 802.15.6 MAC protocol.

3 Related Works

Most of these techniques have focused only on how to design efficient schemes for emergency or critical data transmission. There are many factors that degrade the performance of the IEEE 802.15.6 MAC protocol. Generally, the packet errors usually occur due to non-ideal channel condition. Authors in [6] introduced the packet error probability model to study how hidden terminal affects the performance of IEEE 802.11 MAC protocol by incorporating collision probability. This model assumed that the packet errors occur due to collisions or transmission failures. The same model has been further extended to estimate the variability and capacity of the 802.11 MAC protocol [7]. An analytical model presented in [8] to assess the performance of the IEEE 802.15.6 CSMA/CA under the query-based traffic. This model has been validated without considering the hidden terminal problem among the nodes and assumed that collision is the only reason to packets loss. The normalized throughput and mean frame service time of IEEE 802.15.6 CSMA/CA has been validated by considering a portion of the access phases of the superframe and eight user priorities [9]. In [10], a simple and accurate analytical model has been proposed under the assumption of independent busy channel

and packet error probabilities. The performance of IEEE 802.15.6 MAC protocol with finite retry limit has been considered [11] to accurately predict the different states of a node in each backoff phase and reduce the energy consumption. Many analytical models have been designed for verifying the performance under various traffic regime and channel conditions. However, the maximum number of backoff stages [12], CW size of a node after even number of channel access failures [13], and fixed retry limit value [14] in the existing models can increase the possibility of packet drop ratio. To overcome such issues, few authors have been used fixed retry limit [14,15] to reduce the packet error probability and collisions during the data transmission. On other hand, many authors in [16–18] used an infinite retry limit to increase the retransmission probability of each sensor node until succeed. Number of retransmissions can provide a chance to minimize the transmission failures of a node but it will increase the backoff conflicts in the ongoing transmission [19]. Alternatively, the short and long retry limit has been introduced [20] to decrease the overlapping backoff durations and collisions between different UPs.

Moreover, several analytical models [21–23] have been developed under different transmission channel conditions and network traffic. The analytical models mentioned so far assume that packet error occurs only due to packet collisions and packet errors. However, they failed to consider the primary contributor to collisions and packet errors. Hence, to accurately corroborate the performance of IEEE 802.15.6 MAC protocol, a very careful analysis needs to be considered with respect to retry limit in order to enhance the normalized throughput and transmission reliability.

4 Theoretical Analysis

4.1 Transmission Retries

The retry limit value plays a prominent role in practical CSMA/CA of IEEE 802.15.6 MAC protocol. A retry limit (R_l) is the number of times a sensor node attempts to retransmit a data frame before discarding it. Unlike other works, an analytical model adopts the retransmission procedures according to the different UPs and it classifies R_l into two types: Short Retry Limit (SRL) and Long Retry Limit (LRL). Since every sensor node in a network has two retry limits associated with it. The long retry limit, which applies to low priority data frames lesser than the UP6, is set to 4 by default. For instance, a data frame is retransmitted four times before it is discarded or when it is reached the maximum. The short retry limit, which applies to higher priority data frames, equal or higher than the UP6, is set to 3 and 2 by default. Decreasing the RL value reduces the packet errors due to unnecessary frame retransmissions of UP6 and UP5 nodes. Increasing the RL values may decrease the network throughput due to backoff value conflicts (B_{cf}) with UP7 nodes.

This p_r -model introduces the term called intra and inter that represent the conflicts within the same or different group of UP nodes. Longer retry limits may increase the amount of backoff delay and collisions rate. So the R_l values are

incremented every time when a sensor node fails to access the channel and reset to zero after a successful data transmission. In addition, the size of UP nodes will not be increased after even number of channel attempt failures in order to complete a frame transmission. Figure 1 shows the sample frame transmission procedures of p_r -model. As shown in Fig. 1, each sensor node with different UPs selects a Random Backoff Counter (RBC) value from Table 1 and starts its backoff procedures after every pSIFS period using CSMA/CA mechanism. In the first phase, there is no increment in and values, since each and every node can successfully transmitted the data packets when it backoff value reaches zero. In other case, UP6 node only increments its R_i value but not doubled its size after the even number of channel attempt failures (C_f). Therefore, backoff delay and packet errors due to collisions or transmission failures are significantly minimized for UP7 and UP6 node.

No increment in retry limit and CW									
UP	RBC	pSIFS	Is1	Is2	Is3	Is4	Is5	Is6	Is7
UP7	1	pSIFS	1	0(D)					
UP6	3	pSIFS	3	2	1	0(D)			
UP6	6	pSIFS	6	5	4	3	2	1	0(D)
Increment only in retry limit									
UP	RBC	pSIFS	Is1	Is2	Is3	Is4	Is5	Is6	Is7
UP7	1	pSIFS	1	0(D)					
UP6	3	pSIFS	Cf1	Cf2	3	2	1	0(D)	
UP6	6	pSIFS	6	5	4	3	2	1	0(D)

Fig. 1. Sample frame transmission procedures of p_r -model

5 Packet Error Probability Model

5.1 T_c Model

Tay and Chua developed an analytical model called TC [6] to study the effect of packet errors in 802.11 MAC protocol under important metrics that include transmission failure probability (p_f), collision probability (p_c), and saturation throughput. This model assumes that transmission failure occurs only due to collisions. Table 2 lists the notations and descriptions used in the p_r -model. The backoff duration (B_{off}) of a particular node is computed as the sum of average CW size $\frac{2^i * CW}{2}$, where, $i = (0, 1, 2...m)$.

$$B_{off} = \frac{1 - P_c - 1 - P_c(2P_c)^m}{1 - 2P_c} * \frac{CW}{2} \tag{1}$$

The probability of one sensor node colliding with another node in the ongoing transmission is given by $\frac{1}{B_{off}}$. Then, the probability of colliding with $(n - 1)$ nodes is measured as

$$P_c = 1 - (1 - \frac{1}{B_{off}})^{n-1} \tag{2}$$

By substituting the value of B_{off} from Eq. (1) into (2), we can obtain

$$P_c = 1 - (1 - \frac{2(1 - 2P_c)}{1 - P_c - P_c(2P_c)^m} * \frac{1}{CW})^{n-1} \tag{3}$$

The total transmission duration of a data packet T_{tx} is represented as

$$T_{tx} = T_{pCSMslot} + T_{SIFS} + T_{ACK} + T_{payload} \tag{4}$$

Then, the saturation throughput is derived from Eq. (3) and (4)

$$S = \frac{2(1 - P_c)}{2 - P_c} * \frac{T_{payload}}{T_{tx} + \frac{T_{pCSMAslot}}{n+1}} \tag{5}$$

5.2 Proposed p_r -model

The p_r -model (retry limit probability) is an extension of both T_c (collision probability) and p_e -model (packet error probability) by incorporating the retry limit. The model assumes that transmission failures do not only occur due to collisions or transmission failures but it also occurs when a sensor node experiences retransmission or loss of the data packet after reaches its maximum R_l . Another assumption is that n sensor nodes all have different retry limit probability (P_r), as described in Sect. 4. Then, the transmission failure probability (P_f) can be expressed as

$$P_f = P_c + P_e + P_r - P_cP_eP_r \approx P_c + P_e + P_r \tag{6}$$

For simplification, the term $P_c + P_e + P_r$ used in the above Eq. (6) is ignored but it does not affect much in the prediction accuracy. The collision probability p_c is calculated using Eq. (2) as in the T_c model. But in p_r Model, p_f is different from p_e model so the transmission failure probability p_f is used instead of p_c to determine B_{off} in Eq. (1).

$$P_c = 1 - (1 - \frac{2(1 - 2P_f)}{1 - P_f - P_f(2P_f)^m} * \frac{1}{CW})^{n-1} \tag{7}$$

By using Eq. (6), the transmission failure probability p_f is given by

$$P_c = 1 - (1 - \frac{2(1 - 2P_f)}{1 - P_f - P_f(2P_f)^m} * \frac{1}{CW})^{n-1} + p_e + p_r \tag{8}$$

Table 2. Symbols and Descriptions used in p_r -Model

Symbols	Description
R_l	Maximum Retry Limit
C_f	Channel Attempt Failures
B_{off}	Average Contention Window (CW) size
$T_{pCSMAslot}$	Slot time
T_{pSIFS}	Time Duration of Short Inter-Frame Space
$T_{payload}$	Time to Transmit Payload Bits
T_{ACK}	Transmission Time for an Acknowledgment
β_{cycle}	Time between the Start of Two Payload Transmissions
φ_{succ}	Rate of Successful Transmissions
φ_{coll}	Rate of Collisions
φ_{pkterr}	Rate of Packet Errors
φ_{txn}	Rate of Collisions
p_c	Probability of Collisions
p_f	Probability of Transmission Failure
p_e	Probability of Packet Error
p_r	Probability of R_l Exceed
S	Channel Utilization by Successful Transmission of Payload Bits

Let φ_{txn} and φ_{succ} be the rate of transmissions or transmission failures and the rate of successful transmissions, respectively. Then, the probability of successful transmission ($1 - p_f$) is computed as

$$\frac{\varphi_{txn}}{\varphi_{succ}} = (1 - p_f) \quad (9)$$

Let φ_{coll} and φ_{pkterr} be the rate of collisions and the rate of packet errors respectively. If the backoff counter values of two UP nodes are same in the ongoing transmission, then the collision will occur between these two UP nodes. In a similar way, if more than one UP nodes use R_l for their retransmissions, then the probability of collisions will be higher when nodes reached their maximum R_l or retransmit the collided data packets due to B_{cf} . Thus, the rate of transmission failures due to packet errors is calculated by

$$\varphi_{txn} - \varphi_{succ} = 2\varphi_{coll} + \varphi_{pkterr} + R_l \quad (10)$$

$$\frac{2\varphi_{coll}}{\varphi_{pkterr} + R_l} = \frac{p_c}{p_r} \quad (11)$$

Let β_{cycle} be the average time between the starts of two data transmissions under saturation network traffic condition. Assume that the collisions with each other

nodes occur at the same time, thus the φ_{coll} will lead to β_{cycle} as follows

$$\frac{1}{\beta_{cycle}} = \varphi_{succ} + \varphi_{coll} + \varphi_{pkterr} + R_l \tag{12}$$

By solving the above Eqs. (9), (10), (11) and (12), the following expressions such as φ_{txn} , φ_{succ} , φ_{coll} and φ_{pkterr} can be obtained in terms of p_f , p_r and β_{cycle} ,

$$\varphi_{coll} = \frac{p_f - p_r}{2 - p_f + p_r} * \frac{1}{\beta_{cycle}} \tag{13}$$

$$\varphi_{succ} = \frac{2(1 - p_f)}{2 - p_f + p_r} * \frac{1}{\beta_{cycle}} \tag{14}$$

$$\varphi_{pkterr} = \frac{2p_r}{2 - p_f + p_r} * \frac{1}{\beta_{cycle}} \tag{15}$$

$$\varphi_{txn} = \frac{2}{2 - p_f + p_r} * \frac{1}{\beta_{cycle}} \tag{16}$$

The saturation throughput of IEEE 802.15.6 MAC protocol is measured by using (17)

$$S = \varphi_{succ} + T_{payload} = \frac{2(1 - p_f)}{2 - p_f + p_r} * \frac{T_{payload}}{\beta_{cycle}} \tag{17}$$

6 Simulation Results and Model Validations

The simulation result of model is validated by using OMNeT++ Castalia framework. To investigate the effect of packet errors due to R_l and the saturation

Table 3. Packet format and Timing parameters

Packet Format		
Parameters	Values	Unit(s)
pMaxFrameBodyLength	0-255	Bytes
MAC header size	7	Bytes
Beacon size	15	Bytes
mCSMA backoffs	5	In numbers
mCSMATxLimit	2,3 and 4	R_l for UPs
Timing Parameters		
Parameters	Values	Unit(s)
pCSMAslot	125	Microseconds
Superframe period	150	Milliseconds
Beacon period	0.45	Milliseconds
pShortInterframeSpace	75	Microseconds
pAllocationSlotMin	500	Microseconds
pAllocationSlotResolution	500	Microseconds
pCSMAMACPHYTime	40	Microseconds

throughput due to the p_f . The results are obtained by varying the value of R_l according to the different UP nodes. The retry limit values for UP7, UP6 and UP7 are fixed to 2, 3, and 4, respectively. Table 3 depicts the packet format and timing parameters used in the simulation.

6.1 Numerical Validation of p_r -Model

Figure 2 shows the validation of p_r -model with respect to different R_l and varying payload size. From the simulation analysis, when the payload size increases from 0–250 bytes, the probability of transmission failures also increases for all UP nodes. But it is also noticed that the packet errors due to transmission failures of UP5 and UP6 are much higher than UP7. Further, UP7 typically increases error rate when the value of R_l is predicted to rise above its inflation limit. However, it produces low error rate even for the maximum payload size. On the contrary, the transmission failure probability of UP6 and UP5 are gradually increased when the payload size increases from 50–250 bytes. Therefore, the packet errors decreases, there is a decrease in both minimum CW size and retry limit values. From the overall analysis, an increase of both retry limit and CW size due to channel attempt failures led to an increase in transmission failures. However, the transmission failure probability of p_r model achieved better results as compared with existing analytical models by minimizing the CW boundary values and introducing SRL for UP7 and LRL for UP6 and UP5, respectively. The impact of packet error rate with respect to the retry limit for different UP nodes is illustrated in Fig. 3. From the results, it is noted that the packet error

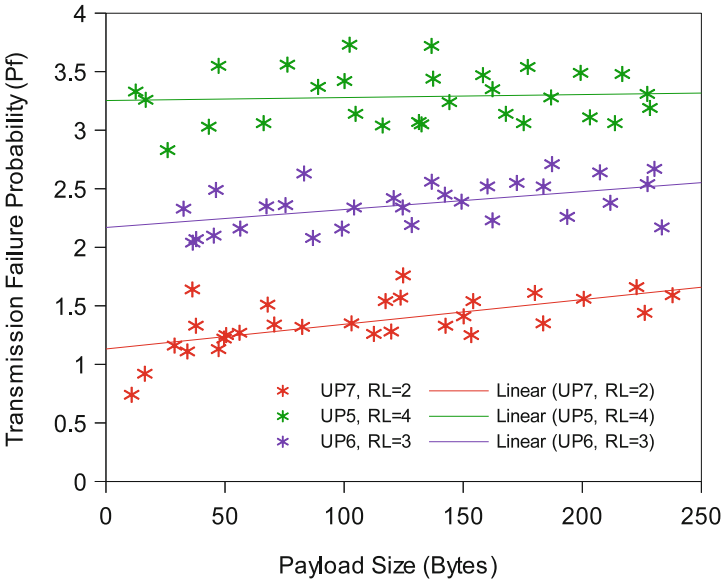


Fig. 2. Validation results of with and payload size

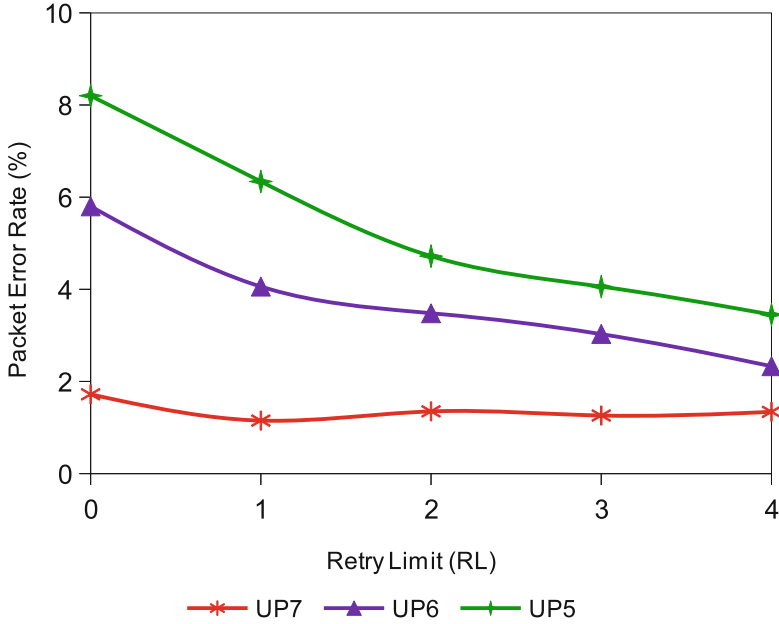


Fig. 3. The effects of and with different UP nodes

rate of UP7 is very less up to 2 (i.e. $R_l = 2$) then it starts slowly increase from $R_l = 2$ to 4. As the value of increases from 0 to 4, there is an increase in packet error rate. Since it is evident that the packet error of UP7 is reduced from 1.72% to 1.34% for the maximum as compared with other two UP nodes. Moreover, the variability of packet error rate for UP6 is systematically decreased by minimizing the CW and retry limit. Though, it fails to yield better results at the beginning it reduces the error rate after this point or the value of starts decreases from 2 to 4. The infinite retry limit and higher retransmission probability of collided data packets during the backoff phase increases the packet error rate of UP6 nodes. Likewise, if there is an increase in, the packet error rate of UP5 is decreased from 8.2% to 3.45%. Further, the proposed analytical model is greatly minimized the packet error rate by 2.11% over UP6 and 0.99% over UP5, respectively. From the overall validation results, it is evident that it is possible to minimize the packet error even if the value of p_r , p_f and p_c increases during the data transmission. However, this value should be fixed according the priority of data traffic or UP nodes. The higher probability of packet error occurs when each node has the minimum value of R_l , which further verifies that the variability of p_r -model. The variations of saturation throughput for three different UP nodes with payload is shown in Fig. 4. It is noted that the throughput value of UP5 increases slightly with the increase of the payload size. In this graph, the achieved throughput result of UP6 is 31 kbps, which is much higher than UP5 and lower than UP7 node when the payload increases from 0 to 250 bytes. For the same payload size, UP7 significantly increases the value of S (41 kbps) than UP5 nodes. Moreover,

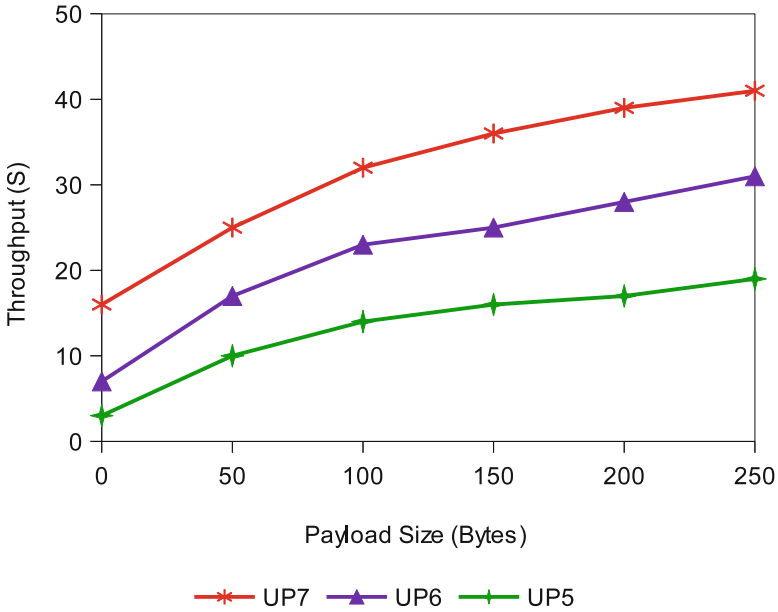


Fig. 4. The effects of saturation throughput with different UP nodes

the quantitative analysis shows that the average throughput value of UP7 is improved by 21% over UP5 and 9% over UP6. By minimizing the CW size and incorporating R_l , the throughput rate of UP7 and UP6 is significantly improved for the maximum payload size. However, in case of UP5 it fails to produce better results due to LRL and p_r .

7 Conclusion

This paper presents an analytical model for validating the effects of packet errors on performance of the IEEE 802.15.6 MAC protocol using CSMA/CA technique. To effectively predict the packet error rate, p_r -model introduces the SRL and LRL values and reduces the ranges of CW size based on the various UP nodes. Moreover, the modified values for all three UP nodes are fixed to 2, 3 and 4, respectively. Thus, it can significantly increase the transmission probability of UP nodes with minimum errors and collisions. The simulation results implies that the p_r -model has the ability to reduce the packet error rate of all UPs with respect to R_l and CW size. The quantitative analysis reveals that the UP7 node has reduced the packet error rate by 2.11% and 0.99% against UP6 and UP5 nodes, respectively. The further analysis of this study will focus on the effects of packet errors with respect to number of sensor nodes and retry limit conflicts under saturated network traffic condition.

References

1. Wan, J., et al.: Wearable IoT enabled real-time health monitoring system. *J. Wirel. Commun. Netw.* **298**, 1–10 (2018)
2. Ullah, S., et al.: A comprehensive survey of wireless body area networks on PHY, MAC, and network layers solutions. *J. Med. Syst.* **36**, 1065–1094 (2012)
3. Huang, R., Nie, Z., Duan, C., Liu, Y., Jia, L., Wang, L.: Analysis and comparison of the IEEE 802.15.4 and 802.15.6 wireless standards based on MAC layer. In: Yin, X., Ho, K., Zeng, D., Aickelin, U., Zhou, R., Wang, H. (eds.) *HIS 2015. LNCS*, vol. 9085, pp. 7–16. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19156-0_2
4. Sagar, A.K., Singh, S., Kumar, A.: Energy-aware WBAN for health monitoring using critical data routing (CDR). *Wirel. Pers. Commun.* (2020)
5. Yuan, X., Li, C., Yang, L., Yue, W., Zhang, B., Ullah, S.: A token-based dynamic scheduled MAC protocol for health monitoring. *J. Wirel. Commun. Netw.* **125**, 1–18 (2016)
6. Sagar, A.K., Singh, S., Kumar, A.: Packet error model for the IEEE 802.11 MAC protocol. In: *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications* (2003)
7. Sagar, A.K., Singh, S., Kumar, A.: Capacity and variability analysis of the IEEE 802.11 MAC protocol (2003)
8. Cavallari, R., Buratti, C.: On the performance of IEEE 802.15.6 CSMA/CA with priority for query-based traffic. In: *European Conference on Networks and Communications*, pp. 1–5 (2014)
9. Khan, P., Ullah, N., Ullah, S., Kwak, K.S.: Analytical modeling of IEEE 802.15.6 CSMA/CA protocol under different access periods. In: *International Symposium on Communications and Information Technologies (ISCIT)*, pp. 151–155 (2014)
10. Ullah, S., Tovar, E.: Performance analysis of IEEE 802.15.6 contention-based MAC protocol. In: *IEEE International Conference on Communications (ICC)*, pp. 1–7 (2015)
11. Sarkar, S., Misra, S., Bandyopadhyay, B., Chakraborty, C., Obaidat, M.S.: Performance analysis of IEEE 802.15.6 MAC protocol under non-ideal channel conditions and saturated traffic regime. *IEEE Trans. Comput.* **64**, 2912–2925 (2015)
12. Jacob, A.K., Kishore, G.M., Jacob, L.: Lifetime and latency analysis of IEEE 802.15.6 WBAN with interrupted sleep mechanism. *Sadhana* **42**, 865–878 (2017)
13. Kumar, V., Gupta, B.: Performance analysis of IEEE 802.15.6 CSMA/CA protocol for WBAN medical scenario through DTMC model. *J. Med. Syst.* **40**, 1–16 (2016)
14. Sadra, S., Abolhasan, M.: On improving the saturation performance of IEEE 802.15.6-based MAC protocols in wireless body area networks. In: *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1233–1238 (2017)
15. Su, X., Li, C., Yuan, X.: IEEE 802.15.6-based Prototype System for WBAN. Design and Implementation. [arXiv:1701.02421v1](https://arxiv.org/abs/1701.02421v1) [cs.NI] (2017)
16. Deepak, K.S., Babu, A.V.: Improving reliability of emergency data frame transmission in IEEE 802.15.6 wireless body area networks. *IEEE Syst. J.* 1–12 (2017)
17. Frigo, G., Giorgi, G.: IEEE 802.15.6 compliant WBSN: a case study. In: *International Workshop on Measurement and Networking*, pp. 1–6 (2017)
18. Suzuki, T.: Performance analysis of IEEE 802.15.6 CSMA/CA using equilibrium point analysis. In: *23rd Asia-Pacific Conference on Communications (APCC)* (2018)

19. Ambigavathi, M., Sridharan, D.: Traffic priority based channel assignment technique for critical data transmission in wireless body area network. *J. Med. Syst.* **42**, 1–19 (2018)
20. Ambigavathi, M., Sridharan, D.: Saturation throughput analysis of IEEE 802.15.6 under ideal transmission channel condition. *Int. J. Comput. Int. IoT* **2**(4), 790–798 (2018)
21. Saboor, A., Ahmad, R., Ahmed, W., Alam, M.M.: A unique backoff algorithm in IEEE 802.15.6 WBAN. In: *IEEE 88th Vehicular Technology Conference (VTC-Fall)* (2018)
22. Rashwand, S., Misic, J., Misic, V.B.: Analysis of CSMA/CA mechanism of IEEE 802.15.6 under non-saturation regime. *IEEE Trans. Distrib. Syst.* **27**(5), 1279–1288 (2016)
23. Gupta, S.H., Devarajan, N.: Performance exploration of on-body WBAN using CM3A-IEEE 802.15.6 channel model. *J. Ambient. Int. Hum. Comput.* 1–12 (2020)



Semantic Meaning Based Bengali Web Text Categorization Using Deep Convolutional and Recurrent Neural Networks (DCRNNs)

Md. Rajib Hossain and Mohammed Moshiul Hoque^(✉)

Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology,
Chittagong 4349, Bangladesh
moshiul_240@cuet.ac.bd

Abstract. Web text categorization is a procedure of deliberately assigning a web text document into one of the pre-defined classes or categories. It is a very challenging task to manipulate, organize, and categorize an enormous amount of web text data in manually. This paper proposes an automatic text categorization framework to classify Bengali web text data using deep learning. The proposed framework comprises of three key constituents: text to feature extraction, training, and testing. The categorization framework is trained, validated, and tested at 120K, 12K, and 36K datasets, respectively. The proposed system achieved 99.00% accuracy in the training phase, 96.00% in the validation phase, and 95.83% in the testing phase, respectively.

Keywords: Bengali language processing · Web text categorization · Web crawler · Semantic feature extraction · Deep convolution networks · Recurrent neural network

1 Introduction

In recent years, Bengali text resources are proliferating on the web due to the cyclopean utilization of digital contrivances with Internet availability. Variants of the web pages and digital portals are the most common voluminous reserves of texts in subsistence that are liberatingly available. It covers a vast range of contexts/domains and perpetually updated by the incipient kinds of information which are multi-dimensional. Hence, it is a very arduous task to manipulate, retrieve, or categorize such an abundance of web text data efficiently. To have facilely accessible or sorted out to these enormous amounts of text data, the process of these data should accomplish with some widely accepted text processing approaches. Manual labelling of immense quantities of data into their predefined classes requires substantial time and cost of money, which are inconvenient, fallible, or impracticable. Consequently, the perspicacious text categorization technique is one of the most exceptional desirable solutions to process

such an immense amount of text data that truncates human labour, preserves time, and reduce the cost of money.

There are a variety of possible applications of web text categorization system. Academicians, industry, security agencies, data scientists, and some other areas are essential to access the voluminous categorical web text data. With the concern of aims, the problem of web text categorization attracted by many NLP scholars in recent years. There are various challenges to develop a text categorization system from Bengali web texts due to the scarcity of e-resources and the unavailability of the benchmark dataset. The web is a source of the large volume of text data that contains multi-dimensional information, and it is quite impossible to process and retrieve information from these raw text data manually. This work suggests a deep convolution and recurrent neural networks (DCRNNs) based on a predicated framework that automatically categorizes Bengali web texts. The proposed framework extracts the semantic text feature from the web text and engenders a DCRNNs predicated web text classifier model.

2 Previous Work

Although there are a remarkable amount of work has been conducted on classifying web texts in English, Chinese, Arabic, and other European languages, no eminent research activities are conducted till now in the Bengali language perspectives. Web text conventionally found in an unstructured form that should convert into structured textual form before utilizing to text categorization purpose. This conversion is commonly known as feature extraction. Mikolov et al. [11] were developed a prognostication predicated feature extraction algorithm name Word2Vec to extract semantic features from the unstructured texts. A couple of work used Word2Vec models to extract features and statistical machine learning techniques used to classify text documents [9, 14]. The Global Vectors for Word Representation (GloVe) is a frequency-predicated text to feature representation technique that captured the semantic and syntactic linguistic features [16]. In recent years, the morphological feature extraction technique has gained more attention by the researchers for Arabic, and English text processing [21, 22]. However, it is quite arduous to implement this technique for low resource languages (i.e., Bengali) due to the intricacy in hyper-parameters adjustment and desirable rule cull [5, 6].

SVM and statistical machine learning-predicated techniques achieved better results for minuscule dataset than the larger dataset [1, 2, 9]. Stochastic gradient descent (SGD) predicated system can handle a voluminous amount of data but required colossal time and recollection for classifier model generation [12]. A supervised machine learning-based approach proposed for text categorization, which utilized SVM and C4.5 [10]. This approach is unable to deal with the semantic meaning of texts. A clustering-predicated framework designed for Bengali document categorization, but this work is unable to achieve plausible precision due to high dimensional feature space [14]. Salama et al. [13] developed a semi-supervised machine learning technique for text tag categorization

in industry perspectives. Liu et al. [4] engendered a fuzzy approach for text classification, which abstracted ambiguity is utilizing two-stage learning. It gained excellent precision but required huge training time. Bassine et al. [19] proposed a decision tree based categorization system for Arabic text, which achieved 92.9% precision in 6 different categories. A hierarchical text classification system proposed by Stein et al. [3], which utilized the word embedding technique. They used FastText to extract semantic features and tree-predicated classification techniques.

Convolution neural networks and semantic word embedding based document categorization systems for English text documents has introduced in [15, 18]. A character-CNN based model is proposed to classify short text documents which gained good accuracy but required much time in training and testing phases [17]. Another work based on Gated recurrent neural network (GRNN) and RNN techniques produced higher precision for short texts but demanded more preponderant training time for colossal scale data sets [7]. Hossain et al. [8] developed a DCN based Bengali document categorization system that dealt with 86,199 training, 10,707 test document. This system achieved acquired 94.96% accuracy for 12 categories. In this paper, GloVe algorithm used for extracting semantic and syntactic features [16]. A deep convolution and recurrent neural networks (DCRNNs) based approach propose for Bengali web text categorization purposes.

3 Methodology

The primary purpose of our research is to build up a text categorization framework that can classify or categorize the web texts (written in Bengali) using DCRNNs techniques. The framework comprises of three significant components: text to feature extraction, training, and testing. Figure 1 depicts the overview of the suggested framework.

3.1 Text to Feature Extraction Module

In the initial step, GloVe embedded model trained for feature extraction purposes using our developed corpus, which contained texts of 12 categories. The categories are accident (Acd), art (At), crime (Cm), economics (Em), education (Et), entertainment (Etm), environment (Env), health (Ht), opinion (O), politics (Plt), science & technology (Sct) and sports (S) respectively. Each text data is added zeros padding or truncates the first 2000 words ($W_1, W_2, W_3 \dots W_{2000}$). Figure 1 shows a set of 200 extracted features $F_1, F_2, F_3 \dots F_{200}$ for each word and add row-wise one after another. GloVe feature extractor provided a 2D (2000×200) feature matrix. The 2D feature matrix is the final feature illustration that forgiven the unstructured labelled data.

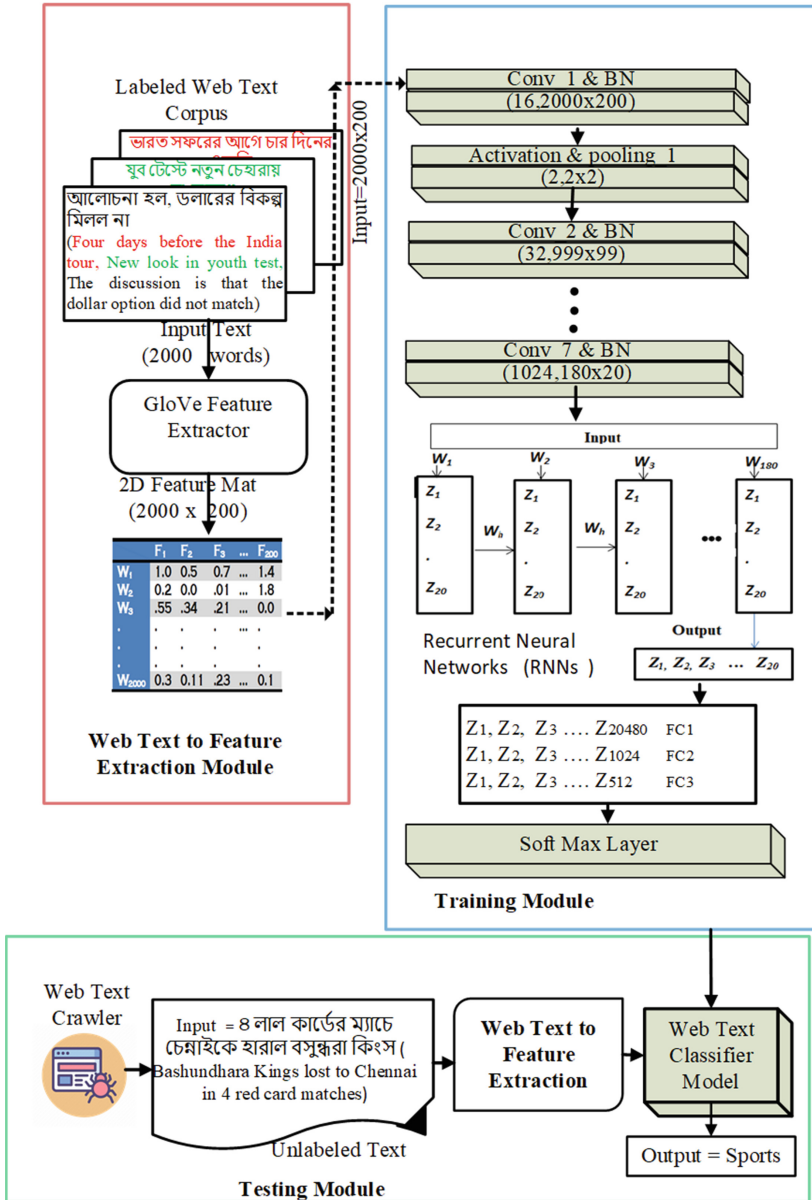


Fig. 1. Proposed framework of Bengali web text categorization.

3.2 Training Module

The propose DCRNNs architecture will train with the developed dataset (D). Suppose, a web text training corpus (WTC), where the $WTC =$

$D_1, D_2, D_3 \dots D_n$, and D_n refers to the n^{th} web texts and n is the number of text documents in training corpus. Here $D = W_1, W_2, W_3 \dots W_N$ and N is the number of words in each documents. Semantic and syntactic features is extracted for each words (W) and $W_N = F_1, F_2, F_3 \dots F_e$, where F_e is the e^{th} feature of N^{th} word in which $e = 200$ mentioned as the embedding feature length. The input layer is feed with a $X_{2000 \times 200}$ feature matrix at first convolution (Conv) layer with 16 kernels ($K_{2 \times 2}$). For each index of feature matrix $X(i, j)$ a convolution operation is applied based on Eq. 1.

$$X(p, q) = \sum_{i=0}^{k_h+p} \sum_{j=0}^{k_w+q} X(p+i, q+j) \times K(i, j) \quad (1)$$

Where, k_h and k_w indicates the kernel height and width. Each of the convolution operations followed by the batch normalization (BN) operation. BN operation is changed the element-wise matrix values. The normalized value improves the training speed and reduces the model over-fitting problem. After BN operation, X is passing through the activation layer. Leaky ReLU activation function is applied according to Eq. 2 [20].

$$X(p, q) = \max(X(p, q), (X(p, q) \times 0.01)) \quad (2)$$

Each of the index values changed after the execution of the ReLU activation function. The pooling operation reduces the feature dimension. DCRNNs framework contains seven convolution layers, three pooling layers, and four activation layers. The outputs of the last convolution layer are 1024 kernels (2×2) and $X_{180 \times 20}$ feature dimension. RNN layers start with 180 hidden units ($W_1, W_2, W_3 \dots W_{180}$) and each of unit feed with 20 features ($Z_1, Z_2, Z_3, \dots Z_{20}$). Equations 3–4 are used to generate RNNs output from the input.

$$H_t = W_z \times W_t + W_h \times H_{t-1} + b_t \quad (3)$$

$$Y_t = W_y \times H_t + b_y \quad (4)$$

Where, H_t , W_z , W_h and b_t are means t^{th} time hidden units output, embedding weights, hidden weights and bias value respectively. Y_t and W_y indicates the unit's output and activation weights. For each word, input and output are the same dimensions, and RNNs layer produced a flatted 20480 feature map. This feature map is feed to the first fully connected layers $FC1$ and followed by $FC2$ and $FC3$. The fully connected layer output is a 512 length feature vector that feeds to the softmax layer or output layer. This layer calculates the expected category name using Eq. 5

$$P(\text{category} = i | X) = \frac{(e^{X^T W_i})}{\sum_{j=1}^C e^{X^T W_j}} \quad (5)$$

Here, X is the i^{th} category feature value, and W_i indicates the softmax layer weights matrix. Total category (C) is equal to 12. The outcome of the softmax layer provides an expected value which used for calculating the deviation.

The deviation value used for backpropagation throws the backward direction. The whole process continues for several epochs and finally produces a web text classifier model.

3.3 Testing Module

Classifier model kernels value is initializing through the architecture. Web crawler crawled web text, and prepared a one-hot vector with size 2000, extracted feature for each word and generate a 2D feature matrix ($X_{2000 \times 200}$). This feature matrix projected with the classifier model and this model produced a score vector ($V_1, V_2, V_3 \dots V_{12}$) where each value represents an expected category value. The Eq. 5 calculates the score value. The maximum value selected among these values for a particular category and the corresponding index will be the category name.

4 Experiments

The proposed system implemented by Python-based Keras deep learning library and ran in GPU 1070Ti based Ubuntu16.04 operating system. The system contains 32 GB physical memory and 8 GB GPU internal memory. The DCRNNs framework is tuned with various hyperparameters: regularization = 0.01, batch size = 64, training epochs = 70, feature extraction = GloVe, coefficient decay = 1.31, dropout = 0.47 and kernel initialized by Xavier function.

4.1 Web Text Corpus

Data acquisition is one of the toughest tasks for digitally resource constraint dialects like Bengali. The web texts are collecting from various online Bengali blogs and newspapers. GloVe embedding algorithm trained with 150K labelled and unlabeled data for embedding purposes. Table 1 depicts the summary of the web text embedding dataset.

Table 1. Statistics of the developed data set.

Attributes	Value
Number of web text documents	150000
Number of sentences	425301
Number of words	73407511
Number of unique words	1200000
Text to feature extraction method	GloVe
Number of context size	12
Feature dimension	200

The web crawler is crawled data from the web and annotated by the crowdsourcing technique. A group of 7 undergraduate students of computer science and engineering background who are doing their thesis on Bengali language processing annotated the data by hand in crowdsourcing phase. Another group consisting of ten participants rechecked all labelled data for validation. Three of them have a postgraduate degree in computer science and engineering, and seven have graduated degrees on Bengali literature. In the corpus, 120K labelled training data used for classifier model generation. Table 2 shows the category-sapient training and testing data summary. The maximum and the minimum number of sentences in the corpus are 90 and 2, respectively. The maximum length of the sentences is 15 words, and a minimum length of 3 words.

Table 2. Summary of training, testing and validation data sets.

Category name	#Training text	#Validation text	#Testing text
Accident (Acd)	10385	1000	3402
Art (At)	6589	700	2146
Crime (Cm)	10638	1100	3812
Economics (Em)	7842	700	2743
Education (Et)	6475	700	2865
Entertainment (Etm)	10874	1100	3934
Environment (Env)	4239	500	1510
Health (Ht)	4320	500	1280
Opinion (O)	11795	1100	3848
Politics (Plt)	26452	2500	4834
Science & Technology (Sct)	6422	600	1994
Sports (S)	13969	1500	3839
Total	120000	12000	36207

4.2 Evaluation Measures

Training/validation loss and accuracy are measured to evaluate the performance of Bengali web text categorization system. The performance of the unlabeled test dataset measured by precision (P), recall (R), F_1 , and accuracy (A). In each training time, losses and accuracy are calculating by Eqs. 6–7.

$$L_i = -W * F_i^t + \sum_{j=1}^C e^{W_j * F_i^t} \quad (6)$$

$$A_i = \frac{P_i}{H_i} \quad (7)$$

Here, L_i and F_i refers to i^{th} time loss and feature vector respectively. The symbol W indicates the soft max layer weight matrix. A_i , P_i and H_i represents by i^{th} time accuracy, number of web text predicted correctly and total number of web text process at that time (t) respectively.

In testing phase precision, recall, accuracy and F_1 -measure are calculated by Eqs. 8–11.

$$\text{Precision} = \frac{T_{pos}}{T_{pos} + F_{pos}} \tag{8}$$

$$\text{Recall} = \frac{T_{pos}}{T_{pos} + F_{neg}} \tag{9}$$

$$\text{Accuracy} = \frac{T_{pos} + T_{neg}}{T_{pos} + F_{pos} + T_{neg} + F_{neg}} \tag{10}$$

$$\text{F}_1 - \text{measure} = \frac{2 * P * R}{R + P} \tag{11}$$

5 Results and Analysis

Our developed dataset evaluates the proposed web text categorization system. Figure 2 and Fig. 3 shows the training/validation loss and accuracy in respect

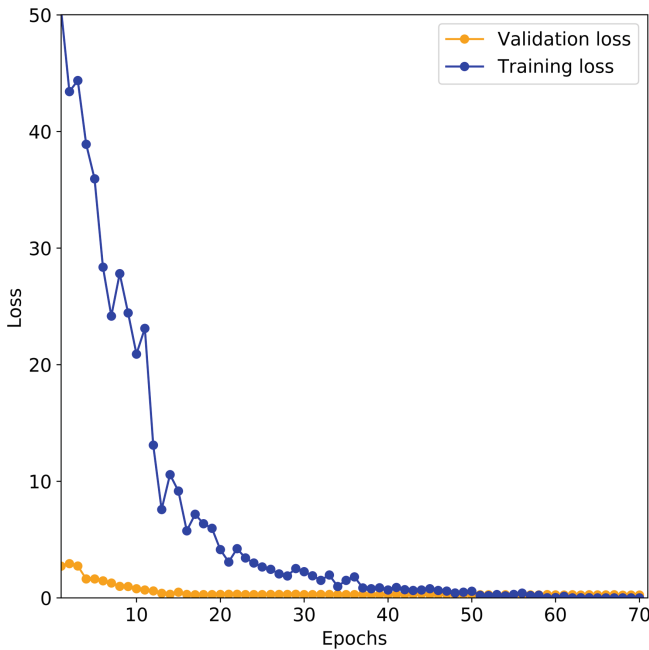


Fig. 2. Training and validation epochs vs loss

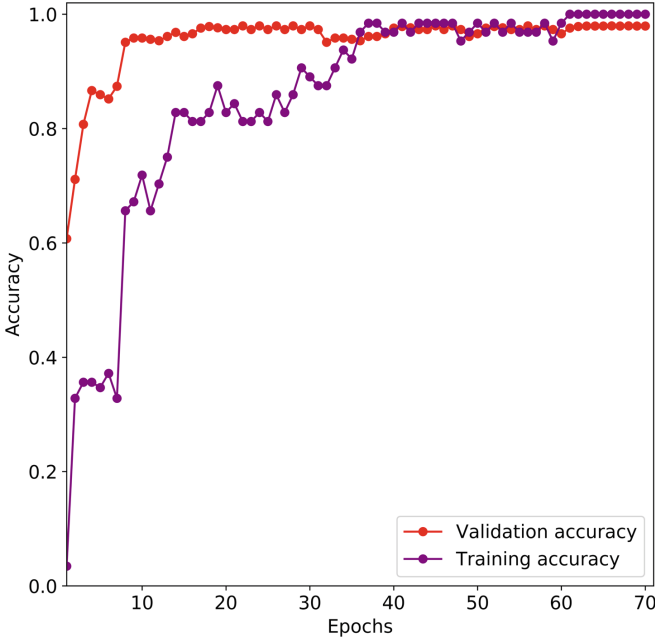


Fig. 3. Training and validation epochs vs accuracy

of a number of epochs. The logarithmic training loss commences with 50.0, and validation loss commences with 2.8 at the first epoch. The loss values gradually minimized from 2 to 30. Validation and training loss is stable after epoch 30 and maintains steady to 70 epochs. Training accuracy is amending up-to 40 epoch and after the session has no astronomically immense vicissitude in accuracy according to epoch number. Ergo, both the precision and loss are converged at epoch number 70.

Table 3 exhibits the precision, recall and F_1 -measure of the web text categorization system. The result indicates that the utmost accuracy (99.00%) is attained for entertainment (Etm) category, while minimum accuracy (94.00%) found for economics (Em). The overall accuracy of the system degraded due to the intra-class overlapping during the data distribution.

5.1 Comparison with Previous Techniques

In order to evaluate the effectiveness of the proposed system, we compared it with the available techniques. Table 4 illustrates the number of training and testing data with categories. The proposed system is working with more train, and testing data compare to previous techniques and also has a more significant number of classes.

Table 5 shows the outcomes of a comparison between the suggested technique and the available techniques concerning accuracy. The comparison result

Table 3. Summary of statistical analysis of web text categorization

Category name	Precision	Recall	F ₁ -score	support
Health (Ht)	0.99	0.97	0.97	1280
Accident (Acid)	0.96	0.92	0.94	3402
Art (At)	0.97	0.93	0.94	2146
Crime (Cm)	0.99	0.96	0.97	3812
Economics (Em)	0.93	0.94	0.94	2743
Education (Et)	0.98	0.93	0.95	2865
Entertainment (Etm)	0.99	0.99	0.99	3934
Environment (Env)	0.95	0.94	0.94	1510
Opinion (O)	0.96	0.96	0.96	3848
Politics (Plt)	0.96	0.98	0.97	4834
Science & Technology (Sct)	0.94	0.98	0.96	1994
Sports (S)	0.96	0.98	0.97	3839
Avg./total	0.956	0.957	0.958	36207

Table 4. Comparison regarding the number of classes and size of the dataset.

Techniques	No. of train data	No. of test data	No. of category
SVM and TF-IDF [10]	1000	118	5
Word2Vec and K-NN [14]	19750	4713	7
DCNNs and Word2Vec [8]	86199	10707	12
SGD and Word2Vec [9]	10000	4651	9
Proposed (DCRNNs)	120000	36207	12

indicates that the proposed web text categorization system outperformed the existing techniques in terms of higher accuracy and a more significant number of document categories.

The accuracy of Mandal et al. [10] and Ahmed et al. [14] techniques are lower due to the limitations of their feature extraction techniques. Hossain et al. [8] developed a Word2Vec and DCNNs based system with 94.96% accuracy, which is lower than the proposed system (95.89%). Dhar et al. [2] developed a

Table 5. Performance comparison.

Techniques	Accuracy (%)
SVM and TF-IDF [10]	89.14
Word2Vec and K-NN [14]	91.02
DCNNs and Word2Vec [8]	94.96
SGD and Word2Vec [9]	93.00
Proposed (DCRNNs)	95.89

web text categorization system with 97.09% accuracy, which is a bit higher than the proposed technique but they considered a fewer number of categories (8). Moreover, this technique is not capable of handling wide-scale datasets.

6 Conclusion

Web text categorization is a popular research agenda for highly resource languages. It is comparatively a challenging task to develop an automatic text categorization system for resource-constrained languages like Bengali. This paper presents a DCRNNs based text categorization system to classify Bengali web text contents. A corpus consisting of 150000 web text documents in 12 categories developed to evaluate the system which is larger than the existing techniques. The proposed system achieved 95.89% accuracy on the developed corpus. The accuracy of the current implementation can be improved by including more labelled data and categories. RNNs may be superseded by long short term recollection (LSTM) for ameliorating the overall accuracy of the system. These are kept for future research issues.

Acknowledgement. This work was supported by the Establishment of CUET IT Business Incubator Project, BHTPA, ICT Division, Bangladesh under the research project, “Automatic Bengali Document Categorization based on Summarization Techniques”.

References

1. Dhar, A., Dash, N.S., Roy, K.: Classification of Bangla text documents based on inverse class frequency. In: 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 05 November, pp. 1–6 (2018)
2. Dhar, A., Dash, N.S., Roy, K.: Categorization of Bangla web text documents based on TF-IDF-ICF text analysis. In: Social Transformation - Digital Way CSI, vol. 836. Springer, Singapore (2018)
3. Stein, R.A., Jaques, P.A., Valiati, J.F.: An analysis of hierarchical text classification using word embeddings. *Journal of CoRR*, vol. abs/1809.01771 (2018)
4. Liu, H., Burnap, P., Alorainy, W., Williams, M.L.: A fuzzy approach to text classification with two-stage training for ambiguous instances. *IEEE Trans. Comput. Soc. Syst.* **6**, 227–240 (2019)
5. Cotterell, R., Schütze, H.: Morphological Word Embeddings. American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.1287–1292 (2019)
6. Salama, R.A., Youssef, A., Fahmy, A.: Morphological word embedding for Arabic. In: The 4th International Conference on Arabic Computational Linguistics (ACLing), Dubai, UAE, vol. 142, pp. 83–93 (2018)
7. Wang, D., Gong, J., Song, Y.: W-RNN: news text classification based on a weighted RNN. *Journal of CoRR*, vol. abs/1909.13077, 28 September 2019

8. Hossain, M.R., Hoque, M.M.: Automatic Bengali document categorization based on deep convolution nets. In: Emerging Research in Computing, Information, Communication and Applications. Advances in Intelligent Systems and Computing, vol. 882. Springer, Singapore (2019)
9. Hossain, M.R., Hoque, M.M.: Automatic Bengali document categorization based on word embedding and statistical learning approaches. In: International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) (2018)
10. Mandal, A.K., Sen, R.: Supervised learning methods for Bangla web document categorization. *Int. J. Artif. Intell. Appl. (IJAIA)* **5**(5), 93–105 (2014)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *Journal of CoRR* (2013)
12. Kabir, F., Siddique, S., Kotwal, M., Huda, M.: Bangla text document categorization using stochastic gradient descent (SGD) classifier. In: Proceedings of the International Conference on Cognitive Computing and Information Processing, pp. 1–4 (2015)
13. Liebeskind, C., Kotlerman, L., Dagan, I.: Text categorization from category name in an industry motivated scenario. *J. Lang. Resour. Eval.* **49**(2), 227–261 (2015)
14. Ahmad, A., Amin, M.R.: Bengali word embeddings and its application in solving document classification problem. In: 19th International Conference on Computer and Information Technology, pp. 425–430 (2016)
15. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), vol. 01, pp. 562–570 (2017)
16. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
17. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolution networks for text classification. In: 28th International Conference on Neural Information Processing Systems, NIPS 2015, vol. 01, pp. 649–657 (2015)
18. Xu, K., Feng, Y., Huang, S., Zhao, D.: Semantic relation classification via convolutional neural networks with simple negative sampling. In: Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 536–540 (2015)
19. Bahassine, S., Madani, A., Kissi, M.: Arabic text classification using new stemmer for feature selection and decision trees. *J. Eng. Sci. Technol.* **12**, 1475–1487 (2017)
20. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the ICML (2013)
21. Al-Taani, A.T., Al-Sayadi, S.H.: Classification of Arabic text using singular value decomposition and fuzzy c-means algorithms. In: Johri, P., Verma, J., Paul, S. (eds.) Applications of Machine Learning. Algorithms for Intelligent Systems, 05 May, pp. 111–123. Springer, Singapore (2020)
22. Bhagat, C., Mane, D.: Text categorization using sentiment analysis. In: Bhalla, S., Kwan, P., Bedekar, M., Phalnikar, R., Sirsikar, S. (eds.) Proceeding of International Conference on Computational Science and Applications. Algorithms for Intelligent Systems, 05 January, pp. 361–368S. Springer, Singapore (2020)



Stability and Power Analysis of Read Decoupled 8T SRAM Cell

Saloni Bansal^(✉) and V. K. Tomar

GLA University, Mathura 281406, UP, India
{Saloni.bansal_mt18, vinay.tomar}@gla.ac.in

Abstract. Stability and power consumption in the circuit are the major column of any SRAM cell idolization. In this paper, a read decoupled 8T SRAM cell has been implemented and compared with conventional 6T SRAM and Differential 8T SRAM cell, analyzed on 45 nm technology with Cadence virtuoso tool. It has been observed that read stability and write ability of considered cell is improved by $1.4\times$ and $1.02\times$ in comparison of conventional 6 T SRAM cell while $1.7\times$ and $1.07\times$ with respect to Differential 8T SRAM cell. The read delay and write delay of read decoupled 8 T SRAM cell is 46.40% and 29% better in comparison of conventional 6T SRAM cell. Additionally, the read power of 8T SRAM cell is reduced by a factor of 55% as comparison of conventional 6T at 1 V supply voltage.

Keywords: Read power · RSNM · WSNM · Read delay · Write delay

1 Introduction

Now a day, demand of portable devices is drastically increasing with the advancement of technology in terms of computing speed of processors, memory storage capacity and network access [1]. Static Random-Access Memory (SRAM) is employ as cache memory, which are inbuilt in microprocessor, System on Chip and Network on Chip products. SRAM cell also occupied maximum chip area when coupling with CMOS circuits [2]. SRAM cell is favored over DRAM (Dynamic Random-Access Memory) due to no need to provide a refresh cycle and having transistors in a cross-coupled flip-flop configuration. The digital sub-threshold logic circuit design is the optimistic method for ultra low power applications. Circuits implemented in the sub-threshold region operate at a supply voltage just closer or to below than threshold voltages (V_{th}) of the transistors [3]. SRAM also shows major concern in internet of things (IOT). Devices based on IOT, runs on very low supply voltages. applications of IOT are biomedical equipments, sensor nodes, wireless sensors, robotics etc. Many researchers [4–6] have been reported power efficient SRAM cells with improved performance as explored in subsequent paragraph. Ashish et al. [7] reported a 11-T SRAM cell with improved stability along with delay at 45 nm technology node. Ultra-low leakage (ULL) 6T SRAM bit-cell had been analyzed with operating frequency 5 to 20 MHz under sub-threshold region with supply voltage 0 to 0.34 V on 55 nm technology which provides effective SNM [8]. The advance 8T SRAM

cell [9] take the advantage of differential oscillation and uses column interleaving model on 180 nm technology to get 59.8% read stability and 3.7 times improved write stability superior to conventional 6T SRAM cell. To form 11T SRAM, an supplementary circuit was added with conventional 6T SRAM [10] on 65 nm technology to lift the SNM and read power by 4 times as comparison with conventional 6T SRAM cell. However, area consume by the circuit was more and different lines were used for read and write. 10T SRAM [11] with sleep transistor through which single bit line and dynamic feedback loop was operate on 45 nm technology on the consequences SNM was improve by $2.91 \times$. The single ended 8T SRAM [12] was implemented with differential word-lines and decoupled bit-lines to improve the RSNM by 67.3%, saves total average power by 41.5%, write power by 39.7% and read power by 89.91 times. The 7T SRAM [13] was implemented on 90 nm technology to improve read static noise margin by 8.7% and power reduction by 46.59% with respect to 6T SRAM cell for 1 V supply voltage. With reference to the above mentioned literature we have simulated and analyzed 6T SRAM, differential 8T SRAM and read decoupled 8T SRAM cell and compared the obtained results.

2 SRAM Cell

A. 6T SRAM Cell Design

Inverter circuit is formed, when two transistors of NMOS (NM1,NM2) & PMOS (PM1,PM2) are connected with each other and form a cross coupled connection thus, works as storage cells,. The transistors N5 and N6 are termed as access transistors used to execute the storage cell. The wordline (wl) is work for both operation logic high or logic low. Read and write operation can be performed by using bitline (bl) and bitline bar (blb) [14]. If wl is raised to logic '1' or logic high then storage cell can be on.so, bl and blb are to be access VDD tends to bl along with blb tends to 0. If write driver goes high then value of q is change from 0 to 1 as consequence write operation is completed. Fig. 1 displays the complete schematic of 6T SRAM cell. Issues related with 6T SRAM cell, i.e. it have greater speed than dynamic RAM although it suffers from poor stability by cause of its noise margin and more power hungry.

B. Read Decoupled 8T SRAM Cell

The Read Decoupled 8T SARM has discrete read and write terminals while read terminal run through RWL and mainly helps in operating the bitcells at the time of read and write operation occurs. Read Decoupled 8T SRAM cell is shown in Fig. 2. The three modes of operation i.e. hold operation, read operation and write operation are discussed below.

Hold Operation. To perform hold operation, initially word-line is set to at 0 supply voltage. Hence, both bl and blb are connected to vdd through which bit-line leakage current can be control.

Read Operation. To performing read operation, initially wordline (wl) should be high by which and both access transistors NM11 and NM12 are on. The bitline (bl) and bitline bar (blb) are connected to precharge circuit. The memory should hold some value. if q

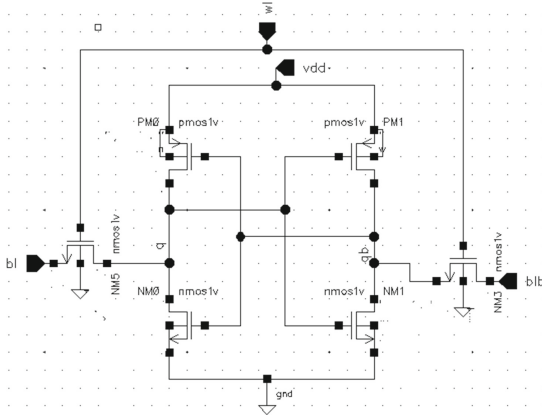


Fig. 1. Conventional 6T SRAM cell

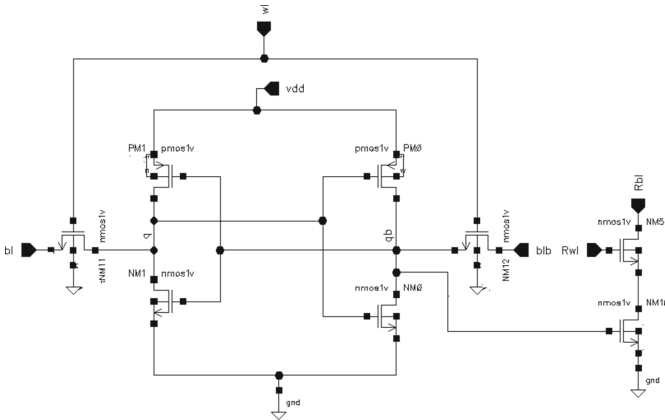


Fig. 2. Read decoupled 8T SRAM

$= 1$ and $qb = 0$, due to $q = 1$ the transistor NM0 should be on, although NM0 is also connected to the ground so transistor NM10 should be at logic 0 due to active NM0 as consequences the output is shown through Rbl. Meanwhile, if $q = 0$ and $qb = 1$ the transistor NM1 should be on due to qb and NM1 is also connected to the ground .the Rbl finds the path to connect with ground from Fig. 3. The read transfer characteristics of proposed Read Decoupled 8T SRAM are shown above. At time the word line is raised to high, the output of bl is high and the output of blb is low execute read operation, (vice versa).

Write Operation. To performing write operation wl is raised to Vdd. Because of high vdd both the access transistor NM11 and NM12 are on. Let us assume, initially both the bitlines bl and blb are tends to 0 after that bl is kept logic high ($bl = 1$) and blb is at logic low ($blb = 0$) vice versa, the access transistor NM11 are on and BLB stores logic 0.Storage node qb discharges through NM12 by BLB, at the same time voltage at node

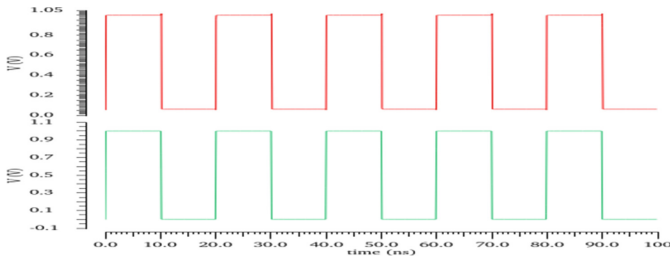


Fig. 3. Read waveform of read decoupled 8T SRAM at 1 V.

q proceeding towards V_{dd} through NM11 by bl. therefore the required data is written accurately. From Fig. 4 shown above The write transfer characteristic of proposed Read Decoupled 8T SRAM when wl is high and bl is high.

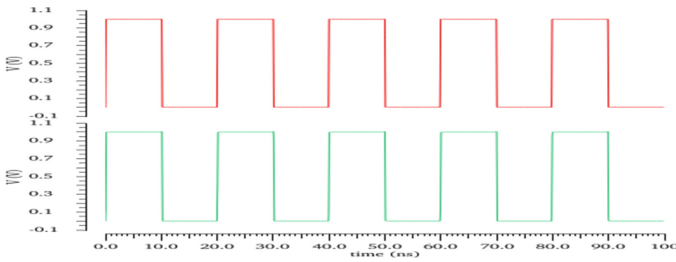


Fig. 4. Write waveform of read decoupled 8 T SRAM at 1 V

3 Results and Simulation

Comparison results are discussed below of Conventional 6T, Differential 8T, Proposed 8T SRAM cell based on Read power, Write power, Read stability, Write stability on 45 nm technology from which proposed 8T SRAM cell gives better results on following aspects:-

A. Read power

On analyzing read power, observations says that Read Decoupled 8T SRAM is 0.44 better than Conventional 6T SRAM cell and Differential 8T SRAM cell, Conventional 6T SRAM cell and Differential SRAM cell is approximately same as shown in Fig. 5 and Table 1 shows the appropriate values of read power.

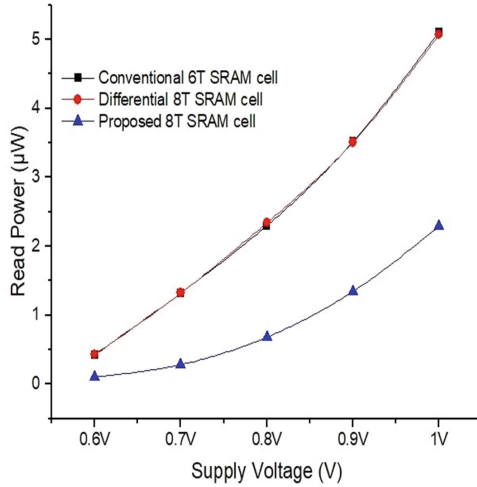


Fig. 5. Read power at different supply voltage

Table 1. Comparison of read power for different supply voltage

Supply voltage	Conventional 6T SRAM cell (µW)	Differential 8T SRAM cell (µW)	Proposed 8T SRAM cell (µW)
0.6 V	0.4291	0.42647	0.1007
0.7 V	1.32	1.327	0.2852
0.8 V	2.3	2.338	0.6812
0.9 V	3.517	3.506	1.344
1 V	5.108	5.065	2.29

B. Write power

Write power of RD 8T SRAM cell should be high as compared to conventional 6T SRAM cell and Differential 8T SRAM cell due to its single ended structure. Table 2 subtitles the appropriate values of read power & Fig. 6 displays the comparison graph of RD 8T SRAM cell with considered SRAM cells.

C. Read stability

Read stability can be measure by RSNM, (Read Static Noise Margin). The gain in dc noise of the circuit, likewise higher the read stability. The read stability of RD 8T SRAM cell is far better than other considered SRAM cells because it has discrete read terminal consists of transistors NM5 and NM10 by which read stability unbound of SNM. However, RSNM can be determine by applying Vdd in between pulldown transistor applying dc analysis in that case keeps w1 should be low. The waveform can be generated by all-encompassing the voltage at the storage node Q (or QB) and bit-lines (bl,blb) [15]. The figuring of noise margin through butterfly curve by shape of square can be fix neatly

Table 2. Comparison of write power for different supply voltage

Supply voltage	Conventional 6T SRAM cell (μW)	Differential 8T SRAM cell(μW)	Proposed 8T SRAM cell(μW)
0.6 V	20.52	24.69	42.37
0.7 V	41.13	39.75	70.96
0.8 V	74.99	60.04	119.0
0.9 V	121.94	87.65	184.0
1 V	178.9	123.3	261.6

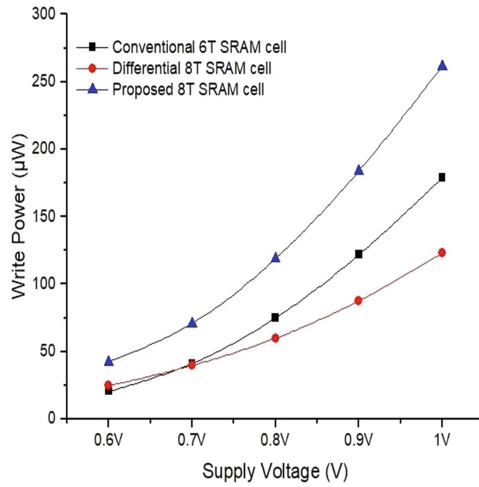


Fig. 6. Write power at different supply voltage

into the briefer lobe of such curve [2]. observation says that RSNM results of proposed Read Decoupled 8T SRAM is 1.7 times better than differential 8T SRAM and 1.4 times better than conventional 6T SRAM cell for supply voltage range from 0.6 V to 1 V due to extra circuitry of NMOS Table 3 and Fig. 7 shows the appropriate value and comparison of RSNM fro proposed Read Decoupled 8T SRAM with Differential 8T SRAM & Conventional 6T SRAM for 0.6 V to 1 V although Fig. 8 displays the n-curve model by which RSNM can be identified manually at 1 V and Fig. 7 shows the appropriate value and comparison of RSNM for pro-posed Read decoupled 8T SRAM with Differential 8T SRAM & Conventional 6T SRAM for 0.6 V to 1 V. Although Fig. 8 displays the n-curve model by which RSNM can be identified manually at 1 V.

Table 3. Comparison of read stability for different supply voltage

Supply voltage	Conventional 6T SRAM cell (mV)	Differential 8T SRAM cell (mV)	Proposed 8T SRAM cell (mV)
0.6 V	130	117	200
0.7 V	150	137	210
0.8 V	165	150	235
0.9 V	203	172	280
1 V	250	210	350

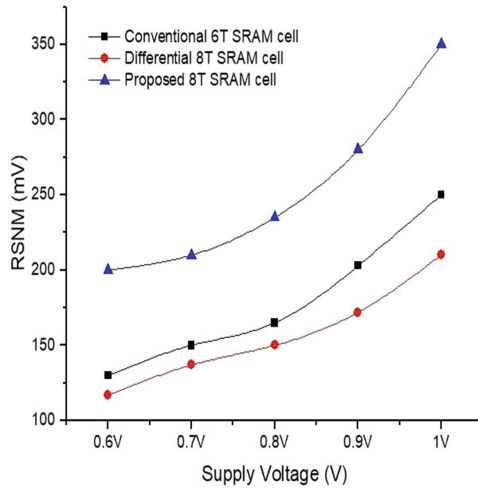


Fig. 7. RSNM at different supply voltage

Analysis of RSNM of Read Decoupled 8T SRAM Cell Through Butterfly Curve

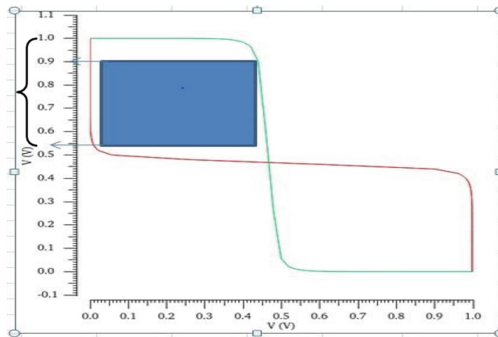


Fig. 8. RSNM through butterfly curve at 1 V.

D. Write ability

Write ability of SRAM can be estimated by WSNM. The write ability can be generated by the unification of both read characteristic curve and write characteristic curve. The read stability is estimated by all-compassing the storage node q and qb. for establishing the write stability, the bl and wl is biased with vdd while blb is with Vss. [15] write VTC can be leveling by fetching the voltage at the storage node q with bl. after this WSNM can be obtained by fixing square in the smallest lobe of the read VTC and write VTC union. Observation says that proposed Read Decoupled 8T SRAM cell is 1.07 times better than conventional 6T SRAM and 1.02 times better than differential 8T SRAM cell. Meanwhile, comparison graph in Fig. 9 and Table 4 discloses the comparison results of RD 8T SRAM cell with considered SRAM cells. the WSNM can be calculated for supply voltage 0.6 V to 1 V.

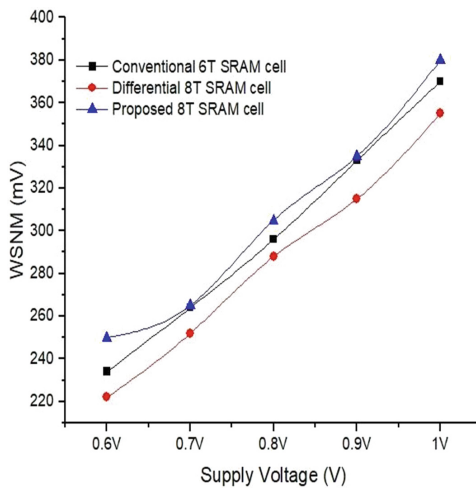


Fig. 9. WSNM at different voltage.

Table 4. Comparison of write stability for different supply voltage

Supply voltage	Conventional 6T SRAM cell (mV)	Differential 8T SRAM cell (mV)	Proposed 8T SRAM cell (mV)
0.6 V	234	222	250
0.7 V	264	252	265
0.8 V	296	288	305
0.9 V	333	315	335
1 V	370	355	380

E. Read Delay

In proposed 8T SRAM cell read delay is higher on comparison of other considered cell It

happen because of higher resistance in read path provides lower read current due to this read bitline capacitance take more time to charging and discharging. Table 5 & Fig. 10 shows the comparison results of read delay of RD 8T SRAM cell with considered SRAM cells.

Table 5. Comparison of read delay for different supply voltage

Supply voltage	Conventional 6T SRAM cell (ps)	Differential 8T SRAM cell (ps)	Proposed 8T SRAM cell (ps)
0.6 V	207.5	92.08	184.7
0.7 V	115.32	57.25	90.43
0.8 V	72.0	47.78	60.48
0.9 V	67.81	41.91	45.51
1 V	67.23	36.9	36.03

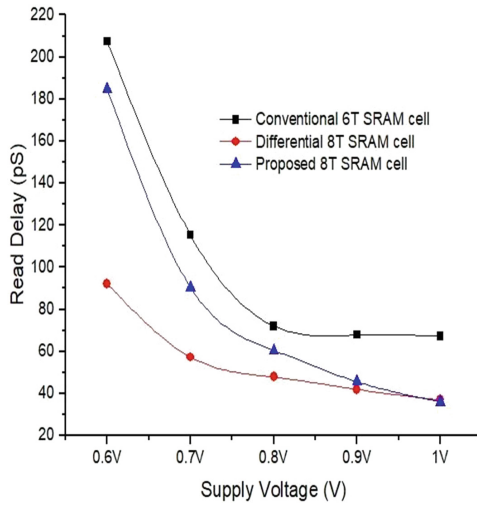


Fig. 10. Read Delay at different supply Voltage

F. Write Delay

Write delay of Proposed 8T SRAM cell is 1.40 times better than Conventional 6T SRAM cell and 1.16 times better than Differential 8T SRAM cell shown by Table 6 & Fig. 11.

4 Conclusion

In this work, the performance parameters such as read power, read stability, write ability, read delay and write delay of a read decoupled 8T SRAM cell has been compared

Table 6. Comparison of write delay for different supply voltage

Supply voltage	Conventional 6T SRAM cell (ps)	Differential 8T SRAM cell (ps)	Proposed 8T SRAM cell (ps)
0.6 V	795.6	194.5	258.2
0.7 V	271.1	149.9	116.8
0.8 V	138.4	96.22	77.65
0.9 V	95.2	74.0	62.1
1 V	75.79	62.62	53.77

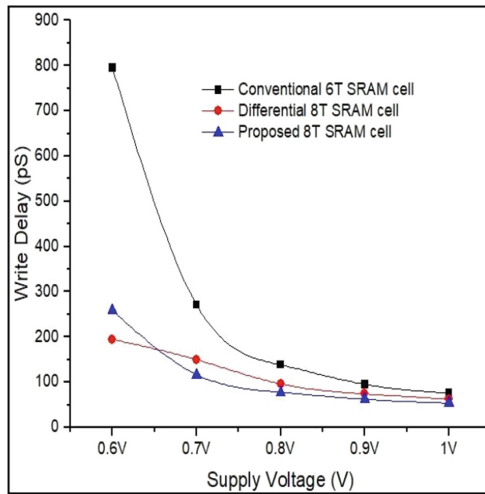


Fig. 11. Write delay at different supply voltage.

and analyzed with conventional 6T SRAM and differential 8T SRAM cells. This paper proposes a RD 8T SRAM cell is robust in terms of stability i.e. read stability and write ability. Focused parameters are read power, write power, read stability, write stability, read delay, write delay are explored. Upgradation in most of the parameters are recognized on measuring with other SRAM cells. The proposed cell is a workable option in terms of stability.

Reference

1. Reddy, T.V., Madavi, B.K.: Designing of schmitt trigger-based architecture 8T SRAM of 256 bit cells under 45 NM technology for low power applications. In: 2017 International Conference on Intelligent Computing and Control (I2C2). IEEE (2017)
2. Pal, S., Islam, A.: Variation tolerant differential 8T SRAM cell for ultralow power applications. IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. **35**(4), 549–558 (2015)

3. Huang, C.-R., et al.: Ultra-low standby power SRAM with adaptive data-retention-voltage-regulating scheme. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE (2018)
4. Kumar, H., Tomar, V.K.: Stability analysis of sub-threshold 6T SRAM cell at 45 nm for IoT application. *Int. J. Recent Technol. Eng. (IJRTE)* **8**(2), 2432–2438 (2019)
5. Tomar, V.K., Sachdeva, A.: Implementation and analysis of power reduction techniques in charge transfer sense amplifier for sub 90 nm SRAM. In: 8th International Conference on Computing, Communication and Networking Technologies ICCCNT- 2017 held on 3rd -5th July- 2017, IEEE-40222, IIT Delhi, New Delhi (2017)
6. Naik, S., Sonia, K.: A novel 8T SRAM with minimized power and delay. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE (2017)
7. Sachdeva, A., Tomar, V.K.: Design of a stable low power 11-t static random access memory cell. *J. Circ. Syst. Comput.* **29**(13) (2020)
8. Misawa, N., et al.: SNM analytical approach to robust subthreshold SRAM operation based on the 55 nm DDC technology. In: 2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S). IEEE (2017)
9. Kim, T., Kim, H., Chung, Y.: Design of advanced subthreshold SRAM array for ultra-low power technology. In: 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE). IEEE (2018)
10. Moradi, F., et al.: 65 nm sub-threshold 11t-sram for ultra low voltage application. In: 2008 IEEE International SOC Conference. IEEE (2008)
11. Grace, P.S., Sivamangai, N.M.: Design of 10T SRAM cell for high SNM and low power. In: 2016 3rd International Conference on Devices, Circuits and Systems (ICDCS). IEEE (2016)
12. Raikwal, P., Neema, V., Verma, A.: High speed 8T SRAM cell design with improved read stability at 180nm technology. In: 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), vol. 2. IEEE (2017)
13. Surekha, G., Balaji, N., Sai, Y.P.: Stability analysis of 7T-SRAM cell. In: 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, pp. 1–4 (2016)
14. Madhavi, B.K., Vasudeva Reddy, T.: Design strategy & analysis of Subthreshold SRAM in power & delay for wearable applications. In: 2017 2nd International Conference on Communication and Electronics Systems (ICES). IEEE (2017)
15. Singh, J., Saraju, P.M., Dhiraj, K.P.: Robust SRAM Designs and Analysis. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-1-4614-0818-5>



Analysis of Higher Stable 9T SRAM Cell for Ultra Low Power Devices

Harekrishna Kumar^(✉) and V. K. Tomar

GLA University, Mathura 281406, U.P, India
harekrishnabgp@gmail.com

Abstract. VLSI designers are inspired by the widespread use of portable low power devices. In this paper, a 9T SRAM cell has been analyzed and implemented at 45 nm technology node with Cadence virtuoso tool. The read stability and write ability of considered cell is improved by $2.05\times$ and $1.13\times$ in comparison to conventional 6T SRAM cell. The write access time of 9T SRAM cell is $3.37\times$ and $2.94\times$ better in comparison of conventional 6T and differential (DF) 8T SRAM cell respectively. Furthermore, the write power of 9T SRAM cell is reduced by a factor of $2.07\times$ and $1.77\times$ as comparison of conventional 6T and Differential 8T SRAM cell respectively at 0.5 V supply voltage. The data retention voltage of 9T SRAM cell is better at all corners in comparison of conventional 6T and differential 8T SRAM cell respectively. The 9T SRAM cell may be utilized in IoT based devices such as medical equipments, space applications, etc.

Keywords: Subthreshold · Low power · Stability

1 Introduction

In the modern era, the extensive use of portable devices coupled with high-speed communication has resulted in an exponential demand of low power Static Random Access Memory (SRAM) cell. SRAM covers a noteworthy segment of a system on chip (SoC). The conventional 6T (Conv.6T) SRAM cell offers huge stockpiling capacity with simple cell structure. It works properly in super threshold region. However, Conv.6T SRAM does not work properly in sub-threshold region due to change in transistor characteristics that degrade the stability of the circuit with higher leakage current [1]. As leakage current rises exponentially in sub-threshold region the static power consumption will increase. Therefore, it is necessary to minimize the power associated with SRAM in order to have a power efficient design. The legitimate execution of Conv.6T SRAM cell with maintains stability during read activity, the size of access transistors should be minimum that degrades the bit-line capacitances that improves the read stability. On the other hand, during write operation access transistor should be higher current capability for better write ability in the cell. It shows conflicting relation between read and write ability of Conv.6T SRAM cell [2]. The Conv.6T

SRAM cell is shown in Fig. 1. In this figure transistors P1, N3 and P2, N4 form a core latch circuitry to store the information. The read and write operation is performed by transistors N1 and N2. Bit-lines (BL and BLB) acts like input/output line during write/read activities of the cell.

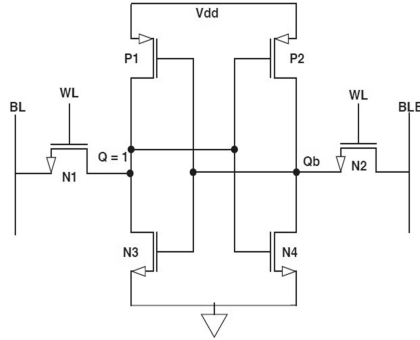


Fig. 1. Conv.6T SRAM [3]

A lot of SRAM topologies have been reported so far that mitigate the issues related to standard 6T SRAM cell. To resolve the inconsistent relationship between read and write stability in standard 6T SRAM cell, separate read and write activity is performed [4, 5]. This approach offers a higher read static noise margin (RSNM) and it is approximately equal to hold static noise margin (HSNM) of the cell. The write ability is also improved due to mitigate the access transistors conflict. The read stability of SRAM cell is commonly improved by the isolation of storage nodes from the read path which is known as the read decoupling approach. Inverter loop cutting approach also improves the read stability of the SRAM cell during read operation [6]. In order to improve the overall stability i.e. read as well as write stability a Schmitt trigger based topology that works on feedback mechanism is reported [7]. Furthermore, in the Conv.6T SRAM cell power dissipation is more due to its differential bit line structure. To reduce the power consumption a single-ended 5T bit cell is reported that diminishing the area and reduces active and standby power consumption as compare to Conv.6T SRAM cell [8]. A higher threshold voltage of transistors reduces the power dissipation at a penalty of degrades the performance of device. However lower threshold voltage transistors improve the performance of the devices with higher power dissipation. A multi-level threshold voltage based SRAM cell is reported. In this paper [9] high threshold voltage transistors are utilized in core latch circuitry to reduce the leakage power in the cell. Whereas low threshold voltage transistors are used in read path that improve the read performance of the devices.

2 9T SRAM Cell

The implemented 9T SRAM cell is depicted in Fig. 2. Transistors P1, N1 and P2, N2 form two back to back connected inverter to store one bit information in the cell. Transistor N3, N4 and N7 are write and read access transistors respectively. The read decoupled structure is formed by transistors N5 and N6.

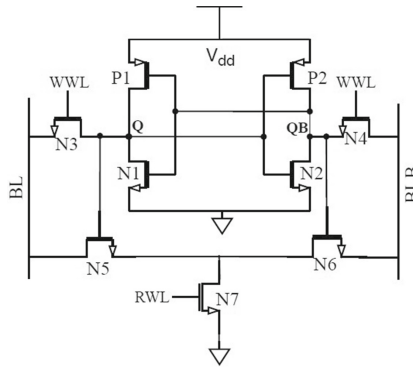


Fig. 2. Proposed 9T SRAM Cell

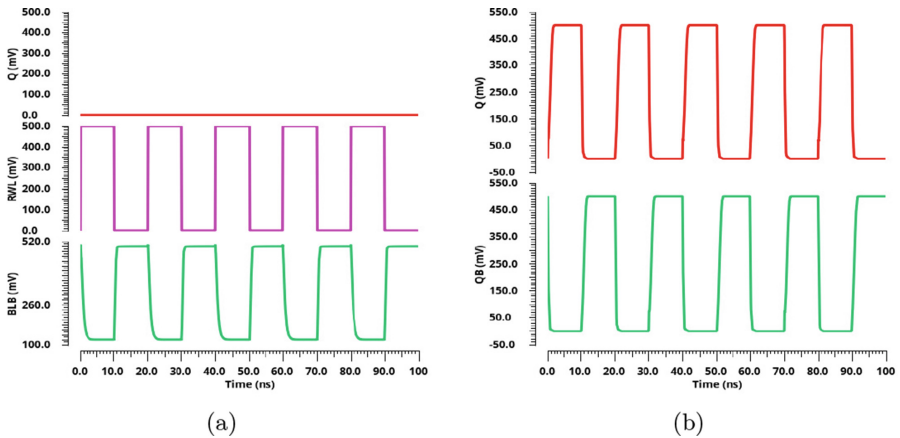


Fig. 3. 9T SRAM cell (a) Transient read wave, (b) Transient Write wave

The 9T SRAM cell is depicted in Fig. 2. Transistors P1, N1 and P2, N2 form two back to back connected inverter to store one bit information in the cell. Transistor N3, N4 and N7 are write and read access transistors respectively. The read decoupled structure is formed by transistors N5 and N6. During read

operation of 9T SRAM cell, RWL signal is at logic ‘1’ while WWL is maintained at logic ‘0’. The bit-line BL and BLB are pre-charge to Vdd. Assume that at the storage node(Q) logic ‘0’ is present due to that transistor P2 is conducted and logic ‘1’ develops at QB. Transistor N6 gate terminal is connected to QB node that turns on the N6 transistor and BLB provides a ground path to discharge the bit-line voltage through N6 and N7 transistors as shown in Fig. 3(a). During write operation WWL signal is at logic ‘1’ and RWL signal is at logic ‘0’. The bit-line BL and BLB acts as input signal and data is written at the storage node Q and QB as depicted in Fig. 3(b). During hold operation WWL and RWL signals is at logic ‘0’ that disconnected the storage node from the bit-line and data is preserved at Q and QB.

3 Simulation Result and Discussion

3.1 Stability

In the design of SRAM cell read static noise margin (RSNM) and write static noise margin (WSNM) are the parameters to define the read and write stability of SRAM cell respectively. In Conv.6T SRAM cell, as the supply voltage reduces the stability of the SRAM cell degrades. It happens due to the voltage divider in between access and pull-down transistors [10].

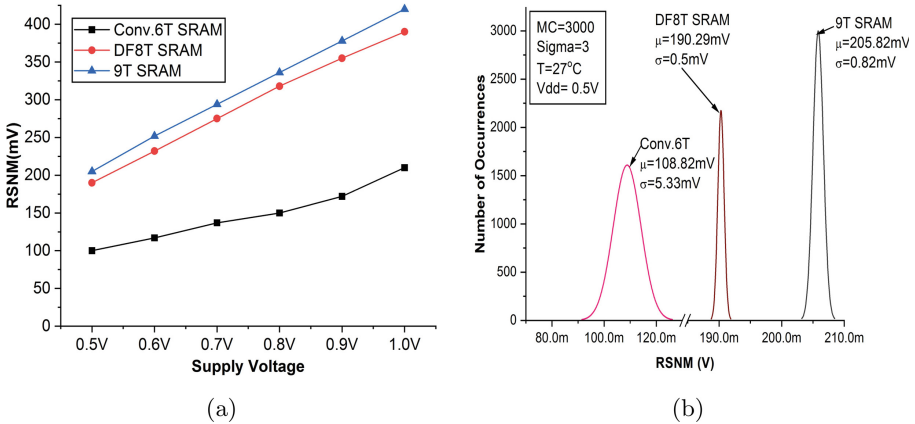


Fig. 4. (a) RSNM of SRAM topologies at different supply voltage, (b) MC simulation of RSNM at 0.5 V supply voltage

It has been observed from Fig. 4(a), that read stability in 9T SRAM cell is $2.05\times$ and $1.07\times$ better as comparison of Conv.6T [3] and DF8T [11] SRAM cell respectively. The process variation tolerance of RSNM is shown in Fig. 4(b). It seems that the 9T SRAM cell has a higher mean value of RSNM at 3000 random samples among the considered cell. It happens due to an isolated read and write

path. The read decoupled eliminate the bit-line disturbances at the storage node that helps to improve the read stability of the cell. The WSNM of 9T SRAM cell is $1.13\times$ and $1.09\times$ better as comparison of Conv.6T [3] and DF8T [11] SRAM cell at 0.5 V supply voltage. It happens due to stronger access transistors (N3 and N4) (Fig. 5).

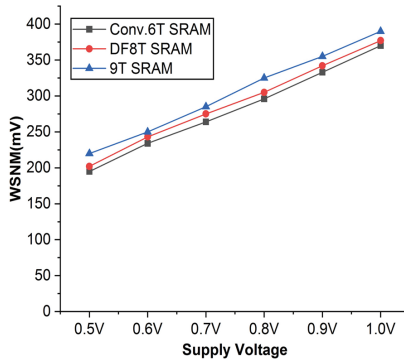


Fig. 5. WSNM at different supply voltage

3.2 Data Retention Voltage

The Data retention voltage (DRV) is characterized as the minimum supply voltage required in the cell for saving information at the storage node [12] (Fig. 6).

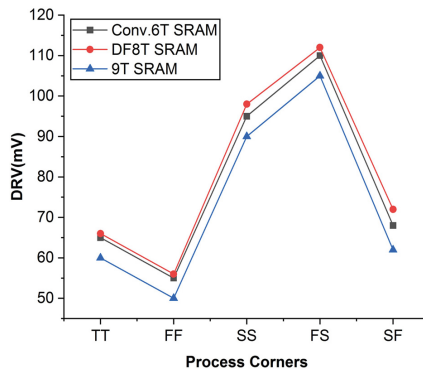


Fig. 6. DRV of SRAM Topologies

The DRV of SRAM cell is determined out at all process corners. In the FS corner (Fast NMOS Slow PMOS) SRAM cells is higher DRV and FF corner (Fast NMOS Fast PMOS) is lower DRV because of variation in PMOS Characteristics. The 9T SRAM cell shows 7.8% and 9.1% better DRV at TT corner as comparison of Conv.6T [3] and DF8T [11] SRAM cell.

3.3 Read/Write Access Time

Read access time (Read delay) is defined as the duration to discharge the read bit line from its higher value by 50 mV as the RWL is activated. The write access time is the time to develop 90% of V_{dd} or 10% of V_{dd} at storage node during write '1' and write '0' respectively [11]. It is depicted in Fig. 7(a).

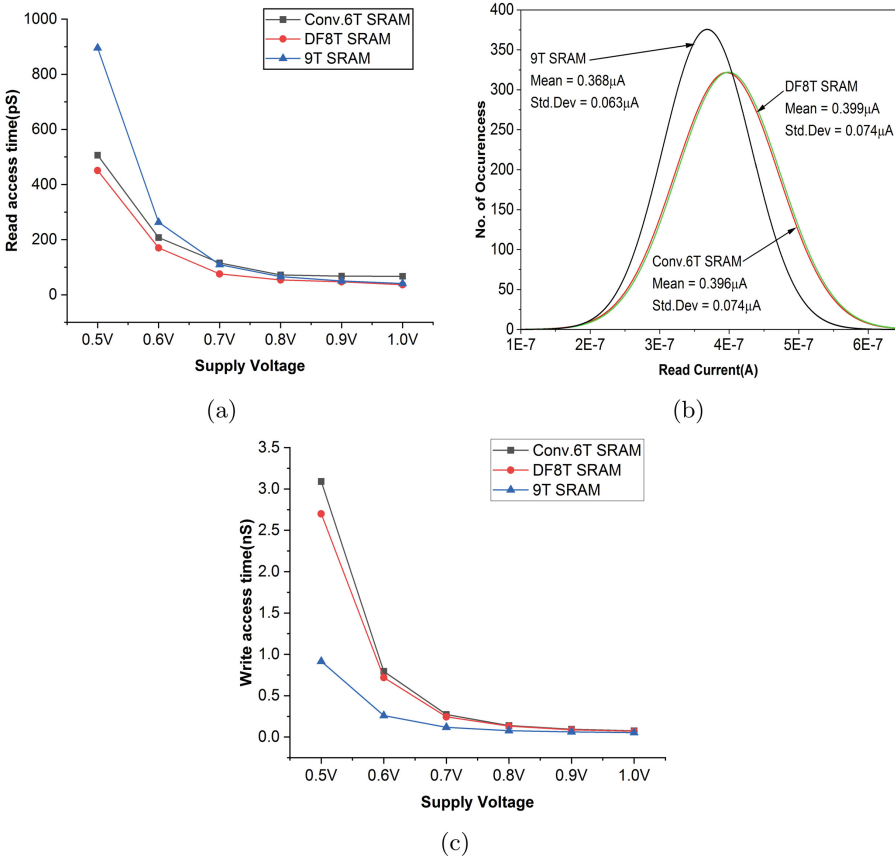


Fig. 7. (a) Read access time at different supply voltage, (b) Read current at 0.5 V supply Voltage, (c) Write access time at different supply voltage

It can be noted that the read delay is higher among all considered cells. The monte-carlo (MC) simulation of read current is depicted in Fig. 7(b). It seems that 9T SRAM cell has lower read current that validates the read access time of 9T SRAM cell is higher. However, it may be noticed from Fig. 7(c) the write access time of 9T SRAM cell is $3.37\times$ and $2.94\times$ better as comparison of Conv.6T [3] and DF8T [11] SRAM cell respectively at 0.5 V supply voltage.

3.4 Read/Write Power Dissipation

Figure 8(a) depicts the plot of average read and write power dissipation for various SRAM cells at different supply voltages. It is notice that the read power of 9T SRAM cell is $1.26\times$ better as comparison of Conv.6T [3] SRAM cell. The write power of 9T SRAM cell is $2.07\times$ and $1.74\times$ better as comparison of Conv.6T and DF8T [11] SRAM cell at 0.5 V supply voltage.

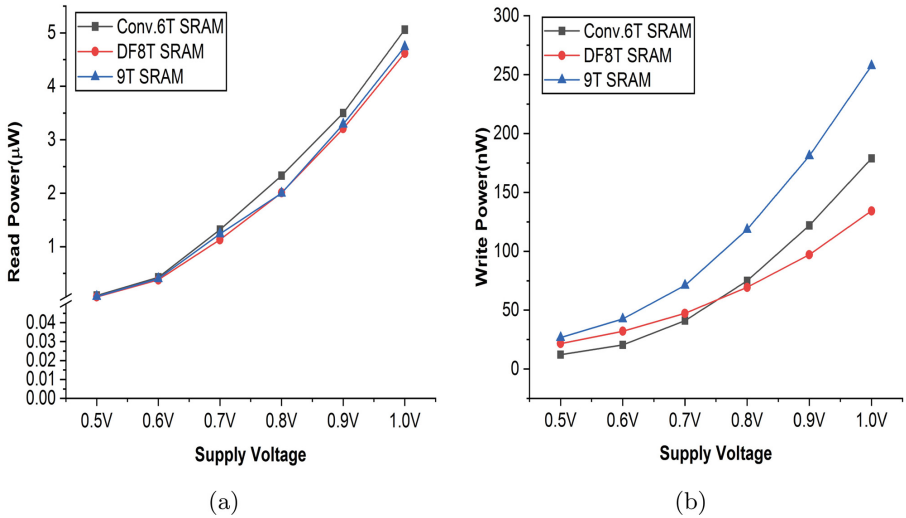


Fig. 8. (a) Read Power of SRAM Topologies, (b) Write Power of SRAM topologies

3.5 Layout of SRAM Cells

Layout area of SRAM cell is characterizing by the number of transistors involved in designing of a SRAM cell. The transistor channel length and width play a very important role to minimize the area of SRAM cell. There is various steps are involved during layout process of SRAM cells. An effective design rule checks (DRC) guarantees that the layout fits with the standards design rule for impeccable fabrication. Layout Versus Schematic (LVS) checking programming software perceives the drawn states of the format that speak to the electrical

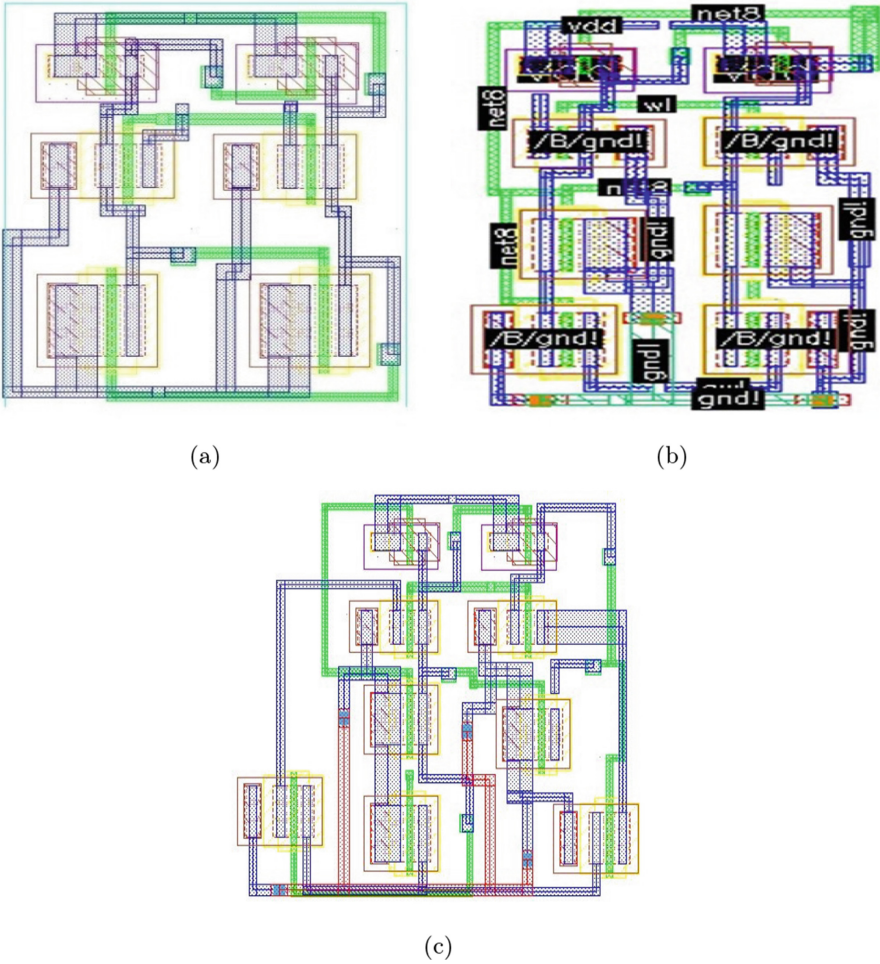


Fig. 9. Layout of (a) Conv.6T SRAM cell, (b) DF8T SRAM cell, (c) Reported 9T SRAM cell

connection between the circuit and input/output pins. Though complexity of the circuit is also play a role for designing a layout of the cell. Figure 9 shows the layout of SRAM cells. It seems that 9T SRAM cell occupies $1.45\times$ and $1.35\times$ larger area as compared to Conv.6T and DF8T SRAM cell respectively.

4 Conclusion

In this paper, 9T SRAM cell with higher stability and low power dissipation is implemented, analyzed and compared with Conv.6T and DF8T SRAM cell. 9T SRAM cell shows an improvement in read stability and write ability in sub-threshold region. Simulation results show that 9T SRAM cell has higher read

stability i.e. 205 mV and higher write ability i.e. 195 mV. The simulated cell shows lowest DRV amongst the considered SRAM topologies. 9T SRAM cell is also required only 60 mV to preserve the data at the storage node. 9T SRAM cell also has lowest write power among the considered cell at 0.5 V supply voltage. This 9T SRAM cell is may be utilized in ultra-low power application such as sensor-node processors, medical equipments.

References

1. Singh, P., Reniwal, B.S., Vijayvargiya, V., Sharma, V., Vishvakarma, S.K.: Ultra low power-high stability, positive feedback controlled (PFC) 10T SRAM cell for look up table (LUT) design. *Integration* **62**, 1–13 (2018)
2. Kumar, H., Tomar, V.K.: Stability analysis of sub-threshold 6T SRAM cell at 45 nm for IoT application. *Int. J. Recent Technol. Eng. (IJRTE)* **8**(2), 2432–2438 (2019)
3. Pavlov, A., Sachdev, M.: *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies: Process-Aware SRAM Design and Test*, vol. 40. Springer, Heidelberg (2008)
4. Verma, N., Chandrakasan, A.P.: A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy. *IEEE J. Solid-State Circuits* **43**(1), 141–149 (2008)
5. Calhoun, B.H., Chandrakasan, A.P.: A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation. *IEEE J. Solid-State Circuits* **42**(3), 680–688 (2007)
6. Takeda, K., et al.: A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications. *IEEE J. Solid-State Circuits* **41**(1), 113–121 (2005)
7. Kulkarni, J.P., Kim, K., Roy, K.: A 160 mv robust schmitt trigger based subthreshold SRAM. *IEEE J. Solid-State Circuits* **42**(10), 2303–2313 (2007)
8. Arya, J.K., Raman, A.: Design of single bitline novel 5T SRAM cell to reduce the power dissipation using cadence. *J. VLSI Des. Tools Technol.* **6**(1), 49–54 (2019)
9. Wang, B., Nguyen, T.Q., Do, A.T., Zhou, J., Je, M., Kim, T.T.-H.: Design of an ultra-low voltage 9T SRAM with equalized bitline leakage and cam-assisted energy efficiency improvement. *IEEE Trans. Circuits Syst. I Regul. Pap.* **62**(2), 441–448 (2014)
10. Sachdeva, A., Tomar, V.K.: Design of a stable low power 11-T static random access memory cell. *J. Circuits Syst. Comput.* **29**, 2050206 (2020)
11. Pal, S., Islam, A.: Variation tolerant differential 8T SRAM cell for ultralow power applications. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **35**(4), 549–558 (2015)
12. Qin, H., Cao, Y., Markovic, D., Vladimirescu, A., Rabaey, J.: Standby supply voltage minimization for deep sub-micron SRAM. *Microelectron. J.* **36**(9), 789–800 (2005)

Author Index

A

Abayomi-Alli, Adebayo, 33
Abayomi-Alli, Olusola, 169
Afroze, Sadia, 458
Agbaegbu, John Bosco, 33
Aggarwal, Mayank, 366
Agwarwal, Arpit, 51
Ahmed, Fatema, 235
Ahuja, Ravin, 33, 169
Al-Hasan, Md., 470
Alonge, Christian Yetunde, 33
Ambigavathi, M., 481
Anwar, Shamama, 278
Arefin, Mohammad Shamsul, 182
Arogundade, Oluwasefunmi 'Tale, 33
Arvind, P., 1

B

Badjate, S. L., 306
Bansal, Saloni, 506
Barisal, Swadhin Kumar, 8
Behera, Sourajit, 134
Beyene, Asrat Mulatu, 105
Brahma, Banalaxmi, 74

C

Chakraborty, Sourav, 1
Chalamalasetti, Yaswanth, 425
Chander, Bhanu, 379
Chatterjee, Avishikta, 447
Chaturvedi, Ankur, 366
Choubey, Anurag, 134, 333
Choudhary, Jaytrilok, 389
Choudhary, Roshani, 193

D

Dada, Mojisola, 33
Damasevicius, Robertas, 169
Das, Subhranil, 1
Dash, Manish Kumar, 297
Deepak Kumar, S., 1
Devidutta, Supriya, 297
Dwivedi, Ashish Kumar, 400

F

Fatima, Sameera, 64

G

Garg, Umang, 51
Gill, Karanjit Singh, 134
Greeshma, R. R., 333
Gupta, Rohit Kumar, 134, 333, 410
Gyanchandani, Manasi, 74

H

Hasan, Mohammad, 470
Hoque, Mohammed Moshil, 158, 458, 494
Hossain, Md. Rajib, 494
Hussain, Mohammad Equebal, 201
Hussain, Muzzammil, 319
Hussain, Rashid, 201

J

Jain, Shlok, 333
Jain, Usha, 319
Jaiswal, Rishabh, 147
Jangamashetti, Sneha, 64
Jayaraj, Mahima, 18

Jena, Debasish, 297
Jeneetaa, Tayenjam, 249

K

Kafi, Hasan Muhammad, 470
Kanani, Chandresh S., 134
Khetawat, Saritha, 235
Kirti, Shivangi, 85
Kishore, Pushkar, 8
Krishna, Kushagra, 389
Kumar, Harekrishna, 517
Kumar, Kulamala Vinod, 436, 447
Kumar, Sachin, 51
Kumaravelan., 379
Kumari, Priyanka, 436
Kumari, Rashmi, 1

L

Laishram, Debina, 249
Lilhare, Monika, 211

M

Maheshwari, Drashti, 366
Mangal, Akansha, 259
Manikandan, M. Sabarimalai, 425
Maru, Harsh, 225
Miryala, Yashwanth, 225
Mishra, Preeti, 51
Mishra, Shivendu, 96
Mishra, Vartika, 211
Misra, Rajiv, 85, 96, 134, 147, 333, 410
Misra, Sanjay, 33, 169
Mohanta, Bhabendu Kumar, 297
Mohapatra, Durga Prasad, 8, 436, 447
Moid, Md Ashraf, 410

O

Odusami, Modupe, 169
Oseni, Olaoluwa, 169

P

Pandey, Savyasachi, 147
Parikh, Satyen M., 348
Patel, Hiral, 348
Patel, Yashwant Singh, 85, 147
Pradhan, Rahul, 366
Pranav, P., 225
Pruthvi, M. G., 64

Q

Qureshi, Saman, 121

R

Rabdo, Beza Mamo, 105
Raghavendra, C. G., 45
Raghu Srivatsa, M. P., 45
Ramkumar, Barathram, 425
Ranjan, Amit, 410
Rashid, Md Mamunur, 470
Rastogi, Aksh, 287
Rath, Santanu Kumar, 211
Reddy, Alle Giridhar, 8
Rewatkar, Rajendra, 306
Roy, Animesh Chandra, 182
Roy, Bholanath, 235
Rudra, Bhawana, 225
Rudraswamy, S. B., 64

S

Sahoo, Sudhir Kumar, 425
Sakib, M. A. Muhaimin, 158
Sanober, Adla, 278
Saritha, Sri. Khetwat, 121
Satapathy, Shashank Mouli, 400
Sathya, M., 64
Shali, Yedhu, 74
Sharaff, Aakanksha, 400
Sharif, Omar, 158
Sharma, Dilip Kumar, 366
Shukla, Sanyam, 193, 259
Singh, Dharendra Pratap, 389
Singh, Krishan Pal, 319
Singh, Ranjeet Kumar, 267
Singh, Saurabh, 354
Sinha, Ankit, 51
Sridharan, D., 481
Suthar, Amit, 348

T

Tiwari, Namita, 354
Tomar, Shubham, 319
Tomar, V. K., 287, 506, 517
Tuithung, Themrichon, 249

W

Wadhvani, Rajesh, 74