

Studies in Autonomic,  
Data-driven and Industrial Computing

Mohiuddin Ahmad  
Mohammad Shorif Uddin  
Yeong Min Jang *Editors*

# Proceedings of International Conference on Information and Communication Technology for Development

ICICTD 2022

 Springer

# **Studies in Autonomic, Data-driven and Industrial Computing**

## **Series Editors**

Swagatam Das, Indian Statistical Institute, Kolkata, West Bengal, India

Jagdish Chand Bansal, South Asian University, Chanakyapuri, India

The book series Studies in Autonomic, Data-driven and Industrial Computing (SADIC) aims at bringing together valuable and novel scientific contributions that address new theories and their real world applications related to autonomic, data-driven, and industrial computing. The area of research covered in the series includes theory and applications of parallel computing, cyber trust and security, grid computing, optical computing, distributed sensor networks, bioinformatics, fuzzy computing and uncertainty quantification, neurocomputing and deep learning, smart grids, data-driven power engineering, smart home informatics, machine learning, mobile computing, internet of things, privacy preserving computation, big data analytics, cloud computing, blockchain and edge computing, data-driven green computing, symbolic computing, swarm intelligence and evolutionary computing, intelligent systems for industry 4.0, as well as other pertinent methods for autonomic, data-driven, and industrial computing.

The series will publish monographs, edited volumes, textbooks and proceedings of important conferences, symposia and meetings in the field of autonomic, data-driven and industrial computing.

Mohiuddin Ahmad · Mohammad Shorif Uddin ·  
Yeong Min Jang  
Editors

# Proceedings of International Conference on Information and Communication Technology for Development

ICICTD 2022



*Editors*

Mohiuddin Ahmad  
Institute of Information  
and Communication Technology  
Khulna University of Engineering &  
Technology  
Khulna, Bangladesh

Mohammad Shorif Uddin  
Department of Computer Science  
and Engineering  
Jahangirnagar University  
Dhaka, Bangladesh

Yeong Min Jang  
School of Electrical Engineering  
Kookmin University  
Seoul, Korea (Republic of)

ISSN 2730-6437

ISSN 2730-6445 (electronic)

Studies in Autonomic, Data-driven and Industrial Computing

ISBN 978-981-19-7527-1

ISBN 978-981-19-7528-8 (eBook)

<https://doi.org/10.1007/978-981-19-7528-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

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

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

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

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.

The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

# Preface

This book gathers outstanding research articles as the proceedings at the 1st International Conference on Information and Communication Technology for Development (ICICTD 2022), which was organized by the Institute of Information and Communication Technology (IICT), Khulna University of Engineering & Technology (KUET), Bangladesh, during July 29–30, 2022, in both virtual and physical modes due to the COVID-19 pandemic. The conference was conceived as a platform for congenial networking, disseminating and exchanging ideas, concepts, and results of researchers from academia and industry in the field of Information and Communication Technology (ICT) from the further developmental perspective. The conference focused on data and information security, communication network, natural language processing, collective intelligence, soft computing, optimization, cloud computing, machine learning, intelligent software, robotics, data science, and big data analytics.

We have tried our best to ensure the quality of the ICICTD 2022 proceedings book through the stringent and careful peer-review process of the submitted manuscripts. ICICTD 2022 received a significant number (239) of technical contributed articles from distinguished participants from home and abroad. After a very stringent peer-reviewing process, only 43 high-quality articles were finally accepted for presentation, and the final proceedings book contains these articles as chapters.

In fact, this book presents novel contributions in areas of ICT and it serves as reference material for advanced research.

Khulna, Bangladesh  
Dhaka, Bangladesh  
Seoul, South Korea

Mohiuddin Ahmad  
Mohammad Shorif Uddin  
Yeong Min Jang

# About This Book

This book gathers outstanding research papers presented at the 1st International Conference on Information and Communication Technology for Development (ICICTD 2022), which was held during July 29–30, 2022, at Khulna University of Engineering & Technology (KUET), Bangladesh, in both virtual and physical modes due to the COVID-19 pandemic. ICICTD 2022 was organized by the Institute of Information and Communication Technology (IICT), Khulna University of Engineering & Technology (KUET), Bangladesh. The conference is conceived as a platform for congenial networking, disseminating and exchanging ideas, concepts, and results of researchers from academia and industry in the field of Information and Communication Technology (ICT) from the further developmental perspective. We have tried our best to ensure the quality of the ICICTD 2022 proceedings book through the stringent and careful peer-review process of the submitted manuscripts. In fact, this book presents novel contributions in areas of ICT and it serves as reference material for advanced research.

The topics covered are data and information security, communication network, natural language processing, collective intelligence, soft computing, optimization, cloud computing, machine learning, intelligent software, robotics, data science, and big data analytics.

Khulna, Bangladesh  
Dhaka, Bangladesh  
Seoul, South Korea

Mohiuddin Ahmad  
Mohammad Shorif Uddin  
Yeong Min Jang

# Contents

|          |                                                                                                                                                                     |           |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Notes on Image Watermarking and Steganography</b> .....                                                                                                          | <b>1</b>  |
|          | Mahbuba Begum and Mohammad Shorif Uddin                                                                                                                             |           |
| <b>2</b> | <b>Vehicle Detection Using Deep Learning Method and Adaptive and Dynamic Automated Traffic System via IoT Using Surveillance Camera</b> .....                       | <b>15</b> |
|          | Nafiz Arman, Syed Rakib Hasan, and Md. Ruhul Abedin                                                                                                                 |           |
| <b>3</b> | <b>Predicting Average Localization Error of Underwater Wireless Sensors via Decision Tree Regression and Gradient Boosted Regression</b> .....                      | <b>29</b> |
|          | Md. Mostafizur Rahman and Sumiya Akter Nisher                                                                                                                       |           |
| <b>4</b> | <b>Machine Learning Approaches for Classification and Diameter Prediction of Asteroids</b> .....                                                                    | <b>43</b> |
|          | Mir Sakhawat Hossain and Md. Akib Zabed                                                                                                                             |           |
| <b>5</b> | <b>Detection of the Most Essential Characteristics from Blood Routine Tests to Increase COVID-19 Diagnostic Capacity by Using Machine Learning Algorithms</b> ..... | <b>57</b> |
|          | Faria Rahman and Mohiuddin Ahmad                                                                                                                                    |           |
| <b>6</b> | <b>Usability Evaluation of Adaptive Learning System Rhapsode<sup>TM</sup> Learner</b> .....                                                                         | <b>71</b> |
|          | Md. Saifuddin Khalid, Tobias Alexander Bang Tretow-Fish, and Amalie Roark                                                                                           |           |
| <b>7</b> | <b>A Machine Learning and Deep Learning Approach to Classify Mental Illness with the Collaboration of Natural Language Processing</b> .....                         | <b>83</b> |
|          | Md. Rafin Khan, Shadman Sakib, Adria Binte Habib, and Muhammad Iqbal Hossain                                                                                        |           |

|           |                                                                                                                                                   |            |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>8</b>  | <b>Data Governance and Digital Transformation in Saudi Arabia</b> . . . .                                                                         | <b>95</b>  |
|           | Kholod Saaed Al-Qahtani and M. M. Hafizur Rahman                                                                                                  |            |
| <b>9</b>  | <b>A Waveguide-Fed Hybrid Graphene Plasmonic Nanoantenna for On-Chip Wireless Optical Communication</b> . . . . .                                 | <b>107</b> |
|           | Richard Victor Biswas                                                                                                                             |            |
| <b>10</b> | <b>Effectiveness of Transformer Models on IoT Security Detection in StackOverflow Discussions</b> . . . . .                                       | <b>125</b> |
|           | Nibir Chandra Mandal, G. M. Shahariar, and Md. Tanvir Rouf Shawon                                                                                 |            |
| <b>11</b> | <b>Digital Health Inclusion to Achieve Universal Health Coverage in the Era of 4th Industrial Revolution</b> . . . . .                            | <b>139</b> |
|           | Farhana Sarker, Moinul H. Chowdhury, Rony Chowdhury Ripan, Tanvir Islam, Rubaiyat Alim Hridhee, A. K. M. Nazmul Islam, and Khondaker A. Mamun     |            |
| <b>12</b> | <b>Design of 8 × 8 Microstrip Array Antenna for ISM and K-Band Applications</b> . . . . .                                                         | <b>153</b> |
|           | Prodip Kumar Saha Purnendu, Afrin Binte Anwar, Rima Islam, Debalina Mollik, Md. Anwar Hossain, and Mohiuddin Ahmad                                |            |
| <b>13</b> | <b>Spatial-Spectral Kernel Convolutional Neural Network-Based Subspace Detection for the Task of Hyperspectral Image Classification</b> . . . . . | <b>163</b> |
|           | Md. Hafiz Ahamed and Md. Ali Hossain                                                                                                              |            |
| <b>14</b> | <b>Sentiment Analysis and Emotion Detection from Khasi Text Data—A Survey</b> . . . . .                                                           | <b>171</b> |
|           | Banteilang Mukhim and Sufal Das                                                                                                                   |            |
| <b>15</b> | <b>Classification of Brain Hemorrhage Using Deep Learning from CT Scan Images</b> . . . . .                                                       | <b>181</b> |
|           | Nipa Anjum, Abu Noman Md. Sakib, and Sk. Md. Masudul Ahsan                                                                                        |            |
| <b>16</b> | <b>An Automated Remote Health Monitoring System in IoT Facilities</b> . . . . .                                                                   | <b>195</b> |
|           | Pallab Kumar Nandi and Mohiuddin Ahmad                                                                                                            |            |
| <b>17</b> | <b>Surface Reflectance: A Metric for Untextured Surgical Scene Segmentation</b> . . . . .                                                         | <b>209</b> |
|           | Shahnewaz Ali, Yaqub Jonmohamadi, Yu Takeda, Jonathan Roberts, Ross Crawford, Cameron Brown, and Ajay K. Pandey                                   |            |

|           |                                                                                                                                         |            |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>18</b> | <b>Classification and Segmentation on Multi-regional Brain Tumors Using Volumetric Images of MRI with Customized 3D U-Net Framework</b> | <b>223</b> |
|           | Md. Faysal Ahamed, Md. Robiul Islam, Tahmim Hossain, Khalid Syfullah, and Ovi Sarkar                                                    |            |
| <b>19</b> | <b>Explainable AI and Ensemble Learning for Water Quality Prediction</b>                                                                | <b>235</b> |
|           | Nakayiza Hellen, Hasibul Hasan Sabuj, and Md. Ashraful Alam                                                                             |            |
| <b>20</b> | <b>Diagnosis of Autism Spectrum Disorder Through Eye Movement Tracking Using Deep Learning</b>                                          | <b>251</b> |
|           | Nasirul Mumenin, Md. Farhadul Islam, Md. Reasad Zaman Chowdhury, and Mohammad Abu Yousuf                                                |            |
| <b>21</b> | <b>Real-Time Lightweight Bangla Sign Language Recognition Model Using Pre-trained MobileNetV2 and Conditional DCGAN</b>                 | <b>263</b> |
|           | Abdullah Al Rafi, Rakibul Hassan, Md. Rabiul Islam, and Md. Nahiduzzaman                                                                |            |
| <b>22</b> | <b>A Novel Method of Thresholding for Brain Tumor Segmentation and Detection</b>                                                        | <b>277</b> |
|           | Tanber Hasan Shemanto, Lubaba Binte Billah, and Md. Abrar Ibtesham                                                                      |            |
| <b>23</b> | <b>Developing Wearable Human–Computer Interfacing System Based on EMG and Gyro for Amputees</b>                                         | <b>291</b> |
|           | Md. Rokib Raihan and Mohiuddin Ahmad                                                                                                    |            |
| <b>24</b> | <b>Implementation of Twisted Edwards Curve-Based Unified Group Operation for ECC Processor</b>                                          | <b>305</b> |
|           | Asif Faisal Khan and Md. Jahirul Islam                                                                                                  |            |
| <b>25</b> | <b>IoT-Based Smart Office and Parking System with the Implementation of LoRa</b>                                                        | <b>317</b> |
|           | Robi Paul and Junayed Bin Nazir                                                                                                         |            |
| <b>26</b> | <b>Bandwidth Borrowing Technique for Improving QoS of Cluster-Based PON System</b>                                                      | <b>331</b> |
|           | Mehedi Hasan, Sujit Basu, and Monir Hossen                                                                                              |            |
| <b>27</b> | <b>Ensemble of Boosting Algorithms for Parkinson Disease Diagnosis</b>                                                                  | <b>343</b> |
|           | Maksuda Rahman, Md. Kamrul Hasan, Masshura Mayashir Madhurja, and Mohiuddin Ahmad                                                       |            |

|           |                                                                                                                                     |            |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>28</b> | <b>GLCM and HOG Feature-Based Skin Disease Detection Using Artificial Neural Network</b>                                            | <b>355</b> |
|           | Nymphia Nourin, Paromita Kundu, Sk. Saima, and Md. Asadur Rahman                                                                    |            |
| <b>29</b> | <b>Subject Dependent Cognitive Load Level Classification from fNIRS Signal Using Support Vector Machine</b>                         | <b>365</b> |
|           | Syeda Umme Ayman, Al Arrafuzzaman, and Md. Asadur Rahman                                                                            |            |
| <b>30</b> | <b>Smoke Recognition in Smart Environment Through IoT Air Quality Sensor Data and Multivariate Logistic Regression</b>              | <b>379</b> |
|           | S. M. Mohidul Islam and Kamrul Hasan Talukder                                                                                       |            |
| <b>31</b> | <b>Learning from Learned Network: An Introspective Model for Arthroscopic Scene Segmentation</b>                                    | <b>393</b> |
|           | Shahnewaz Ali, Feras Dayoub, and Ajay K. Pandey                                                                                     |            |
| <b>32</b> | <b>A Multigene Genetic Programming Approach for Soil Classification and Crop Recommendation</b>                                     | <b>407</b> |
|           | Ishrat Khan and Pintu Chandra Shill                                                                                                 |            |
| <b>33</b> | <b>Soft Error Tolerant Memristor-Based Memory</b>                                                                                   | <b>421</b> |
|           | Muhammad Sheikh Sadi, Md. Mehedy Hasan Sumon, and Md. Liakot Ali                                                                    |            |
| <b>34</b> | <b>Efficient Malaria Cell Image Classification Using Deep Convolutional Neural Network</b>                                          | <b>435</b> |
|           | Sohag Kumar Mondal, Monira Islam, Md. Omar Faruque, Mrinmoy Sarker Turja, and Md. Salah Uddin Yusuf                                 |            |
| <b>35</b> | <b>Detection of Acute Myeloid Leukemia from Peripheral Blood Smear Images Using Transfer Learning in Modified CNN Architectures</b> | <b>447</b> |
|           | Jeba Fairouz Rahman and Mohiuddin Ahmad                                                                                             |            |
| <b>36</b> | <b>Automatic Bone Mineral Density Estimation from Digital X-ray Images</b>                                                          | <b>461</b> |
|           | Abdullah Al Mahmud and Kalyan Kumar Halder                                                                                          |            |
| <b>37</b> | <b>Machine Learning Algorithms for the Prediction of Prostate Cancer</b>                                                            | <b>471</b> |
|           | M. M. Imran Molla, Julakha Jahan Jui, Humayan Kabir Rana, and Nitun Kumar Podder                                                    |            |
| <b>38</b> | <b>Detection of Ventricular Fibrillation from ECG Signal Using Hybrid Scalogram-Based Convolutional Neural Network</b>              | <b>483</b> |
|           | Md. Faisal Mina, Amit Dutta Roy, and Md. Bashir Uddin                                                                               |            |

|           |                                                                                                                                               |            |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>39</b> | <b>Lung Cancer Detection Using Ensemble Technique of CNN</b> .....                                                                            | <b>497</b> |
|           | Zebel-E-Noor Akhand, Afridi Ibn Rahman, Anirudh Sarda,<br>Md. Zubayer Ahmed Fahim, Lubaba Tasnia Tushi, Katha Azad,<br>and Hiya Tasfia Tahiat |            |
| <b>40</b> | <b>Identification of the Similarity of Bangla Words Using<br/>Different Word Embedding Techniques</b> .....                                   | <b>509</b> |
|           | Aroni Saha Prapty and K. M. Azharul Hasan                                                                                                     |            |
| <b>41</b> | <b>Energy Consumption Optimization of Zigbee Communication:<br/>An Experimental Approach with XBee S2C Module</b> .....                       | <b>521</b> |
|           | Rifat Zabin and Khandaker Foysal Haque                                                                                                        |            |
| <b>42</b> | <b>PredXGBR: A Machine Learning Based Short-Term<br/>Electrical Load Forecasting Architecture</b> .....                                       | <b>535</b> |
|           | Rifat Zabin, Labanya Barua, and Tofael Ahmed                                                                                                  |            |
| <b>43</b> | <b>Sn Doped <math>\text{Ge}_x\text{Si}_{1-x}\text{O}_y</math> Films for Uncooled Infrared Detections</b> .....                                | <b>547</b> |
|           | Jaime Cardona, Femina Vadakepurathu, and Mukti Rana                                                                                           |            |
|           | <b>Author Index</b> .....                                                                                                                     | <b>557</b> |



## About the Editors

**Prof. Mohiuddin Ahmad** received his B.Sc. Engineering degree in Electrical and Electronic Engineering from Chittagong University of Engineering and Technology (CUET), Bangladesh securing First Class First with Honours grade, and his MS degree in Electronics and Information Science (Major Biomedical Engineering) from Kyoto Institute of Technology of Japan in 1994 and 2001, respectively. Dr. Ahmad was awarded the Institute Gold Medal for Securing the Highest Grade in his Batch by the Prime Minister of Bangladesh. He received a Ph.D. degree in Computer Science and Engineering from Korea University, Republic of Korea, in 2008. From November 1994 to August 1995, he served as a part-time Lecturer in the Department of Electrical and Electronic Engineering at CUET, Bangladesh. From August 1995 to October 1998, he served as a Lecturer in the Department of Electrical and Electronic Engineering at Khulna University of Engineering & Technology, Bangladesh. In June 2001, he joined the same Department as an Assistant Professor. In May 2009, he joined the same Department as an Associate Professor and now he is a Full Professor. Moreover, Dr. Ahmad served as the Head of the Department of Biomedical Engineering from October 2009 to September 2012. Furthermore, Prof. Ahmad served as the Head of the Department of Electrical and Electronic Engineering from September 2012 to August 2014. Currently, he is serving as a Full Professor in the Department of Electrical and Electronic Engineering. Prof. Ahmad was the Sub-Project Manager of a project funded by the World Bank and Coordinated by UGC, Bangladesh. The project title was “Postgraduate Research in BME” and was named Higher Education Quality Enhancement Project (HEQEP) Sub-Project, CP#3472 from July 2014 to June 2017. Prof. Ahmad served as the Dean of the Faculty of Electrical and Electronic Engineering from August 14, 2018, to August 13, 2020. Currently, Prof. Ahmad has been serving as the Director of the Institute of Information and Communication Technology (IICT), KUET. He had acted as a General Chair or TPC Chair or Co-chair of many IEEE-indexed International conferences and Springer Nature—indexed Conference, such as EICT 2015, EICT2017, EICT 2019, ICICTD 2022, etc. Moreover, Prof. Ahmad supervised a good number of Doctoral and Master’s Theses. Three Ph.D. degrees and Twenty M.Sc. Engineering Degree was awarded by his supervision and currently, he has been supervising many PG

students. His research interests include Biomedical Signal and Image Processing for disease analysis, Computer Vision and Pattern Recognition with AI, ICT, and IoT in Healthcare, Human Motion Analysis, Circuits and Systems, and Energy Conversion. He has published more than 165 research papers in international journals and conference proceedings. Prof. Ahmad is a life fellow of IEB, a Member of IEEE, and a senior member of the International Association of Computer Science and Information Technology. Prof. Dr. Mohiuddin Ahmad is the founder and president of the “Clinical Engineering Association—Bangladesh (CEAB)” —A non-political, non-profit medical equipment-based patient care social organization where Dr. Ahmad oversees the policies and strategic directions of CEAB as a whole.

**Prof. Mohammad Shorif Uddin** completed his Doctor of Engineering (Ph.D.) at Kyoto Institute of Technology in 2002, Japan, Master of Technology Education at Shiga University, Japan in 1999, Bachelor of Electrical and Electronic Engineering at Bangladesh University of Engineering and Technology (BUET) in 1991 and also Master of Business Administration (MBA) from Jahangirnagar University in 2013. He began his teaching career as a Lecturer in 1991 at Chittagong University of Engineering and Technology (CUET). In 1992, he joined the Computer Science and Engineering Department of Jahangirnagar University and at present, he is a Professor of this department. Besides, he is the Teacher-in-Charge of the ICT Cell of Jahangirnagar University. He served as the Chairman of the Computer Science and Engineering Department of Jahangirnagar University from June 2014 to June 2017. He worked as an Adviser of ULAB from September 2009 to October 2020 and Hamdard University Bangladesh from November 2020 to November 2021. He undertook postdoctoral research at Bioinformatics Institute, Singapore, Toyota Technological Institute, Japan and Kyoto Institute of Technology, Japan, Chiba University, Japan, Bonn University, Germany, Institute of Automation, Chinese Academy of Sciences, China. His research is motivated by applications in the fields of artificial intelligence, imaging informatics, and computer vision. He holds two patents for his scientific inventions and has published more than 170 research papers in international journals and conference proceedings. In addition, he edited a good number of books and wrote many book chapters. He had delivered a remarkable number of keynotes and invited talks and also acted as a General Chair or TPC Chair or Co-chair of many international conferences. He received the Best Paper award in the International Conference on Informatics, Electronics and Vision (ICIEV2013), Dhaka, Bangladesh, and the Best Presenter Award from the International Conference on Computer Vision and Graphics (ICCVG 2004), Warsaw, Poland. He was the Coach of Janhangirnagar University ACM ICPC World Finals Teams in 2015 and 2017 and supervised a good number of doctoral and Master theses. He is currently the President of Bangladesh Computer Society (BCS), a Fellow of IEB and BCS, a Senior Member of IEEE, and an Associate Editor of IEEE Access.

**Prof. Yeong Min Jang** (Member, IEEE) received the B.E. and M.E. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1985 and 1987, respectively, and the Doctoral degree in computer science from

the University of Massachusetts, Amherst, MA, USA, in 1999. He was with the Electronics and Telecommunications Research Institute from 1987 to 2000. Since 2002, he has been with the School of Electrical Engineering, Kookmin University, Seoul, South Korea, where he was the Director of the Ubiquitous IT Convergence Center in 2005 and 2010, and has been the Director of the LED Convergence Research Center since 2010 and the Director of the Internet of Energy Research Center since 2018. He is currently a Life Member of the Korean Institute of Communications and Information Sciences. His research interests include 5G/6G mobile communications, Internet of energy, IoT platform, AI platform, eHealth, smart factory, optical wireless communications, optical camera communication, and the Internet of Things. Prof. Jang has organized several conferences and workshops, such as the International Conference on Ubiquitous and Future Networks from 2009 to 2017, the International Conference on ICT Convergence from 2010 to 2016, the International Conference on AI in Information and Communication from 2019 to 2021, the International Conference on Information Networking in 2015, and the International Workshop on Optical Wireless LED Communication Networks from 2013 to 2016. He was the recipient of the Young Science Award from the Korean Government from 2003 to 2006. He served as the Founding Chair of the KICS Technical Committee on Communication Networks in 2007 and 2008. He was the Executive Director of KICS from 2006 to 2014, the Vice President of KICS from 2014 to 2016, and an Executive Vice President of KICS in 2018. He was the President of KICS in 2019. He serves as the Co-Editor-in-Chief for the *ICT Express* (Elsevier). He was the Steering Chair of the Multi-Screen Service Forum from 2011 to 2019 and has been the Steering Chair of the Society Safety System Forum since 2015. He served as the Chairman of the IEEE 802.15 Optical Camera Communications Study Group in 2014 and the IEEE 802.15.7m Optical Wireless Communications TG. He successfully published IEEE 802.15.7–2018 standard. He is currently the Chairman of IEEE 802.15 VAT IG.

# Chapter 1

## Notes on Image Watermarking and Steganography



Mahbuba Begum  and Mohammad Shorif Uddin 

**Abstract** Digital information can be reproduced, duplicated, and distributed easily which makes information hiding like image watermarking and steganography as powerful security techniques. Image watermarking and steganography techniques make the embedding hidden data unnoticeable where the sender's identity is hidden by image watermarking while the existence message is concealed by image steganography. This paper gives a basic framework for image watermarking and steganography. Then it figures out some basic differences based on the existing research works on image watermarking and steganography. After then, it highlights state-of-the-art methods based on basic design requirements. Some challenges along with suggestions are given, and a future research direction is provided.

**Keywords** Imperceptibility · Robustness · DCT · DWT · SVD

## 1 Introduction

Information hiding is one of the data hiding techniques that protects data from unauthorized access as the information is shared through Internet and being accessed by unauthorized person. Image watermarking and steganography are the information hiding techniques that ensure this authentication. In this section, we have given brief review about image watermarking, steganography, and their basic comparison. The main contributions of this paper are:

- We identify the basic differences between image watermarking and steganography.
- We identify the affecting factors of image watermarking and steganography.

---

M. Begum (✉)

Mawlana Bhashani Science and Technology University, Tangail 1902, Bangladesh  
e-mail: [mahbuba327@yahoo.com](mailto:mahbuba327@yahoo.com)

M. Begum · M. S. Uddin

Jahangirnagar University, Dhaka 1342, Bangladesh  
e-mail: [shorifuddin@juniv.edu](mailto:shorifuddin@juniv.edu)

- We highlight the state-of-the-art methods based on basic design requirements.
- We focus the existing challenges and corresponding suggestions to overcome these challenges.

## 1.1 Image Watermarking

Digital image watermarking is the watermarking of a digital image. This technique identifies the authenticity of the host image and ensures the security of that image [1]. An additional image is embedded into the host image by using watermark embedding algorithm. The embedded image or watermark can be extracted using the watermark extraction algorithm. Watermarking technique can be either visible or invisible. Thus, the embedding function consists of host, watermark, and secret key. Besides, the watermark extraction algorithm consists of watermarked image and secret key.

Watermark Embedding Technique:

$$\text{Watermarked Image} = \text{Function} (\text{Host}, \text{Watermark}, \text{Secret Key})$$

Watermark Extraction Technique:

$$\text{Watermark Image} = \text{Function} (\text{Watermarked Image}, \text{Key})$$

In image watermarking, at first the watermark image is encrypted by the secret key. Then, the embedding algorithm takes the host image and the encrypted watermark image. Thus, the watermarked image is generated and affected by several attacks when it passed over the communication channel. The watermark extraction algorithm extracts the embedded watermark image using the same key. The basic framework of image watermarking is shown in Fig. 1.

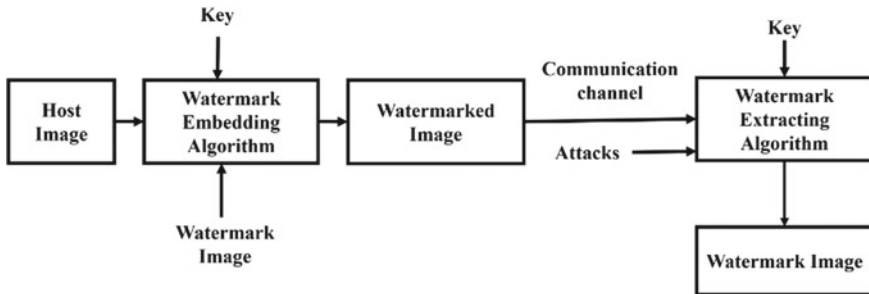


**Fig. 1** Framework of image watermarking [2]

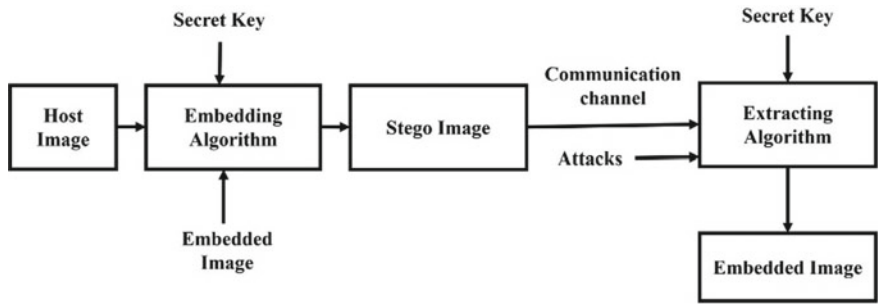


Fig. 2 Framework of image steganography

1.2 Image Steganography

Image steganography hides an image into the host image and is typically invisible to the human eyes [3]. The embedding function consists of host, embedded image, and the secret key, while the extraction function consists of stego image and the same key.

Embedding Method:

$$\text{Stego Image} = \text{Function} (\text{Host}, \text{Embedded Image}, \text{Secret Key})$$

Extraction Method:

$$\text{Embedded Image} = \text{Function} (\text{Stego Image}, \text{Key})$$

Image steganography technique takes the host image, embedded image, and the secret key. Thus, the stego image is generated by using the embedding algorithm. Then, it passes over the communication channel. After then, it is affected by several intentional and unintentional attacks. The embedded image is extracted by the extraction algorithm and same secret key. The basic framework of image steganography is shown in Fig. 2.

1.3 Basic Comparison Between Image Watermarking and Steganography

The basic differences between image watermarking and steganography are shown in Table 1.

**Table 1** Basic comparison between image watermarking and steganography

| Factors           | Image watermarking                                                                     | Image steganography                                           |
|-------------------|----------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Definition        | Hides the sender’s identity of an image                                                | Conceals the existence of an image                            |
| Goal              | Cannot remove or replace the secret image from the watermarked image                   | Cannot detect the presence of the secret image in stego image |
| Input             | Image, video, and others                                                               | Digital content                                               |
| Output            | Watermarked image                                                                      | Stego image                                                   |
| Secret image      | Watermark                                                                              | Payload                                                       |
| Insertion media   | Inserts logo, pattern, and a message                                                   | Inserts host image, audio, or video                           |
| Types             | Robust or fragile, visible or invisible                                                | Invisible                                                     |
| Imperceptibility  | Imperceptible                                                                          | Must be good                                                  |
| Robustness        | Essential requirement                                                                  | Required                                                      |
| Payload capacity  | Reasonable                                                                             | Very high                                                     |
| Affecting factors | Location-where to embed, preprocessing-how to modify host image to embed the watermark | Secret key, capacity, peak-signal-to-noise (PSNR) ratio       |

## 2 Related Works of Image Watermarking and Steganography

Recently, there have been done a lot of works on image watermarking and steganography. These works include optimization, machine learning, Internet of Things (IoT) and cloud, and blockchain-based techniques. Optimization algorithm finds the optimum solution for the system so that a better trade-off between imperceptibility and robustness must be maintained [4]. In machine learning-based approach, machine learns the data and matches the learned data with the test data based on the decisions [5]. In IoT-based approach, objects exchange data through the Internet [6]. Besides, in cloud, information can be accessed remotely using the Internet. In blockchain, data is stored into the block. Hence, anyone cannot manipulate the data. Recent developments of image watermarking and steganography are shown in Table 2.

## 3 Design Requirements of Image Watermarking and Steganography

For developing an effective image watermarking system, it should fulfill some criteria. These criteria must meet up some requirements based on the applications. Therefore, image watermarking must satisfy the five (05) design requirements like imperceptibility, robustness, security, secret key, and embedding capacity.

**Table 2** Recent developments of image watermarking and steganography

| Techniques       | Image watermarking                                                                                                                                                                                                                                                                                                                                                                                                                                              | Image steganography                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Optimization     | <ul style="list-style-type: none"> <li>• Recently [7–9] methods are used for optimization</li> <li>• These methods ensure better imperceptibility and strong robustness against several attacks</li> <li>• But, they show less robustness against cropping, rotation, and clipping attacks</li> </ul>                                                                                                                                                           | <ul style="list-style-type: none"> <li>• Recently [10, 11] methods have been developed for optimization</li> <li>• Improves the embedding capacity and minimizes distortion of the host image</li> <li>• Embeds into the good location of the host image</li> <li>• Eliminates the Gaussian noise</li> </ul>                                                                                                                                                  |
| Machine learning | <ul style="list-style-type: none"> <li>• Methods [12, 13] have been developed using machine learning</li> <li>• Increases the imperceptibility of the watermarked image</li> <li>• Works with any quality image</li> <li>• Ensures better trade-off between robustness and imperceptibility</li> <li>• Shows less robustness against cropping, compression, filtering, histogram equalization, and noise attacks</li> <li>• Security is not observed</li> </ul> | <ul style="list-style-type: none"> <li>• Methods [14, 15] have been developed</li> <li>• Enhances the security of the embedded image</li> <li>• Improves the payload capacity</li> <li>• Less sensitive to the threshold</li> <li>• Robust against edge changes and steganalysis attacks</li> <li>• Requires more time to execute</li> </ul>                                                                                                                  |
| IoT and Cloud    | <ul style="list-style-type: none"> <li>• Methods [16, 17] have been developed</li> <li>• Minimizes data loss in healthcare system</li> <li>• Robust against noise, filtering, rotation, and compression attacks</li> <li>• Not much robust against cropping, rotation, and hybrid attacks</li> <li>• Real-time data are not considered</li> </ul>                                                                                                               | <ul style="list-style-type: none"> <li>• Methods [18, 19] have been developed</li> <li>• Provides improved security, better quality of the stego image, and guarantees high embedding capacity</li> <li>• Minimizes distortion of the host image</li> <li>• No information loss is occurred</li> <li>• Noise can be added</li> <li>• Cannot work for real-time system</li> <li>• Poor performance for noisy image</li> </ul>                                  |
| Blockchain       | <ul style="list-style-type: none"> <li>• Methods [20, 21] have been developed</li> <li>• Solves false positive detection (FPD) problem</li> <li>• Ensures more security</li> <li>• Less robust against motion blur, sharpening, and histogram equalization attacks</li> </ul>                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>• Methods [22, 23] have been developed</li> <li>• Works with large amount of data</li> <li>• Ensures high payload capacity and high visual quality of the stego image</li> <li>• Improves the confidentiality and integrity of the host image</li> <li>• Detects image tampering easily</li> <li>• Robust against chosen-plaintext attack</li> <li>• Cannot tested with real data</li> <li>• Complex system</li> </ul> |



Imperceptibility is the perceptual similarity which indicates that the embedded watermark should be invisible by the human vision system. Robustness is the strength to detect the watermark from the “processed” image [24]. Security makes the embedded watermark unrecoverable even after encoding or decoding is known [25]. Secret key is used for strengthening the watermark’s security level which must be chosen properly. Embedding capacity describes the amount of data that are inserted into the cover image safely without it being discovered. The basic design requirements for image watermarking is shown in Fig. 3. But, there exists a limited and conflicting relation among imperceptibility, robustness, and capacity. If capacity and robustness are increased, then the imperceptibility will be decreased. In addition, the robustness will be decreased when the capacity is increased. Being maintaining a proper balance among these requirements is really a challenging task. The trade-off among robustness, imperceptibility, and embedding capacity is shown in Fig. 4.

For steganography, the basic design requirements are imperceptibility, robustness, payload capacity, and types of keys. Imperceptibility is one of the significant

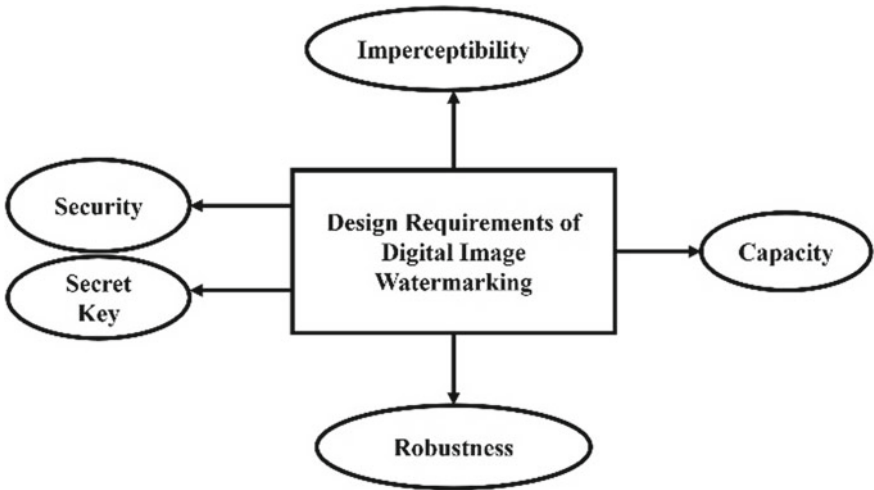
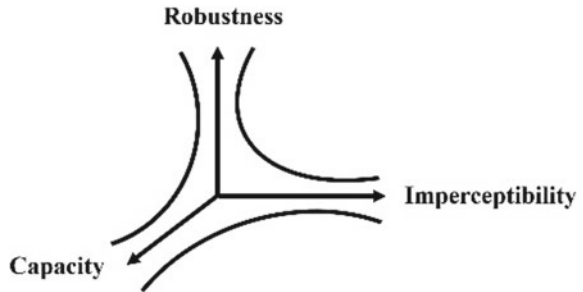


**Fig. 3** Design requirements for image watermarking

**Fig. 4** Trade-off among robustness, imperceptibility, and capacity



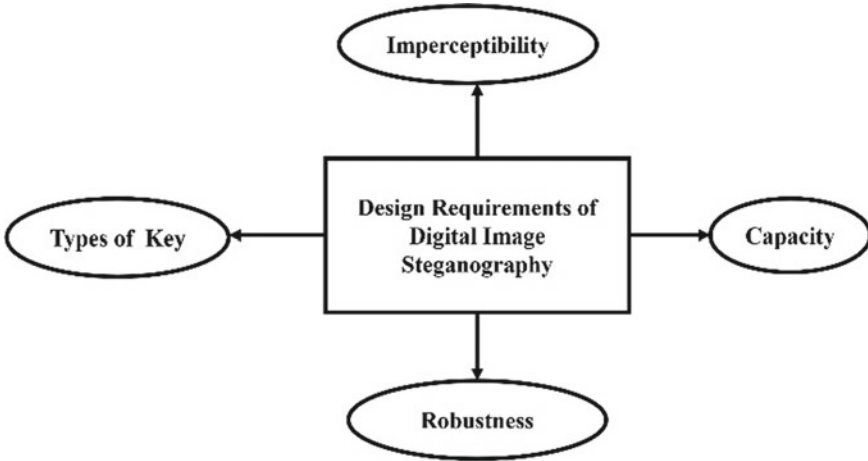


**Fig. 5** Design requirements for image steganography

requirements which identifies the strength of the system. The steganography system is imperceptible only if there exists no visual difference between the host and generated stego images. Robustness is the ability to detect the embedded image even after some intentional and unintentional attacks. Besides, payload capacity defines the total amount of embedded secret data into the cover image. But, it is really a big challenge to maintain a proper balance among these basic requirements. For designing an effective steganography system, two types of keys are used. Steganographic key hides the message by using algorithm where cryptographic key encrypts the message by using cryptographic techniques [26]. These requirements for image steganography are shown in Fig. 5.

#### 4 State-of-The-Art Methods for Image Watermarking and Steganography

There exists a lot of research works based on watermarking in which some of them can be used as the state-of-the-art methods as they meet some of the basic design requirements simultaneously. The performance of the system is evaluated by peak-signal-to-noise-ratio (PSNR), normalized correlation (NC), normalized-cross-correlation (NCC), bit error rate (BER), and bit per pixel (bpp). For example, in 2016 [27], a multiple watermarking method based on discrete wavelet transform (DWT), discrete cosine transform (DCT), and singular value decomposition (SVD) is proposed for healthcare system. The noise of the watermarked image is reduced by applying back propagation neural network (BPNN) to the extracted watermark image. Arnold map encrypts the watermark image before embedding it into the host image. The method is robust against JPEG compression, salt-and-pepper-noise, speckle noise, filtering

**Table 3** State-of-the-art methods for image watermarking

| Used Techniques             | Factors                                              | Results                                                       | Applications                                       |
|-----------------------------|------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------|
| DWT, DCT, and SVD [27]      | Imperceptibility, robustness, security, and capacity | PSNR > 34.35 dB for different gain factors, Average NC = 0.82 | Healthcare                                         |
| Lifting and companding [28] | Imperceptibility, security, and capacity             | NCC > 0.80, BER = 0, Execution time, $T = 1.3543$ s           | DICOM images                                       |
| WHT [29]                    | Imperceptibility, robustness, and security           | PSNR > 49 dB, NC = 1, Capacity, $C = 0.25$ bpp                | Image processing, speech processing, and filtering |

attack, rotation, resizing, cropping, and hybrid attacks. Experimental results demonstrate that the method ensures the superior performance based on imperceptibility, robustness, security, and embedding capacity, simultaneously than existing recent methods. In 2019 [28], another reversible watermarking method is proposed for Digital Imaging and Communications In Medicine (DICOM) images that ensures better imperceptibility, security, and high payload capacity to the host image, simultaneously. The watermark is embedded into the lifting based DWT domain to the host image. The hidden capacity is increased by companding technique. The method is robust against collusion attack and requires less time to execute than single DWT-based methods. In 2020 [29], another blind color image watermarking method is proposed based on Walsh Hadamard Transform (WHT). WHT is applied to each block of the host image. The watermark is embedded into the WHT coefficients. The security is ensured by Arnold map and MD 5 (Message Digest 5). The method ensures improved robustness, imperceptibility, and high security to the watermark image. Table 3 highlights these state-of-the-art methods for image watermarking along with their results.

For steganography system, some methods meet the basic design requirements, simultaneously. Hence, in the following table, we have identified the state-of-the-art methods for steganography. In 2021 [11], a hybrid method based on chaotic map and particle swarm optimization (PSO) has been developed. This hybrid algorithm finds the best pixel location to embed the secret image into the host image. Thus, the method ensures better imperceptibility of the stego image and guarantees high payload capacity. Here, chaotic map ensures the security of the secret image. In 2021 [15], a convolutional neural network (CNN)-based steganography method is developed with deep supervision based edge detector. Here, more bits are embedded into edge pixels while fewer bits are embedded into the non-edge pixels. The stego image is less distorted by ensuring high payload capacity. In 2021 [18], an IoT- and steganography-based method is proposed to enhance the security of the embedded data. Here, binary bit-plane decomposition (BBPD) is used for encrypting the embedded image. Salp swarm optimization algorithm (SSOA) is used for increasing the payload capacity. Security of the medical image is ensured by IoT protocol. The method ensures better imperceptibility of the stego image by using

**Table 4** State-of-the-art methods for image steganography

| Used techniques          | Factors                                  | Results                                                  | Applications           |
|--------------------------|------------------------------------------|----------------------------------------------------------|------------------------|
| Chaotic map and PSO [11] | Imperceptibility, security, and capacity | PSNR = 63.029 dB, Capacity = 204,800 bits                | –                      |
| CNN [15]                 | Imperceptibility, security, and capacity | PSNR > 40 dB, Average embedding capacity, $C = 2.51$ bpp | –                      |
| BBPD and SSOA [18]       | Imperceptibility, security, and capacity | PSNR > 52 dB, Embedding capacity, $C = 2.5$ bpp          | Medical image security |

fuzzy neural network. Table 4 highlights these state-of-the-art methods for image steganography along with their results.

5 Applications of Image Watermarking and Steganography

Image watermarking can be used for access control, copy control, broadcast monitoring, fingerprinting, multimedia streaming, defense security, medical image security, e-governance, health care, halftoning, copyright protection, image authentication, and information hiding. Besides, image steganography can be used for access control, online transaction, smart cards, medical imaging, health care, printing, and in military applications. The associated applications of image watermarking and steganography are shown in Figs. 6 and 7, respectively.

6 Challenges and Suggestions in Developing Image Watermarking and Steganography Techniques

There remain some challenges in developing existing image watermarking and steganography techniques that should be overcome. In this section, we have given some suggestions to overcome potential issues of existing techniques.

- Image watermarking hides less information of an image while steganography hides large blocks of an image. Hence, algorithm should be efficient enough to hide the embedded information.
- Each part of an image has different levels of energy. Therefore, the host image should be divided into blocks to select the best embedding position.
- Image watermarking needs to be more robust to remove or replace hidden message, while in steganography high security is not required for this.

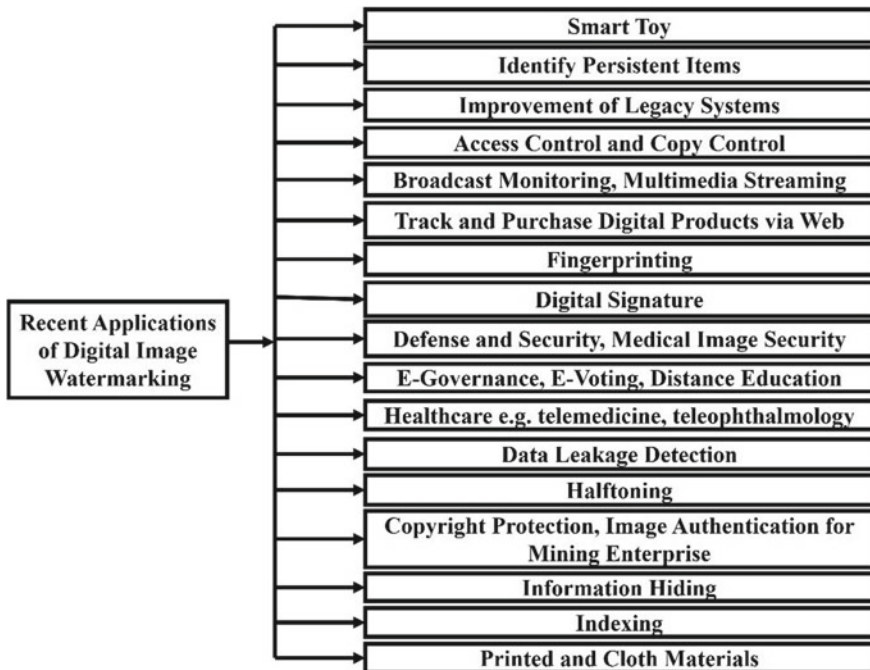


**Fig. 6** Applications of image watermarking [4]

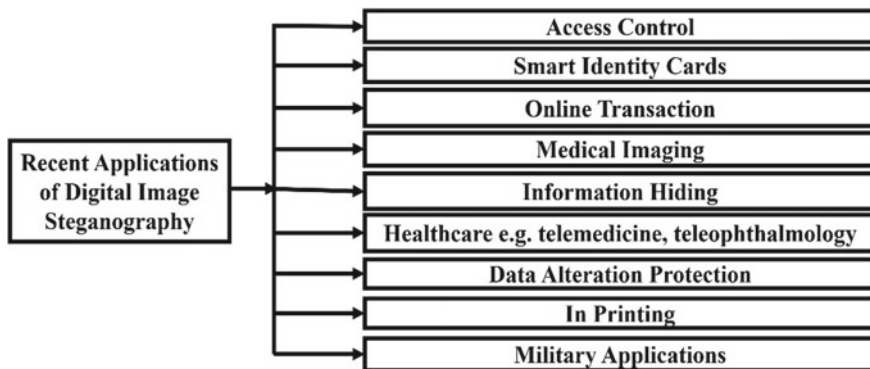


**Fig. 7** Applications of image steganography

- For invisible image watermarking, high robustness and good imperceptibility are required which is a challenge. On the other hand, for steganography, good imperceptibility and high payload capacity are very demanding issues. For this, prediction-based reversible techniques should be developed [4].

- It is a big challenge to maintain a proper balance among imperceptibility, robustness, security, and capacity simultaneously for developing an effective image watermarking system. For steganography, the trade-off among imperceptibility, security, and embedding capacity is a major challenging issue. For minimizing these issues, the statistical parameters of images have to be modelled, effectively.
- In watermarking, best results are obtained after combining it with cryptography. Object-oriented steganography techniques should be developed [30].
- Embedding efficiency should be increased and embedding distortion should be minimized for designing an effective image watermarking or steganography system [31].
- Image coefficients should be modified to overcome false-positive error [32].
- The embedded image can be eliminated using data compression techniques. For minimizing this, optimization techniques would be promising solution
- Algorithms for designing image watermarking and steganography techniques should be designed more carefully by considering all intentional and unintentional attacks.
- Some issues like authenticity, confidentiality, and integrity must be addressed for designing an effective system.
- High-frequency components provide better imperceptibility with less robustness while low frequency components demand better robustness. Also, diagonal sub-band coefficients ensure better robustness than horizontal and vertical coefficients [32]. Hence, existing transform domain techniques need significant modification.
- The algorithm should be incorporated with cryptographic techniques to maintain a proper balance between embedding capacity and security. But, this incorporation doesn't ensure better trade-off better robustness and total complexity.
- Image security techniques need advancement for minimizing delay between embedding and extraction. For security issue, IoT, biometric technique, or cloud computing environment would be an efficient solution.
- Internet of Things provides enhanced security of the system by using sensors. If we transmit the generated watermarked or stego image through IoT network, it will be more secured. IoT ensures authentication so that the network cannot be accessed by unauthorized user. IoT protects malicious attacks as the watermarked or stego image is only be accessible by valid user.
- Hardware-based techniques should be developed.
- Existing methods use different parameters for evaluating performance which makes the comparison of different methods more complex. Hence, some common parameters should be used as benchmark.
- Lack of availability of benchmark dataset.
- Real-time watermarking or steganography is still questionable [3].
- More advancements should be done to develop an efficient image watermarking or steganography system that works with quantum images [3].

## 7 Conclusions and Future Directions

Digital image watermarking and steganography are significant research areas as ownership identification and information sharing in a secret way are very challenging tasks. In this study, we have pointed out the fundamental comparison issues between image watermarking and steganography techniques. In watermarking, the host image is more important than the embedded information while the hidden message is more important than the cover image in steganography. This criterion makes a big comparison issue of these two information hiding techniques. Hence, the trade-off between robustness and imperceptibility should be maintained for watermarking while a proper balance between security and embedding capacity should be preserved in steganography. Future researchers should transmit the generated embedded image into the IoT network to make the communication channel more secured. Also, the researchers should design advanced machine learning and optimization techniques for increasing the imperceptibility and robustness of the system.

**Acknowledgements** The authors are thankful to the Information and Communication Technology Division of the Government of the People's Republic of Bangladesh for a Ph.D. fellowship to Mahbuba Begum.

**Author Contributions** All authors have contributed equally to performing this research.

**Funding** No funding is received for this research.

**Conflicts of Interest** The authors declare that they do not have any commercial or associative conflict of interest in connection with the work submitted.

## References

1. Begum M, Uddin MS (2020) Digital image watermarking techniques: a review. *Information* 11(2):110
2. Begum M, Uddin MS (2020) Analysis of digital image watermarking techniques through hybrid methods. *Adv Multimedia* 2020:7912690
3. Subramanian N, Elharrouss O, Al-Maadeed S, Bouridane A (2021) Image steganography: a review of the recent advances. *IEEE Access* 9:23409–23423
4. Begum M, Uddin MS (2021) Towards the development of an effective image watermarking system. *Secur Priv* e196
5. Machine learning. Available online: [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning). Last accessed 05 Apr 2021
6. Internet of things. [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things). Last accessed 06 Apr 2021
7. Takore TT, Kumar PR, Devi GL (2018) A new robust and imperceptible image watermarking scheme based on hybrid transform and PSO. *Int J Intell Syst Appl (IJISA)* 10(11):50–63
8. Zhang Y, Li Y, Sun Y (2019) Digital watermarking based on joint DWT–DCT and OMP reconstruction. *Circuits Syst Sig Process* 38:5135–5148
9. Cheema AM, Adnan SM, Mehmood Z (2020) A novel optimized semi-blind scheme for color image watermarking. *IEEE Access* 8:169525–169547

10. Dhanasekaran K, Anandan P, Kumaratharan N (2020) A robust image steganography using teaching learning based optimization based edge detection model for smart cities. *Comput Intell* 1–15
11. Jaradat A, Aqieddin E, Mowafi M (2021) A high-capacity image steganography method using chaotic particle swarm optimization. *Secur Commun Netw* 6679284
12. Lee J-E, Seo Y-H, Kim D-W (2020) Convolutional neural network based digital image watermarking adaptive to the resolution of image and watermark. *Appl Sci* 10(6854):1–20
13. Kazemi M, Pourmina MA, Mazinan AH (2020) Analysis of watermarking framework for color image through a neural network-based approach. *Complex Intell Syst* 6:213–220
14. Shang Y, Jiang S, Ye D, Huang J (2020) Enhancing the security of deep learning steganography via adversarial examples. *Mathematics* 8:1446
15. Ray B, Mukhopadhyay S, Hossain S (2021) Image steganography using deep learning based edge detection. *Multimed Tools Appl* 80:33475–33503
16. Al-Shayea TK, Batalla JM, Mavromoustakis CX, Mastorakis G (2019) Embedded dynamic modification for efficient watermarking using different medical inputs in IoT. In: 2019 IEEE 24th international workshop on computer aided modeling and design of communication links and networks (CAMAD). Limassol, Cyprus, Sept 2019, pp 1–6
17. Al-Shayea TK, Mavromoustakis CX, Batalla JM, Mastorakis G, Mukherjee M, Chatzimisios P (2019) Efficiency aware watermarking using different wavelet families for the internet of things. In: ICC 2019—2019 IEEE international conference on communications (ICC), Shanghai, China, May 2019, pp 1–6
18. Dhawan S, Chakraborty C, Frnda J, Gupta R, Rana AK, Pani SK (2021) SSII: Secured and high-quality steganography using intelligent hybrid optimization algorithms for IoT. *IEEE Access* 9:87563–87578
19. Alarood A, Ababneh N, Al-Khasawneh M (2021) IoTSteg: ensuring privacy and authenticity in internet of things networks using weighted pixels classification based image steganography. *Cluster Comput*
20. Aparna P, Kishore PVV (2020) A blind medical image watermarking for secure E-healthcare application using crypto-watermarking system. *J Intell Syst* 29(1):1558–1575
21. Nazir H, Bajwa IS, Samiullah M, Anwar W, Moosa M (2021) Robust secure color image watermarking using 4D hyperchaotic system, DWT, HbD, and SVD based on improved FOA algorithm. *Secur Commun Netw* 2021(6617944):1–17
22. Mohsin AH, Zaidan AA, Zaidan BB (2021) PSO–blockchain-based image steganography: towards a new method to secure updating and sharing COVID-19 data in decentralised hospitals intelligence architecture. *Multimed Tools Appl* 80:14137–14161
23. Horng J-H, Chang C-C, Li G-L, Lee W-K, Hwang SO (2021) Blockchain-based reversible data hiding for securing medical images. *J Healthc Eng* 2021:9943402
24. Piper A, Safavi-Naini R (2009) How to compare image watermarking algorithms. *Trans Data Hiding Multimedia Secur* 5510:1–28
25. Ghouti L (2017) A perceptually-adaptive high-capacity color image watermarking system. *KSII Trans Int Inf Syst* 11(1):570–595
26. Gribermans D, Jeršavs A, Rusakovs P (2016) Development of requirements specification for steganographic systems. *Appl Comput Syst* 20:40–48
27. Zear A, Singh AK, Kumar P (2016) A proposed secure multiple watermarking technique based on DWT, DCT, and SVD for application in medicine. *Multimed Tools Appl* 77:4863–4882
28. Phadikar A, Jana P, Mandal H (2019) Reversible data hiding for DICOM image using lifting and companding. *Cryptography* 3:21
29. Prabha K, Sam IS (2020) An effective robust and imperceptible blind color image watermarking using WHT. *J King Saud Univ Comput Inf Sci*
30. Cheddad A, Condell J, Curran K, Mc Kevitt P (2008) Biometric inspired digital image steganography. In: Proceedings of the 15th annual IEEE international conference and workshops on the engineering of computer-based systems (ECBS'08). Belfast, pp 159–168



31. Roy R (2013) Evaluating image steganography techniques: future research challenges. In: International conference on computing, management and communications. Academia, Vietnam
32. Ray A, Roy S (2020) Recent trends in image watermarking techniques for copyright protection: a survey. *Int J Multimed Info Retr* 9:249–270

# Chapter 2

## Vehicle Detection Using Deep Learning Method and Adaptive and Dynamic Automated Traffic System via IoT Using Surveillance Camera



Nafiz Arman , Syed Rakib Hasan , and Md. Ruhul Abedin

**Abstract** Internet of things (IoT) means the interconnection of various devices where each electrical and mechanical device, be it a fan or a light bulb or even a car, are connected to each other in order to present a fully automated system framework. IoT envisions a future where every electrical peripheral will be connected with each other in some form or another. Technological advancements have driven the developments in the different modes of transport. The number of vehicles in cities keeps increasing day by day, and it is ever present, even in developed countries. Due to the increasing number of vehicles and unplanned road framework, traffic jams at important junctions are reaching critical levels. Therefore, a truly efficient automated system is necessary for improving the scenario. This work proposes an automated system where each vehicle is first detected, and then, the number of vehicles is utilized to control traffic lanes automatically. The whole process is fully automated and requires no human input after activation. Vehicle detection is done via a deep learning algorithm, which improves the detection success percentage over time. Moreover, the system focuses on an analysis of leftover vehicles in a lane to prioritize it and a lot more time for vehicles to pass through.

**Keywords** Deep learning · Internet of things · Machine learning

---

N. Arman (✉)  
University of Calgary (UCalgary), Calgary, Canada  
e-mail: [nafizarman@gmail.com](mailto:nafizarman@gmail.com)

S. R. Hasan  
Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh

Md. R. Abedin  
Bangladesh Army International University of Science and Technology (BAIUST), Cumilla, Bangladesh

## 1 Introduction

Nowadays, the way technology advancements are improving following these steps devices, and machines are increasing too. As a result, controlling these new devices and machines is now a great challenging issue for us in daily life. The Internet of things (IoT) is a modern technological paradigm that envisions a worldwide network of interconnected equipment and gadgets [1]. So, it can be said is a new technology that can bring all the machines and devices into a common platform for controlling, monitoring, and surveillance purpose. Higher data rates, huge bandwidth, high capacity, low latency, and high throughput are required for these applications [2]. Augmented reality, high-resolution video streaming, self-driving vehicles, smart environments, e-health care, and other IoT-centric ideas are now widespread [2].

It is almost impossible to keep eye on a large number of machines and devices. Several types of input, output, and data analysis are needed for controlling, monitoring, and maintaining the complex system. To make this easier, deep learning is the best solution. It also does not require any previous data processing, and it can also extract features automatically [3]. The assessment and learning of enormous volumes of unsupervised data are a core feature of deep learning, making it a powerful tool for analyzing big data when raw data is essentially unlabeled and cannot be categorized [4].

The increasing number of vehicles in this era is becoming increasingly difficult to control and manage. The problem becomes much more critical in bigger cities such as Dhaka and Chittagong. Development of new roads and infrastructures such as flyovers and bridges causes further traffic jam until their completion. So, it becomes a necessity to improve and remake the current framework. Thus, it comes the solution in the form of a dynamic and adaptive automated traffic management system. The system utilizes IoT-based surveillance cameras for vehicle detection. Surveillance cameras have become widely available nowadays due high availability and inexpensiveness. In a smart city, every single electronic device will be connected together and monitored constantly. So, viability and implementation of surveillance cameras become even more logical. Due to this reason, implementation of sensors is not necessary.

## 2 Related Works

Research works carried out by various researchers have been analyzed and scrutinized in order to find out the latest trends of the research in the concerned area. This provided an idea about the best possible methods for the completion of the proposed traffic system framework.

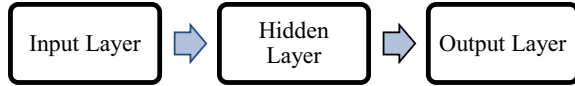
The proposed system makes use of IoT, surveillance cameras, and image processing for detection of vehicles. It requires no additional sensor data for utilization. Till date, most of the research works regarding the solutions of traffic congestions have incorporated sensors of some form or other in order to properly detect the number of vehicles and their density. Primarily, ultrasonic sensors have been utilized for this purpose [5]. Even when using interconnectivity and IoT protocol, the usage of ultrasonic sensors for car detection was prevalent [6]. Alternatively, in an established IoT environment, the vehicles would be connected directly to the network themselves. Many wireless vehicular connectivity protocols are being analyzed and developed by researchers [7].

Traffic surveillance for future planning such as pollution control, traffic signal control, infrastructure and road planning, and accidental blockage avoidance was done via surveillance cameras using image processing techniques such as Canny edge detection [7], spatial temporal analysis with hidden Markov modeling [8], and thresholding and background subtraction [9]. To that effect, several types of cameras were utilized. Cameras such as standard 10 cameras, V8 camcorders, pan-tilt-zoom cameras, and CCTV cameras were used for capturing video footage of vehicles in a lane [10].

Several types of image processing, pattern recognition, and object detection techniques have been developed over the years. For image recognition, the Viola-Jones algorithm [11] and oriented gradient [11] methods were popular in use throughout the years. However, since 2012, the deep learning algorithms have become more and more popular. Deep learning algorithms, although hardware dependent, are blazing fast. The most important attribute of deep learning is the fact that the detection accuracy increases over time. State-of-the-art algorithms such as AlexNet, DenseNet, SqueezeNet, and you only look once (YOLO) can accurately detect objects and present results. For vehicle detection, convolutional neural networks (CNN), regions with convolutional neural network features (R-CNN), and faster R-CNN [12] can be utilized [13]. R-CNN and faster R-CNN show very fast execution times along with highly accurate results [14]. However, R-CNN detection methods require creation of a neural network with layers upon layers of feature extractors, which can become time consuming.

Alternatively, a GPU accelerated aggregate control feature method can be implemented, which searches the entire image using a sliding window approach. Assignment of each window to a separate CUDA thread severely reduces code execution time, which is very important for real-time analysis [15]. Implementation of machine learning algorithms and Internet of things is crucial for development of an IoT connected traffic control system [16].

**Fig. 1** Deep learning layer framework



### 3 Algorithm

#### 3.1 Object Detector Training

Initial project prototype was built via a modified aggregate channel feature (ACF) object detector. Final proposed system utilizes a deep learning algorithm for vehicle detection via surveillance cameras.

The ACF detector is a fast and effective sliding window detector (30 fps on a single core) with an ~1000-fold decrease in false positives (at the same detection rate). Aggregate channel feature object detection is a part of deep learning methodology. Deep learning methods require input data. The provided input data can be designated as the input layer. The input data contains information about the regions of interest. The input is then processed by the detector. This stage is called the hidden layer. Here, the detector trains itself by the provided information embedded in the input data. The output layer provides detected images.

Images are marked with regions of interest, also known as labeling. Labeled images create a ground truth sample data. The data contains information about the regions of interest, their composition, coordinates, and the image source. The object detector learns detection from this ground truth data by analyzing and processing the labeled regions of interest. The detector is then trained on images marked with object of interest, in this case vehicles. It will then automatically use positive instances of image objects and at the same time collect negative instances of objects as well (Fig. 1).

#### 3.2 System Architecture

The proposed system contains a surveillance camera, a central computer with appropriate hardware, and traffic lights, all interconnected with each other. Surveillance cameras capture continuous video footage. Interconnectivity allows data transfer to the central server, which in turn sends the data to the main computer. The computer provides fast decision-making and sends the information to the central server, which in turn sends it to the traffic lights. The traffic lights are thus directly controlled via the computer. After serving a full cycle, they await further instructions from the processor. The whole process is fully automated.

Generic surveillance cameras generally have an angle of view value of 80°. Cameras having higher angle of view are very expensive. Calculations provide a value of 7.5 m as optimal height for viewing a distance of 42.53 m. The entire

arrangement allows for about 40–50 cars to be present in an image, which is ample information for the system to recognize which lanes to prioritize over the others.

The proposed system takes in images from the preinstalled cameras at a regular interval. This is done in order to take time for processing and analyzing images received. The central system connected directly to the surveillance cameras, thereby possessing interconnectivity and establishing an IoT environment. The system then enacts suitable actions based on the flow of the traffic, number of vehicles present, and leftover vehicles that were unable to pass during the allocated time.

Allocated time is based upon the total number of vehicles. A preset amount of time is allocated based upon this data. From the allocated total time, a set amount of time is allotted to each lane depending on the number of vehicles present in each individual lane.

As it is difficult to distinguish each individual vehicle present in the received images, images are taken every 30 s or so. This enables the system to obtain an idea of vehicle congestion in lanes. The system prioritizes these lanes and allocates more time for the vehicles to pass through by deducting the same amount of time from lower priority lanes. The procedure enables the system to keep the total allocated time constant, as no extra time is being allocated.

## **4 Dynamic and Adaptive Vehicle Detection Using Deep Learning**

### ***4.1 Initializing Deep Learning***

Deep learning processes are sophisticated enough to directly extract and process features without any post processing, background subtraction, or filtering. This allows deep learning networks to perform state-of-the-art accuracy and speed during classification tasks such as text-to-speech writing, object detection, image segmentation, and voice recognition.

Vehicles are detected via a deep learning aggregate channel feature (ACF) object detector. For initialization of the detector, it needs a number of sample images which have been marked with objects of interest, which are vehicles in this case. Images are marked via a ground truth creator. Still, images have been chosen due to ease of burden and efficiency of a processor. Processing a video sequence can become very troublesome, especially if the image feed is continuous, such as in a surveillance camera. This occurs due to the fact that even a low configuration 15 fps (frames per second) camera generates 15 fps. That is essentially 15 different images as far as the processor is concerned. For a duration of 30 or seconds, about 450 individual frames would be generated which are quite literally different from each other. Higher configuration cameras would generate even more frames. Processing such a huge number of individual frames will take precious processor cycles and valuable time. Due to this reason, the proposed system architecture has been designed in such a

way that it only processes one or two frames of a constant video feed every 30 s or so. Such a designed framework allows the system to generate decisions and enforce them within a short period of time.

Each individual image is labeled as a “ground truth”. After sufficient amount of labeled data is generated, they are exported to a ground truth file. The ground truth file is crucial for the development of a deep learning detector. The file contains information on the image source, regions of interest, and their coordinates in the images.

For obtaining a higher percentage of successful detection, a high number of sample images are a must. Sophisticated detectors are trained with about 1000 images or so. Initial system framework was built using 100 images. While the number of sample images was indeed low, the focus was given to quality over quantity—to be accurate and precise even in the event of having been provided with low sample images.

## ***4.2 Detector Training and Evaluation***

After creation of ground truth, a detector is created and trained. To do so, the ground truth file is initialized. After that, the specific label definitions are selected, as there may be several instances of image labels in the ground truth matrix. The detector is then run-on sample image sets. These image sets are kept in a folder separate from the rest of the image sets and labeled training data. The images need to be different from the labeled images as otherwise, it will provide an accuracy of 100%. Detected objects are marked via a bounding box. Increasing number of cycles allows for greater chance and predictions on collection of positive and negative instances of sample data. After successful training session, the detector is saved and tested on various images (Figs. 2 and 3).

## ***4.3 Allocation of Time***

Allocation of time for the vehicles is done via detection and calculating the number of cars available in a lane. The program calculates the total number of vehicles at an instance when vehicles are in standstill, waiting for a green light for any particular lane, and using the algorithm, it allocates four predefined time slots. These slots are provided to all the lanes as time duration for the green light. Due to practicality, a minimum of 15 s and a maximum of 240 s of time allotment have been imposed on every situation.

A prioritization factor,  $\beta$ , is incorporated into the calculations when a lane has been selected as a priority lane. The prioritization factor increases the allocated time for that lane. Due to practical reasons, the increment is never higher than 20%. In order to keep the system balanced and effective, the extra time allocated to congested lane is deducted from the lower priority lanes.



**Fig. 2** Testing object detection on random images



**Fig. 3** Detection results on live camera in Dhaka city

Object detection becomes very taxing and complicated in intense situations. Moreover, some vehicles are unable to be detected via conventional methods when they are blocked by other vehicles. To save processing power and improve detection, only the windshield was chosen as region of interest during creation of the ground truth objects. Due to windshields being ever present in all vehicles, this allows for higher vehicle detection across a wide area.



Resolution has a significant effect on object detection in images. Higher resolution photos generally tend to provide more accurate detections. Higher resolution images contain more information in a particular section than a lower resolution image in the same section. More information allows the detector to acquire more positive and negative sample instances, thus increasing its ability to detect desirable portions of an image more accurately (Figs. 4 and 5).



**Fig. 4** Low vehicle detection on random lower resolution image



**Fig. 5** Vehicle detection in dense traffic scenario on random image

**Table 1** Data collection results

| Location name      | Approximate allocated time (s) | Total vehicles | Vehicles passed | Vehicles remaining | Approximate effective clearing time (s) |
|--------------------|--------------------------------|----------------|-----------------|--------------------|-----------------------------------------|
| Science laboratory | 90                             | 50             | 45              | 5                  | 4                                       |
| Bijoy Sharoni      | 120                            | 60             | 40              | 20                 | 3                                       |
| Banglamotor        | 100                            | 45             | 40              | 5                  | 3                                       |

5 Simulation

5.1 Data Collection

Practical data was collected from various locations in Dhaka city, primarily from intersection locations such as science laboratory intersection, Bijoy Sharoni, and Banglamotor. The science laboratory intersection was chosen as the ideal location due to ease of data collection. 30 min of data collection presented the following results.

As evident from Table 1, each vehicle receives only an average of 3–4 s for leaving the lane. This is often insufficient in practical scenarios. The present traffic system fails to incorporate factors such as this for clearing a lane. Due to this reason, vehicles remain stuck in traffic jams for a very long time.

5.2 Simulation Environment

The simulation environment was created based upon the practical data of the science laboratory intersection. Sample image is taken every 30 s from the video footage.

Tables 2 and 3 show that compared to the practical data, a significant improvement of the effective clearing time for each vehicle can be observed in the aforementioned table. Having higher clearing time means more vehicles can successfully pass through while also reducing the possibility of a vehicle being stuck in a particular lane for longer than usual time.

The system captures an extra footage or so at every third cycle to determine congestion on traffic lanes. Lanes having high traffic density at all times are then designated as priority lanes. A congestion clearing factor,  $\beta$ , is then added to the allocated time periods in order to provide more time for vehicles to successfully pass through. In order to keep the allocated time within limits, the extra allocated time is then deducted from the lower priority lanes. This keeps the entire system in balanced condition, while also allowing the lanes facing serious traffic congestion to be cleared, evident by the results of Table 4.

**Table 2** Simulation results

| Serial number | Number of detected cars in lane A | Number of detected cars in lane B | Allocated time (s) |
|---------------|-----------------------------------|-----------------------------------|--------------------|
| 1             | 14                                | 27                                | 180                |
| 2             | 14                                | 23                                | 110                |
| 3             | 13                                | 33                                | 120                |
| 4             | 10                                | 22                                | 120                |
| 5             | 10                                | 30                                | 90                 |
| 6             | 10                                | 31                                | 90                 |
| 7             | 11                                | 32                                | 120                |
| 8             | 13                                | 28                                | 180                |
| 9             | 03                                | 50                                | 230                |

**Table 3** System determined effective clearing time

| Serial number | Allocated time to lane A (s) | Allocated time to lane B (s) | Effective clearing time (s) |
|---------------|------------------------------|------------------------------|-----------------------------|
| 1             | 71.70                        | 138.30                       | 5.12                        |
| 2             | 56.75                        | 93.24                        | 4.05                        |
| 3             | 42.39                        | 107.60                       | 3.30                        |
| 4             | 46.87                        | 103.13                       | 4.68                        |
| 5             | 37.50                        | 112.5                        | 3.75                        |
| 6             | 51.21                        | 158.78                       | 5.12                        |
| 7             | 53.72                        | 156.27                       | 4.88                        |
| 8             | 66.58                        | 143.41                       | 5.12                        |
| 9             | 13.58                        | 226.41                       | 4.52                        |

**Table 4** Congestion clearing algorithm simulation results

| Serial number | Leftover vehicles in lane A | Leftover vehicles in lane B | Allocated time to prioritized lane (s) | Allocated time to lower priority lane (s) | Effective clearing time (s) |
|---------------|-----------------------------|-----------------------------|----------------------------------------|-------------------------------------------|-----------------------------|
| 1             | 13                          | 33                          | 129.12                                 | 20.88                                     | 3.01                        |
| 2             | 10                          | 31                          | 190.5                                  | 19.46                                     | 2.01                        |
| 3             | 03                          | 50                          | 271.7                                  | 31.7                                      | 15.98                       |

### 5.3 Model Effectiveness

The proposed image detection system shows an approximate 91.75% accuracy coupled with a 0.5% loss at detection of images. As evident, lower number of training

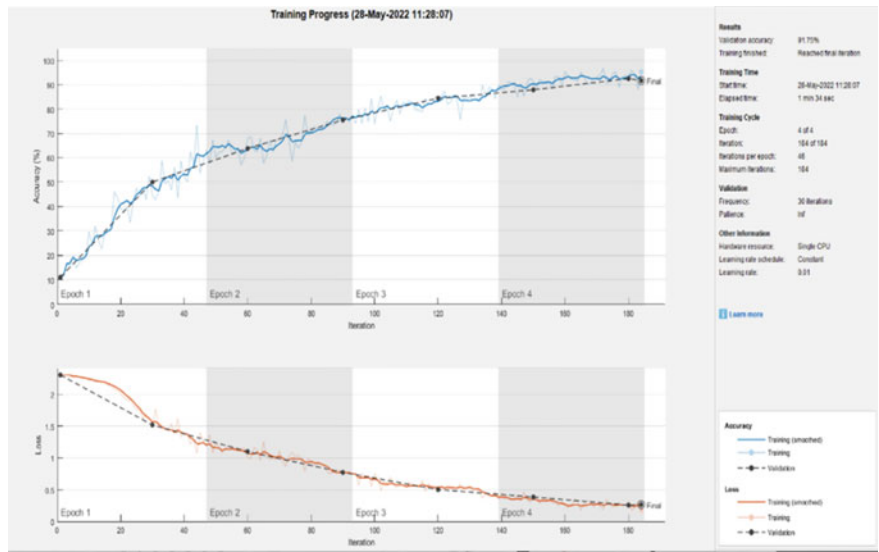


Fig. 6 Accuracy and loss graphs of the proposed detection method

images produce higher initial detection rate and higher loss. The accuracy is predicted to become even higher with the inclusion of appropriate amount of sample data which will, in turn, reduce the losses even further (Fig. 6).

6 Result and Discussion

Implementation of proposed system framework shows an appreciable improvement of the traffic jam condition situation. Current system allocates too little time for vehicles to successfully pass through. Moreover, there are no methods implemented for clearing traffic queuing. Vehicle movement is also hindered severely by the flow of traffic. As a result, some lanes face serious traffic congestion while others are comparatively free of traffic.

Proposed system takes all the variables into account. It detects the number of vehicles and passes judgment based on the number of vehicles present. When a lane remains free for quite some time, it places that lane on a lower priority queue and deducts some allocated time from that lane. However, for ease of traffic movement, no lane will have less than 15 s of allocated time. Deducted time is added to a high priority lane that has been facing more traffic congestion than other lanes. As a result, total allocated time remains the same, and each lane gets green light signal in time.

Each time a lane gets high priority or low priority, and it gets flagged by the system. Continuous flag by the system makes that lane marked as active or passive. The system then sends that information toward the next lane, which utilizes that

information for allocation of vehicle passing. Active lanes will observe a higher than usual allocation time, while the less active lanes will observe lower allocation time.

## 7 Conclusion

Traffic jam and congestion are a very critical prospect of modern civilization. As civilization progresses, the number of vehicles will increase as well and so will the amount of traffic jam. The problem becomes even more critical in densely populated cities. As there are poorly planned road and housing infrastructure in these cities, traffic jam has become almost constant. The traffic controlling methods used at the moment are both inefficient and ineffective. This severely affects the flow of traffic, affecting working hours, and ultimately affecting the economy. Modern approaches must be undertaken in order to tackle this problem effectively in order to attain progress. Evolution is necessary for the cities to become smart cities. A fully adaptive and automated traffic system is the first step toward attaining the mantle of a smart city. It is pretty much a necessity to reach beyond the current scope of the traditional approaches of mitigating the issue of traffic jams. Through the proposed system framework, an effective and efficient means of controlling the traffic has been developed. Implementation of such a framework to the transportation infrastructure will remove the requirement of law enforcement personal at every traffic junction and intersection. The implementation of such an interconnected system allows for a very effective system which can process information of the previous junction and respond accordingly, thus reducing traffic jam further. Thus, it can be ensured that the proposed system will be valuable for the analysis and improvement of road traffic.

## References

1. Lee I, Lee K (2015) The Internet of things (IoT): applications, investments, and challenges for enterprises. *Bus Horiz* 58(4):431–440
2. Shafique K, Khawaja BA, Sabir F, Qazi S, Mustaqim M (2020) Internet of things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios. *IEEE Access* 8:23022–23040
3. Rusk N (2015) Deep learning. *Nat Methods* 13(1):35
4. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. *J Big Data* 2(1):1–21
5. Saifuzzaman M, Moon NN, Nur FN (2018) IoT based street lighting and traffic management system. In: 5th IEEE region 10 humanitarian technology conference 2017, R10-HTC 2017, vol 2018, pp 121–124
6. Nagmode VS, Rajbhoj SM (2018) An intelligent framework for vehicle traffic monitoring system using IoT. In: Proceedings of 2017 international conference on intelligent computing and control. *I2C2 2017*, vol 2018, pp 1–4
7. Lu N, Cheng N, Zhang N, Shen X, Mark JW (2014) Connected vehicles: solutions and challenges. *IEEE Internet Things J* 1(4):289–299

8. Moutakki Z, Ayaou T, Afdel K, Amghar A (2014) Prototype of an embedded system using Stratix III FPGA for vehicle detection and traffic management. In: International conference on multimedia and computer systems—proceedings, pp 141–146
9. Kamijo S, Matsushita Y, Ikeuchi K, Sakauchi M (1999) Traffic monitoring and accident detection at intersections. *IEEE Conf Intell Transp Syst Proc ITSC* 1(2):703–708
10. Kochlan M, Hodon M, Cechovic L, Kapitulik J, Jurecka M (2014) WSN for traffic monitoring using Raspberry Pi board. In: 2014 federated conference on computer science and information systems. *FedCSIS* 2014, vol 2, pp 1023–1026
11. Patel R, Dabhi VK, Prajapat HB (2017) Surveillance and accident detection system. In: International conference on innovations in power and advanced computing technologies [i-PACT2017] A, pp 1–7
12. Lee I et al (2014) Deep residual learning for image recognition. *IEEE Internet Things J* 2(4):770–778
13. Szegedy C et al (2014) Intriguing properties of neural networks. In: 2nd international conference on learning representations. *ICLR* 2014—conference track proceedings, pp 1–10
14. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
15. Mouna B, Mohamed O (2019) A vehicle detection approach using deep learning network. In: 2019 international conference on Internet of things, embedded systems and communications *IINTEC* 2019—proceedings, pp 41–45
16. Van Ranst W, De Smedt F, Goedeme T (2018) GPU accelerated ACF detector. In: *VISIGRAPP* 2018—proceedings of the 13th international joint conference on computer vision, imaging and computer graphics theory and applications (*VISIGRAPP* 2018), vol. 5, pp 242–248

# Chapter 3

## Predicting Average Localization Error of Underwater Wireless Sensors via Decision Tree Regression and Gradient Boosted Regression



Md. Mostafizur Rahman and Sumiya Akter Nisher

**Abstract** Underwater communication in underwater wireless sensor networks is still being researched. For the underwater environment, finding the exact localization of the sensors is a major challenge. It is critical to have accurate sensor location data in order to undertake relevant underwater research. Some algorithms have been proposed that find out the localization of sensors. Due to some factors of underwater wireless sensor networks, the localization error of sensors has occurred. In this paper, the comparative results of predicting average localization error (ALE) have been presented, and using minimal input variable, it has been tried out by applying the decision tree regression and the gradient boosted regression, and the optimal result has been found. Along with these algorithms' simulations, hyperparameter tuning and variables selection have been tuned. Among two algorithm approaches, gradient boosted regression outperformed the previous algorithm approaches, with a RMSE of 0.1379 m.

**Keywords** Average localization error · Decision tree regression · Gradient boosted regression · Wireless sensor networks

### 1 Introduction

More than 80% of the ocean has yet to be discovered. We deploy a set of sensors in underwater to investigate uncharted territory. We need to know the exact location of this set of sensors when we acquire data from sensors to analyze it. It is easier to compute the location in the terrestrial area. However, it is difficult to calculate the exact location of sensors in underwater due to specific characteristics of the underwater wireless sensor network. As a result, every localization technique determines

---

Md. M. Rahman (✉) · S. A. Nisher  
Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh  
e-mail: [mostafiz.johnny@gmail.com](mailto:mostafiz.johnny@gmail.com)

S. A. Nisher  
e-mail: [nisher.sumiyaakter@gmail.com](mailto:nisher.sumiyaakter@gmail.com)

the location with some error. “Prediction of average localization error in WSNs” dataset provides the average localization error for different features of the underwater wireless sensor network. Different support vector machine (SVM) algorithms were used to precisely predict average localization error [1]. This prediction, though, can be more precise. It had been feasible to more precisely anticipate the average localization error in this paper. Different strategies have been used to minimize the root mean square error.

## 2 Related Work

In this section, several techniques have been discussed. Through these techniques, attempts have been made to improve localization accuracy. A number of location schemes have been proposed to determine the localization of sensors. The localization schemes can be broadly categorized into two categories. One is range-based, and another is range-free [2]. Range-free schemes do not use any distance or angle. The centroid scheme Du-hop and density aware hop-count localization are used in range-free schemes. In range-based schemes, accurate distance or angle measurement is needed to estimate the location of sensors. There are different types of range-based schemes. In [3], problem of node localization due to a large number of parameters and the non-linear relationship between the measurements and the parameters is estimated, and they proposed a Bayesian algorithm for node localization in underwater wireless sensor network. They referred to the algorithm as an existing importance sampling method that is referred to as an incremental correlation. In [4], they proposed two localization techniques: neural fuzzy interference system (ANFIS) and artificial neural network (ANN). Artificial neural network (ANN) was hybridized with particle swarm optimization (PSO), gravitational search algorithm (GSA), and backtracking search algorithm (BSA). And in indoor and outdoor, the hybrid GSA-ANN performs a mean absolute distance estimator error with 0.02 m and 0.2 m, respectively. Another common technique is the received signal strength indicator (RSS). In [5], they conducted a machine learning technique survey for localization in WSNs using received signal strength indicator. Decision tree, support vector machine (SVM), artificial neural network, and Bayesian node localization were used in this paper. In [6], they discussed the maximum likelihood (ML) estimator for localization of mobile nodes. After that, they optimized the estimator for ranging measurements exploiting the received signal strength. Then, they investigated the performance of the derived estimator in Monte-Carlo simulations, and they compared it with the simple least squares (LSs) method and exploited received signal strength (RSS) fingerprint. Another technique for improving received signal strength-based localization is discussed in [7]. In this paper, they proposed the use of weighted multilateration techniques to acquire robustness with respect to inaccuracy. Techniques are standard hyperbolic and circular positioning 3 algorithms. In [8], they preferred range-based algorithm over range-free algorithm. They proposed Bayesian formulation of the ranging problem alternative to inverting the path-loss



formula and reduced the gap with the more complex range-free method. In [9], they also preferred range-based algorithm over range-free algorithm. They proposed two step algorithms with reduced complexity, where in first phase, they exploited nodes to estimate the unknown RSS and time of arrival (TOA) model parameters, and in second phase, they combined a hybrid TOA/RSS range estimator with an iterative least square procedure to get unknown position. A localization scheme has been established based on RSS to determine the location of an unknown sensor from a set of anchor nodes [10]. Apart from RSS techniques, there are also some other techniques to determine the localization of sensors. A mathematical model has been proposed in [11], where one beacon node and at least three static sensors are needed. One beacon node from six different positions can determine the localization of static sensors by using Cayley Menger determinant, but sensors plane needs to be parallel to the water surface. For non-parallel situation, they are updated their propose model in [12]. In another paper [13], they also again updated the [11] mathematical model to determine the localization of mobile sensors. Further, in [14], a new mathematical model has been developed to determine the location of a single mobile sensor using the sensor's mobility. Another technique named as IF-ensemble has been proposed in [15] for Wi-Fi indoor localization environment by analyzing RSSs. Another technique in [16] has been proposed to node localization and that is kernel extreme learning machine based on hop-count quantization.

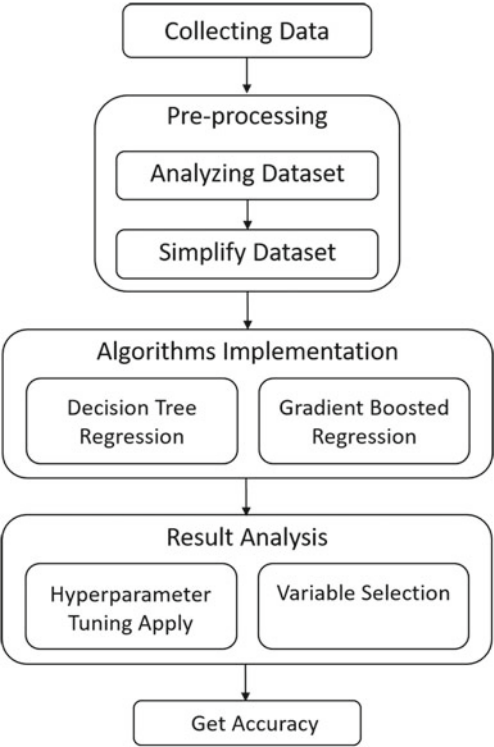
### 3 Methodology

The aim of this research paper is to analyze average localization errors from conducting different machine learning algorithms that can predict precisely and compare the outputs with the previous best output. In our study, we used secondary quantitative data that was acquired by others, and modified cuckoo search simulations were used to generate this dataset. We had not undertaken any studies to change variables because the dataset was already observational. The dataset comprises four valuable attributes that are used to directly achieve the research's main goal. All four features provide generalizable knowledge to validate the research goal just as they do in quantitative research. Figure 1 depicts the workflow of the proposed methodology.

#### 3.1 Dataset Description

The dataset utilized in this paper is “average localization error (ALE) in sensor node localization process in WSNs [17]”. We have used the entire dataset with a total of 107 instances and six attributes, where all attributes represent quantitative data. The dataset contains no missing values as it was already observational data. In order to

**Fig. 1** Workflow of proposed methodology



gain average localization error (ALE), only four variables (anchor ratio, transmission range, node density, and iterations) have been used as input variables, and an average localization error has been used as an output variable. Another attribute (standard deviation value) was ignored in the pre-processing step as our study only focused on generating localization errors.

The number of anchor nodes based on total number of sensors in the network is known as the anchor ratio (AR), which is the first column of this dataset contains numeric values. In the dataset, transmission range also contains numeric values which is measured in meters, which represent the transmission range of a sensor to measure the transmission speed. The node density attribute indicates how densely the activity nodes are connected. The 4th column represents the iteration, which means how many times we took the reading of sensors. The disparity between the actual and projected coordinates of unknown nodes is known as an average localization error.

We have summarized our entire dataset using descriptive statistics. Descriptive statistics provide information on the total count, mean, median, and mode, including standard deviation, variance, and minimum and maximum values.

The dataset is described in Table 1.

Table 1 Description of dataset

|                            | Valid | Mismatched | Missing | Mean | Std. deviation | Quantiles |      |     |     |      |
|----------------------------|-------|------------|---------|------|----------------|-----------|------|-----|-----|------|
|                            |       |            |         |      |                | Min       | 25%  | 50% | 75% | Max  |
| Anchor ratio               | 107   | 0          | 0       | 20.5 | 6.71           | 10        | 15   | 18  | 30  | 30   |
| Transmission range         | 107   | 0          | 0       | 17.9 | 3.09           | 12        | 15   | 17  | 20  | 25   |
| Node density               | 107   | 0          | 0       | 160  | 70.9           | 100       | 100  | 100 | 200 | 300  |
| Iterations                 | 107   | 0          | 0       | 47.9 | 24.6           | 14        | 30   | 40  | 70  | 100  |
| Average localization error | 107   | 0          | 0       | 0.98 | 0.41           | 0.39      | 0.65 | 0.9 | 1.2 | 2.57 |

**Table 2** Pearson's correlation coefficient between input variable and output variable

|                    | Average localization error |
|--------------------|----------------------------|
| Anchor ratio       | −0.074997                  |
| Transmission range | 0.109309                   |
| Node density       | −0.645927                  |
| Iterations         | −0.400394                  |

### 3.2 Variable Importance

In this paper, before applying a machine learning algorithm, the relationship between every input variable and output variable has been evaluated. In this dataset, there are four input variables (anchor ratio, transmission range, node density, and iterations) and one output variable (average localization error). In order to evaluate the relationship between every input variable and output variable, Pearson's correlation coefficient formula has been applied.

So, Pearson's correlation coefficient between two variables,

$$r_{xy} = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

where

$x$  = input feature

$Y$  = output feature.

The range of Pearson's correlation coefficient,

$$\begin{aligned} 0 \leq r_{xy} \leq 1 & \quad (\text{if two features have positive relation}) \\ -1 \leq r_{xy} \leq 0 & \quad (\text{if two features have negative relation}). \end{aligned}$$

Table 2 shows the Pearson's correlation coefficient between every input variable and output variable.

We can see that the highest correlation coefficient exists between node density and average localization error. The average localization error and iterations have the second highest correlation coefficient. The average localization error has the lowest correlation coefficients between anchor ratio and transmission range.

### 3.3 Machine Learning Model

The aim of this research is to predict as close as possible to the average localization error for anchor ratio, transmission range, node density, and iterations. Machine

learning algorithms are divided into three categories: supervised learning, unsupervised learning, and semi-supervised learning. There are two types of supervised machine learning models for predicting something: classification and regression. Regression is used when we want to predict a continuous dependent variable from several dependent variables. The “average localization error” variable is a continuous dependent variable, much like in our dataset. Two regression models have been used to forecast the average localization error.

1. Decision tree regression
2. Gradient boosted regression.

### Decision Tree Regression

One of the most often used supervised learning techniques is the decision tree. Decision tree is a collection of trees that can be used for both classification and regression. It is a tree-structured hierarchical classifier with three sorts of nodes: root, inner, and leaf. The complete sample is represented by root nodes, which can be further divided into sub-nodes. Interior nodes are decision nodes that carry the attributes of decision rules. Decision nodes have several branches, which are called leaf nodes, and the outcome is represented by leaf nodes<sup>1</sup>. A decision tree divides each node into subsets to classify data. It travels over the entire tree, takes into account all attributes, and estimates the average of the dependent variable values from the multiple leaf nodes to produce the best results prediction. In decision tree regression, we use variance to separate variables.

$$\text{Variance} = \frac{1}{N} \sum_{i=1}^N (y_i - \mu)^2$$

where

$y_i$  = Label for an instance

$N$  = Number of instance

$\mu$  = Mean, given by  $\frac{1}{N} \sum_{i=1}^N y_i$ .

### Gradient Boosted Regression

The gradient boosting algorithm is frequently used to find any non-linear relationship from any model that uses a tabular dataset. When a machine learning model has poor predictability, gradient boosting can help to improve the model’s quality by performing model interpretation. Gradient boosted regression (GBR) is an iterative process that optimizes the predictive value of a model in every learning process. It can generalize the model by dealing with missing values and outliers. The primary goal of gradient boosting is to improve the performance of a predictive model and optimize the loss function by boosting a weak learner (the measure of the difference between the predicted and actual target values). This algorithm starts work by training a decision tree. It observe the weight of each tree and classify by its difficulty. In

each iterative approach, gradient boost combines multiple weak models and makes a better strong model by minimizing bias error.

## 4 Experiment and Analysis

### 4.1 Implementation Environment

To conduct the dataset, the Anaconda3 platform has been employed to simulate the code in Python 3.8 with a lot of packages for machine learning. The processor is an Intel Core i5-8265U (1.6 GHz base frequency, up to 3.9 GHz). The RAM is 4 GB, and the operating system is Windows 10 (64 bit).

### 4.2 Splitting Dataset

We have collected our data from a Website (UCI machine learning data repository). We have evaluated algorithms by splitting the dataset in two. We use 80% of the data for training and 20% of the data for testing.

### 4.3 Calculation Formula for Output

A root mean square error (RMSE) has been applied to analyze the output of this dataset for these two machine learning algorithms.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

where

RMSE = Root mean square error

$y_i$  = Observed value

$\hat{y}_i$  = Predicted value

$n$  = Number of observation.

### 4.4 Hyperparameter Tuning

The decision tree regression and gradient boosted regression have some hyperparameters like maximum-depth, minimum-samples-split, minimum-sample-leaf,

minimum-weight-fraction-leaf, and maximum-leaf-nodes that can be tuned to get better results. The most powerful hyperparameter is maximum-depth, which by regularizing can solve the overfitting and underfitting of decision tree regression and gradient boosted regression. In this research, the maximum-depth from 1 to 5 has been regularized, and then, the output has been distinguished.

## 4.5 *Result and Analysis*

The dataset has been analyzed in two combinations of variable selection in every machine learning method. For getting a better result, we have applied our four variables in two different ways for both algorithms. First anchor ratio, transmission range, node density, and iterations have been selected as input variables, and then, only node density and iterations have been selected as input variables, because these two variables have a high correlation coefficient with the average localization error variable that is described in Sect. 3. In addition, in each approach, the maximum-depth hyperparameter has been fine-tuned. Table 3 shows the root mean square error (RMSE) for every individual approach.

Here, we can see that gradient boosted regression for two input variables (node density and iterations) gives the best output where root mean square error 0.1379 m for this dataset.

## 4.6 *Comparison with the Previous Models*

SVR with scale-standardization, Z-score standardization, and range-standardization [100] was all simulated on this dataset. The comparison between the previous and proposed methods is shown in Table 4.

In the previous research, the best output came from range-standardization SVR with four input variables where root mean square error was 0.147 m. And in this research, the best output has come from gradient boosted regression with the last two variables, where root mean square error is 0.1379 m with a maximum-depth of 1, which is better than in the previous research. This best output has come out only for two input variables (node density and iterations), and for this, further research will be more convenient.

## 5 Conclusion

In the field of underwater sensor deployment, it is critical to accurately predict the localization error. In this research article, we have predicted average localization error using two machine learning algorithm. Decision tree and gradient boosting

**Table 3** Analysis is the output from different approaches

|                             |                       | RMSE (m)        |                 |                 |                 |                 |
|-----------------------------|-----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                             |                       | Maximum-depth-1 | Maximum-depth-2 | Maximum-depth-3 | Maximum-depth-4 | Maximum-depth-5 |
| Decision tree regression    | With 4 variables      | 0.2353          | 0.2411          | Null            | Null            | Null            |
|                             | With last 2 variables | 0.1892          | 0.1647          | 0.2343          | 0.2244          | 0.2252          |
| Gradient boosted regression | With 4 variables      | 0.1409          | 0.1451          | 0.1437          | 0.1568          | 0.1715          |
|                             | With last 2 variables | 0.1379          | 0.1393          | 0.152           | 0.1548          | 0.1585          |





regression algorithms have been conducted, and their outputs have been compared with the outputs of the previous methods; better results have been found. It can be concluded that this research will play a good role in predicting errors of average localization error (ALE) in future. In future, we hope to collect more underwater sensor data in order to operate these algorithms and find more efficient prediction of localization error.

## References

1. Singh A, Kotiyal V, Sharma S, Nagar J, Lee CC (2020) A machine learning approach to predict the average localization error with applications to wireless sensor networks. *IEEE Access* 8:208253–208263
2. Chandrasekhar V, Seah WK, Choo YS, Ee V (2006) Localization in underwater sensor networks-survey and challenges. In: 1st ACM international workshop on Underwater networks 33–40. <https://doi.org/10.1145/1161039.1161047>
3. Morelande MR, Moran B, Brazil M (2008) Bayesian node localisation in wireless sensor networks. In: 2008 IEEE international conference on acoustics, speech and signal processing. <https://doi.org/10.1109/ICASSP.2008.4518167>
4. Gharghan SK, Nordin R, Ismail M (2016) A wireless sensor network with soft computing localization techniques for track cycling applications. *Sensors (Switz)* 16
5. Ahmadi H, Bouallegue R (2017) Exploiting machine learning strategies and RSSI for localization in wireless sensor networks: a survey. In: 2017 13th international wireless communications and mobile computing conference (IWCMC). <https://doi.org/10.1109/IWCMC.2017.7986447>
6. Waadt AE, Kocks C, Wang S, Bruck GH, Jung P (2010) Maximum likelihood localization estimation based on received signal strength. In: 2010 3rd international symposium on applied sciences in biomedical and communication technologies. <https://doi.org/10.1109/ISABEL.2010.5702817>
7. Tarrío P, Bernardos AM, Casar JR (2011) Weighted least squares techniques for improved received signal strength based localization. *Sensors* 11:8569–8592
8. Coluccia A, Ricciato F (2014) RSS-based localization via Bayesian ranging and iterative least squares positioning. *IEEE Commun Lett* 18:873–876
9. Coluccia A, Fascista A (2019) Hybrid TOA/RSS range-based localization with self-calibration in asynchronous wireless networks. *J Sens Actuator Netw* 8
10. Nguyen TLN, Shin Y (2019) An efficient RSS localization for underwater wireless sensor networks. *Sensors (Switz)* 19
11. Rahman A, Muthukkumarasamy V, Sithirasenan E (2013) Coordinates determination of submerged sensors using Cayley-Menger determinant. In: Proceedings—IEEE international conference on distributed computing in sensor systems. DCoSS 2013, pp 466–471. <https://doi.org/10.1109/DCOSS.2013.62>
12. Rahman A, Muthukkumarasamy V (2018) Localization of submerged sensors with a single beacon for non-parallel planes state. In: 2018 tenth international conference on ubiquitous and future networks (ICUFN). IEEE, 2018. <https://doi.org/10.1109/ICUFN.2018.8437041>
13. Rahman MM, Tanim KM, Nisher SA (2021) Coordinates determination of submerged mobile sensors for non parallel state using Cayley-Menger determinant. In: 2021 international conference on information and communication technology for sustainable development (ICICT4SD). IEEE, 2021, pp 25–30. <https://doi.org/10.1109/ICICT4SD50815.2021.9396837>
14. Rahman MM (2021) Coordinates determination of submerged single mobile sensor using sensor's mobility. In: 2021 international conference on electronics, communications and information technology (ICECIT). Khulna, Bangladesh, 14–16 Sept 2021. <https://doi.org/10.1109/ICECIT54077.2021.9641096>

15. Bhatti MA et al (2020) Outlier detection in indoor localization and internet of things (IoT) using machine learning. *J Commun Netw* 22:236–243
16. Wang L, Er MJ, Zhang S (2020) A kernel extreme learning machines algorithm for node localization in wireless sensor networks. *IEEE Commun Lett* 24:1433–1436
17. Singh A (2021) Average localization error (ALE) in sensor node localization process in WSNs data set. UCL Machine Learning Repository, viewed 31 Dec 2022. <https://archive.ics.uci.edu/ml/>

# Chapter 4

## Machine Learning Approaches for Classification and Diameter Prediction of Asteroids



Mir Sakhawat Hossain and Md. Akib Zabed

**Abstract** In Astronomy, the size of data is increasing day by day and is becoming more complex than in previous years. Even it is also found in the study of asteroids. There are millions of asteroids to study its classification and calculating its diameter to determine their characteristics. These will help us to know which are potentially hazardous asteroids for Earth. We have applied machine learning methods to classify the asteroids and predict their diameter. For classification task, we have implemented kNN classifier, logistic regression classifier, SGD classifier, and XGBoost classifier algorithms. For the prediction of diameter, we have used linear regression, decision tree, random forest, logistic regression, XGBoost regression, kNN, and neural network models. We have depicted a comparative analysis of our results. Applying these approaches, we have gained significant 99.99% percent accuracy for asteroid classification task.

**Keywords** Asteroid classification · Diameter prediction · Machine learning modeling

## 1 Introduction

The discovery rate of asteroids especially near-Earth objects is on an average of 1000 per year [5]. The number of known asteroids has increased from 10,000 to 75,0000 with their elements, and this number is continuously increasing day by day [4]. This rate is increasing by some important factors, i.e., highly developed ground-based surveys like Catalina Sky Survey, Palomar Transient Factory, Pan-STARRS-2, etc., and space satellite-based surveys like NEOSM, NEOCam, GAIA, etc. Asteroids

---

M. S. Hossain (✉)  
International Astrostatistics Association, Milano, Italy  
e-mail: [s.hossain18@gmail.com](mailto:s.hossain18@gmail.com)

Md. A. Zabed  
Florida International University, Miami, USA  
e-mail: [mkhan149@fiu.edu](mailto:mkhan149@fiu.edu)

families come from a common origin. The resultants of origin can be a product of a collision or rotational fission of a parent body or the satellites [12]. Asteroid families originated by collisions are identified in the domain of proper elements [6] like constant  $s$  of motion over timescales of Myr [7]. For classifying asteroids, hierarchical clustering method (HCM) is widely used in proper element domain like proper semi-major axis  $a$ , proper eccentricity  $e$ , and proper inclination  $i$ . In this method, using a distance metric two asteroids in domains of proper elements or frequencies are calculated. If the distance of the second object from the first is less than the primary characteristics which are called cutoff, the object is assumed as the member of the first object asteroid family. This process is repeated until a second new asteroid is found [14]. In the last years, there are many machine learning algorithms that have been applied to solve different types of classification problems in Astronomy. The traditional HCM method is highly computationally intensive. So, we have applied different machine learning classification and regression algorithms to classify the asteroids and predict their diameter of them. Our main contributions are given below.

1. Feature analysis on asteroid data gained some important features which are mostly important to understand the characteristics of asteroids.
2. Effectively and accurately classified asteroids using different machine learning algorithms which has outperformed the traditional statistical approaches.
3. Predicted the diameter of asteroid efficiently using machine learning regressors.

This paper is organized as follows: In Sect. 2, we have given a review of related works for illustrating the previous works already happened in this Astronomy field. Section 3 describes the methodology to classify asteroids and predict the diameter of asteroids. The result and analysis of our study are presented in Sect. 4. And finally, we have concluded our work in Sect. 5, including discussion points for future works.

## 2 Literature Review

In this section, we are presenting some of the previous works related to asteroid group detection and identification and the uses of machine learning algorithms to detect and classify different asteroid-related data. A machine learning approach for identification of asteroid family groups was presented in [6]. V. Carruba et al. applied hierarchical clustering algorithms to identify and classify different collisional asteroid families where they outperformed the accuracy of traditional methods [2]. They identified six new asteroid families and 13 new clumps providing an accuracy of 89.5% and showed performance evaluation metrics for identifying different asteroid groups. Different regression algorithms and Lambert baselines predictor had been applied in [8] to give a solution to global optimization problems. A. Mereta et al. implemented application of machine learning regressors to evaluate final spacecraft mass after optimal low-thrust transfers between two near-Earth objects. They showed comparative analysis by applying different regression models: decision tree, random forest, extra trees, gradient boosting, AdaBoost, and bagging and also neural net-

works to estimate the spacecraft mass. Their accuracy outperformed the commonly implemented Lambert's estimation, but they could not completely remove the maximum initial mass ( $m^*$ ).

There are lots of studies already happened to understand different characteristics of asteroids. Most of the authors preferred machine learning algorithms and regression models to detect or predict or classify different objects or asteroids on the asteroid data. V. Pasko proposed a model to predict combination of orbital parameters for the undiscovered potentially hazardous asteroids (PHAs) [10]. He used support vector machine algorithms with a nonlinear kernel function named radial basis function (RBF) to find boundaries of different PHA subgroups. For clustering purpose, the author used density-based spatial clustering of applications with noise (DBSCAN) algorithm. In 1982, J. K. Davies et al. presented a method for asteroid classification in [3]. They studied previous works related to asteroid classification and provided a solution based on numerical taxonomy. Numerical taxonomy programs were traditionally used in microbiology, and the authors represented a dendrogram for 82 asteroids. They did not mention about any accuracy of their classification task.

A statistical model had been introduced [1] for asteroids classification task in 1987. MA Barucci et al. used G-mode analysis which is a multivariate statistical approach to classify 438 asteroids. They separated seven taxonomic units of asteroids with 99.7% confidence level by using their method. They classified asteroids into nine classes based on eight-color photometry data and geometric albedo of asteroids. A taxonomic classification model for asteroids was presented in Popescu et al. [11]. M Popescu et al. used K-nearest neighbors algorithm (KNN) and probabilistic methods to classify 18,265 asteroids on MOVIS catalog which contains near-infrared colors. They found 70% matching for identifying siliceous and carbonaceous types while doing comparison with existing model on the basis of SDSS survey. From the above studies, we can say that there are lots of works existing related to classification of asteroids. But no diameter prediction model is available for asteroid data. Here, we have proposed a model to classify 13 asteroid groups using machine learning regression algorithms and predict the diameter of asteroids efficiently.

### 3 Methodology

Our methodology is divided into four parts. They are as follows: (i) description of dataset, (ii) preprocessing of dataset, (iii) different models for asteroids classification, and (iv) diameter prediction models.

#### 3.1 Description of Dataset

The dataset consists of 958,524 rows and 45 columns. The columns consist of significant orbital parameters of the asteroids like semi-major axis  $a$ , eccentricity

e, inclination  $i$ , perihelion distance  $q$ , aphelion distance  $ad$ , period as days per, period as years per-y, etc. The columns also consist of some other important characteristics like absolute magnitude  $H$ , geometric albedo as albedo, potentially hazardous asteroid PHA or not, near-Earth objects NEO or not. The dataset is available in the Jet Propulsion Laboratory of NASA [9]. Orbital elements of Asteroids describe the state of Cartesian position and velocity in a conic orbit in space at a specific epoch. These Keplerian elements express the osculating orbit (tangent orbit approximating the actual orbit) of an object. We have assumed these elements as machine learning features. Here we have described them.

**Absolute Magnitude:** The visual magnitude denoted by  $H$  of an asteroid is measured from where observation was placed at unit heliocentric and geocentric distances at zero phase angle.

**Diameter:** Diameter of an Asteroid at unit kilometer.

**Albedo:** Albedo is the ratio of the light received by a physical body to the light reflected by the body at zero phase angle with the same position and apparent size. Its value ranges from 0 to 1.

**Eccentricity:** Eccentricity denoted by  $e$  is an orbital parameter that describes the structure of orbit. It is a ratio between foci to semi-major axis.

**Semi-major axis:** Semi-major axis of an elliptical orbit is half of the major axis. It is denoted by  $a$ .

**Perihelion distance:** Perihelion distance denoted by  $q$  is the closest distance between the orbiting body (Here, e.g., Asteroids) and the sun.

**Inclination:** Inclination denoted by  $i$  is the angle between a vector normal to the orbiting body's orbit plane and the ecliptic plane as the reference plane.

**Longitude of the ascending node:** The longitude of the ascending node denoted by  $\omega$  is the angle between the inertial frame vernal equinox and the point of passing up through the reference frame.

**Argument of perihelion:** Argument of perihelion denoted by  $w$  is the angle between the ascending node line and the perihelion point in the direction of orbit.

**Mean anomaly:** Mean anomaly denoted by  $ma$  is the product of mean motion of orbiting body and past perihelion passage.

**Aphelion distance:** Aphelion distance denoted by the  $ad$  is the farthest distance between the orbiting body (Here, e.g., Asteroids) and the sun.

**Mean motion:** Mean motion denoted by  $n$  is the angular speed at unit degree per day to complete one orbit around an ideal ellipse.

**Period:** Sidereal period at unit day.

3.2 Preprocessing of Dataset

The dataset that was collected was very well defined and well arranged, but it was needed to perform some preprocessing tasks for a better model outcome. In the dataset, some data were missing that created hindrance for machine learning modeling. We substituted the missing values with median values. We did not remove any row. Then we have analyzed the features of all attributes we have got in our dataset.

3.3 Feature Analysis

For our modeling at first, we have dropped the unnecessary features which are not related to the classifications based on Pearson correlation among the features and classes. We have used two methods extra trees classifier Table 2 and chi square Table 1 to select important features. Then we have computed variance inflation factor (VIF) Table 1 of the rest of the features. In this step, we have also dropped 6 features which may cause over fitting. We then checked the VIF factor above of 8 of the rest features. Based on feature importance, VIF factors, and correlations, we have chosen the most important 8 features. These features are absolute magnitude H, object diameter as diameter, geometric albedo as albedo, eccentricity e, semi-major axis a, perihelion distance q, inclination i, and mean motion n. And we have shuffled the dataset 5 times and then split it into train and test set with an 8 : 2 ratio.

Table 1 Feature extraction

| (a) Chi square method |               | (b) Variance inflation factor (VIF) |            |
|-----------------------|---------------|-------------------------------------|------------|
| Feature name          | Score         | Feature name                        | VIF factor |
| moid ld               | 36,815.078985 | H                                   | 2.16       |
| q                     | 21,762.640286 | diameter                            | 1.71       |
| albedo                | 17466.459663  | albedo                              | 1.29       |
| e                     | 9153.018613   | e                                   | 1.20       |
| n                     | 6853.726787   | a                                   | 1.01       |
| i                     | 3213.704662   | q                                   | 1.64       |
| H                     | 2197.257868   | i                                   | 1.12       |
| diameter              | 2084.505512   | n                                   | 1.86       |
| ma                    | 287.077008    |                                     |            |
| per                   | 284.962249    |                                     |            |



**Table 2** Extra tree classifier method

| Feature name | Score of feature importance |
|--------------|-----------------------------|
| H            | 0.053427                    |
| diameter     | 0.079550                    |
| albedo       | 0.145152                    |
| e            | 0.076104                    |
| a            | 0.100125                    |
| q            | 0.119450                    |
| i            | 0.024435                    |
| om           | 0.000915                    |
| w            | 0.000996                    |
| ma           | 0.001626                    |
| ad           | 0.050871                    |
| n            | 0.151369                    |
| per          | 0.112984                    |
| moid ld      | 0.082996                    |

**3.4 Different Models for Asteroids Classification**

In our machine learning modeling for asteroids classification task, we have applied supervised learning algorithms. We have applied K-nearest neighbors (KNN), logistic regression, stochastic gradient descent (SGD), and extreme gradient boosting (XGBoost).

**K-nearest neighbors:** K-nearest neighbors is a simple nonparametric classification method. It is used for both classification and regression problems. For the classification method, the output of this algorithm indicates a class. Here a target is classified according to the plural vote of its closest neighbors, with this target being assigned to the class which is the most common among its K-nearest neighbors where K is a positive integer. In regression problems, the target is a numerical value. The calculation process is the same, but the output will be the predicted numerical value. In our modeling, we have applied the KNN algorithm with default hyperparameters. Then we have applied *k*-fold cross-validation in this model.

**Logistic regression:** Logistic regression is a probabilistic algorithm where the probability of a class or numerical target is predicted. The mathematically logistic regression model has a dependent variable which can be one of the two possible values that are labeled 0 or 1. In this model, the logarithmic values of odds for the labeled 1 are a linear combination of one or more independent variables. Each of the independent variables can be a binary variable like two classes. The corresponding probability of the binary variable varies. But our target class is multi-label classes. So we used one vs rest classifier which is a heuristic method for the multi-label classification tasks. Then we have implemented *k*-fold cross-validation in the logistic regression model.

**Stochastic gradient descent:** Stochastic gradient descent (SGD) is a linear classifier where this is optimized by the SGD method. Here it is used to minimize the cost function. This value of logistic regression cannot be computed directly, but it is possible with the SGD method. In this procedure, we descend along with the cost function toward local minima to the global minimum. Since it is a logistic regression, here we have also applied one vs. rest classifier and the  $k$ -fold cross-validation method in this model.

**Extreme gradient boosting:** Extreme gradient boosting (XGBoost) is an ensemble learning model based on a decision tree that uses a gradient boosting framework. Trees are added to the ensemble each time and fit to minimize prediction errors by the prior model using any arbitrary differentiable loss function and gradient descent optimization. To handle multi-label target class, we have used one vs. rest classifier. Then we have implemented  $k$ -fold cross-validation in this model. Our overall procedure of applying machine learning models is illustrated in Fig. 1.

### 3.5 Diameter Prediction Models

For the diameter prediction task, we have taken only absolute magnitude  $H$  and geometric albedo as inputs. We have analyzed features of different columns of our dataset, and finally, we have taken  $H$  and albedo columns as inputs and predicted diameter of asteroids as output using seven different machine learning models. For predicting the numeric value, we have implemented regression models: linear regression, decision tree, random forest, logistic regression,  $K$ -nearest neighbors, XGBoost, and the neural network model. Linear regression, decision tree, and random forest models are very popular regressors that are mostly used for predicting numeric values. We have used logistic regression, KNN, and XGBoost for both classification and prediction tasks. The working principle of these models is the same as the classification task. But in the prediction task, we have got only one predicted numeric output which is the value of diameter where we have found different classes as the output before. A neural network model takes inputs and creates a network of neurons. These inputs are passed to hidden layers and fully connected layers. We have used a neural network model and passed  $H$  and albedo as inputs to predict the diameter as output in the output layer. We have used default hyperparameters for the learning process in all the models. While training and testing, we have calculated mean squared error, mean absolute error, and root mean square error for all the models.

## 4 Result and Analysis

To evaluate the implemented models for the classification of asteroids, we have calculated the accuracy and the performance evaluation metrics which include precision,

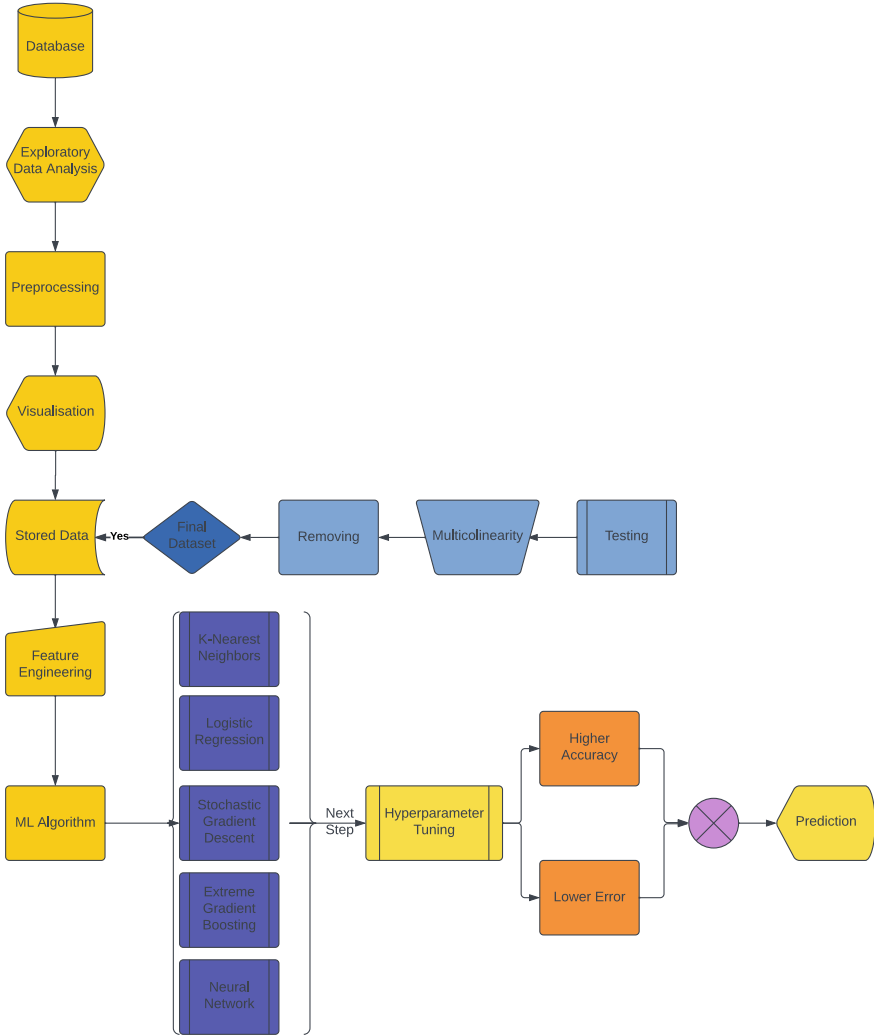


**Fig. 1** Overview of methodology

recall, and F1-score of each of the models. The following formulas [Eqs. (1)–(4)] are being used to calculate these metrics.

$$\text{Accuracy : } A = ((TP + TN)/(TP + TN + FN + FP)) * 100\% \quad (1)$$

$$\text{Precision : } P = TP/(TP + FP) \quad (2)$$

$$\text{Recall Rate : } R = TP/(TP + FN) \quad (3)$$

**Table 3** Accuracy, precision, recall, and F1-score for KNN

| Class name | Accuracy | Precision | Recall | F1-score |
|------------|----------|-----------|--------|----------|
| AMO        | 99.42%   | 0.99      | 0.98   | 0.98     |
| APO        |          | 0.99      | 0.99   | 0.99     |
| AST        |          | 0.93      | 0.93   | 0.93     |
| ATE        |          | 0.99      | 0.99   | 0.99     |
| CEN        |          | 0.99      | 0.95   | 0.97     |
| HYA        |          | 1.00      | 1.00   | 1.00     |
| IEO        |          | 1.00      | 1.00   | 1.00     |
| IMB        |          | 1.00      | 1.00   | 1.00     |
| MBA        |          | 1.00      | 1.00   | 1.00     |
| MCA        |          | 0.99      | 0.99   | 0.99     |
| OMB        |          | 0.96      | 0.86   | 0.91     |
| TJN        |          | 1.00      | 1.00   | 1.00     |
| TNO        |          | 1.00      | 1.00   | 1.00     |

F1-Score :  $F1 = (2 * Precision * Recall)/(Precision + Recall)$

(4)

Firstly, we have classified the asteroids into 13 classes using KNN, logistic regression, SGD, and XGBoost models. Table 3 shows the accuracy and class-wise evaluation metrics of the K-nearest neighbors model. We have achieved 99.42% accuracy using KNN model. And we have got 98.20% accuracy using  $k$ -fold cross-validation in this model. After cross-validation, the average precision, average recall, and average F1-score are 0.987, 0.982, and 0.984 respectively.

The accuracy and class-wise evaluation metrics of the logistic regression model are presented in Table 4. We have got 94.73% accuracy using the logistic regression model using the one vs rest classifier. After doing  $K$ -fold cross-validation in this model, we have achieved 95.06% accuracy, and the average precision, average recall, and average F1-score are 0.965, 0.960, and 0.960, respectively.

Then, we have classified the asteroids using the stochastic gradient descent model. Table 5. depicts the accuracy and class-wise evaluation metrics of the SGD model. 90.22% accuracy has been achieved using the SGD model. After using  $k$ -fold cross-validation in SGD model, we have gained 92.14% accuracy, average precision: 0.936, average recall: 0.937, and average F1-score: 0.932.

And finally, the accuracy and class-wise evaluation metrics of the XGBoost model are illustrated in Table 6. Here, we have achieved a significant accuracy which is 99.99% using the XGBoost model. We have got the same accuracy using  $k$ -fold cross-validation in XGBoost model. And the average precision, average recall, and average F1-score are 0.999, 0.999, and 0.999, respectively.

**Table 4** Accuracy, precision, recall, and F1-score for logistic regression

| Class name | Accuracy | Precision | Recall | F1-score |
|------------|----------|-----------|--------|----------|
| AMO        | 94.73%   | 0.05      | 0.02   | 0.03     |
| APO        |          | 0.98      | 0.98   | 0.98     |
| AST        |          | 0.00      | 0.00   | 0.00     |
| ATE        |          | 0.99      | 0.99   | 0.99     |
| CEN        |          | 0.47      | 0.23   | 0.31     |
| HYA        |          | 1.00      | 1.00   | 1.00     |
| IEO        |          | 0.86      | 0.75   | 0.80     |
| IMB        |          | 0.99      | 0.99   | 0.99     |
| MBA        |          | 0.98      | 1.00   | 0.99     |
| MCA        |          | 0.38      | 0.41   | 0.41     |
| OMB        |          | 0.96      | 0.51   | 0.67     |
| TJN        |          | 0.99      | 0.99   | 0.99     |
| TNO        |          | 1.00      | 1.00   | 1.00     |

**Table 5** Accuracy, precision, recall, and F1-score for SGD

| Class name | Accuracy | Precision | Recall | F1-score |
|------------|----------|-----------|--------|----------|
| AMO        | 90.22%   | 0.05      | 0.02   | 0.03     |
| APO        |          | 0.98      | 0.98   | 0.98     |
| AST        |          | 0.00      | 0.00   | 0.00     |
| ATE        |          | 0.99      | 0.99   | 0.99     |
| CEN        |          | 0.47      | 0.23   | 0.31     |
| HYA        |          | 1.00      | 1.00   | 1.00     |
| IEO        |          | 0.86      | 0.75   | 0.80     |
| IMB        |          | 0.99      | 0.99   | 0.99     |
| MBA        |          | 0.98      | 1.00   | 0.99     |
| MCA        |          | 0.38      | 0.14   | 0.21     |
| OMB        |          | 0.96      | 0.51   | 0.67     |
| TJN        |          | 0.99      | 0.99   | 0.99     |
| TNO        |          | 1.00      | 1.00   | 1.00     |

By analyzing the presented results (from Tables 1, 2, 3, and 4), we can say that the overall accuracy and performance of the XGBoost model are the best compared to other machine learning regression models to classify asteroid groups into 13 categories. Furthermore, we can understand the accuracy of a model by generating ROC-AUC curve. So, we have generated ROC curves for all the applied models. Here, we are presenting the ROC curves of the XGBoost model which is the best model for classifying asteroids. The macro-average ROC curve is illustrated in Fig. 2. For macro-average, ROC-AUC score of the model is 0.99996. And Fig. 2 presents the micro-average ROC curve of XGBoost model. For weighted by prevalence, ROC-AUC score is 1.00000 for the XGBoost model.

**Table 6** Accuracy, precision, recall, and F1-score for XGBoost

| Class name | Accuracy | Precision | Recall | F1-score |
|------------|----------|-----------|--------|----------|
| AMO        | 99.99%   | 1.00      | 1.00   | 1.00     |
| APO        |          | 1.00      | 1.00   | 1.00     |
| AST        |          | 0.96      | 0.90   | 0.93     |
| ATE        |          | 1.00      | 1.00   | 1.00     |
| CEN        |          | 1.00      | 0.98   | 0.99     |
| HYA        |          | 1.00      | 1.00   | 1.00     |
| IEO        |          | 1.00      | 1.00   | 1.00     |
| IMB        |          | 1.00      | 1.00   | 1.00     |
| MBA        |          | 1.00      | 1.00   | 1.00     |
| MCA        |          | 1.00      | 1.00   | 1.00     |
| OMB        |          | 1.00      | 1.00   | 1.00     |
| TJN        |          | 1.00      | 1.00   | 1.00     |
| TNO        |          | 1.00      | 1.00   | 1.00     |

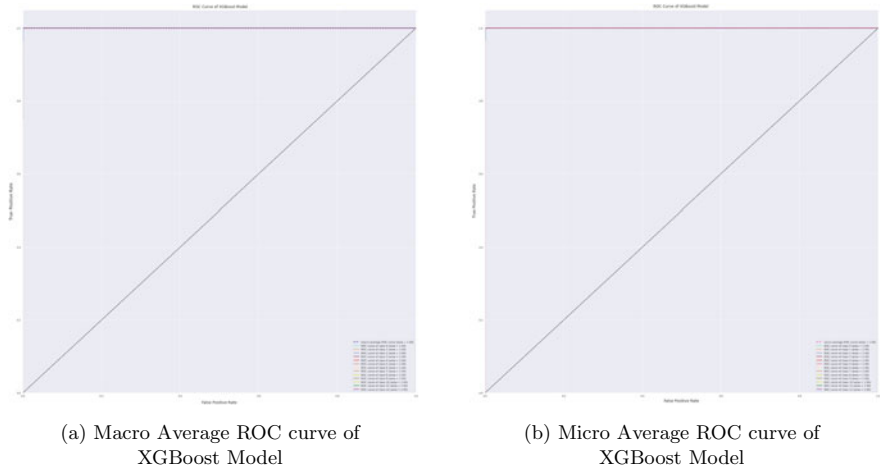


**Fig. 2** ROC curve of XGBoost model (*X*-axis: false positive rate, *Y*-axis: true positive rate)

For evaluating the applied models for diameter prediction, we have calculated mean squared error (MSE), mean absolute error (MAE), and root mean square error (RMSE) for each of the models. For predicting numeric data, the lower MSE, MAE, and RMSE determine the higher accuracy of the prediction model. A comparative analysis is depicted in Table 7 to show the MSE, MAE, and RMSE we have got implementing different machine learning regression models: (i) linear regression, (ii) decision tree, (iii) random forest, (iv) logistic regression, (v) XGBoost, (vi) K-nearest neighbors, and (vii) neural network model for predicting the diameter of the asteroids. We have achieved the lowest MSE and RMSE using XGBoost model

**Table 7** MSE, MAE, and RMSE for different machine learning models

| Model name          | MSE   | MAE  | RMSE |
|---------------------|-------|------|------|
| Linear regression   | 58.91 | 2.76 | 7.55 |
| Decision tree       | 2.84  | 0.55 | 1.68 |
| Random forest       | 2.05  | 0.53 | 1.43 |
| Logistic regression | 60.69 | 1.45 | 7.79 |
| XGBoost             | 1.84  | 0.55 | 1.36 |
| KNN                 | 2.21  | 0.49 | 1.49 |
| Neural network      | 4.74  | 0.49 | 2.18 |

which are 1.84 and 1.36, respectively, and lowest MAE: 0.49 using the K-nearest neighbors model. Overall, we can say that the XGBoost model is the best model for diameter prediction as it has the lowest RMSE among all the models.

## 5 Conclusion

We have applied eight machine learning models. Among these models, XGBoost plays the best performance for both classification and prediction tasks. The main purpose of this work is to find out the best way of asteroid classification and its diameter prediction. Here we have also got that machine learning algorithms can achieve the most accurate class and diameter. So classical methods that are computationally intensive and time consuming can be replaced with this model. In the future, we will find out the best model to detect potentially hazardous asteroids by applying different machine learning algorithms and deep learning models.

**Acknowledgements** We are grateful to the Jet Propulsion Laboratory of NASA for its amazing Asteroid Database and Sciserver [13] revolutionary data science platform for Astronomy and other scientific fields run by John Hopkins University.

## References

1. Antonietta Barucci M, Teresa Capria M, Coradini A, Fulchignoni M (1987) Classification of asteroids using g-mode analysis. *Icarus* 72(2):304–324
2. Carruba V, Aljbaae S, Lucchini A (2019) Machine-learning identification of asteroid groups. *Month Not Roy Astron Soc* 488(1):1377–1386
3. Davies JK, Eaton N, Green SF, McCheyne RS, Meadows AJ (1982) The classification of asteroids. *Vistas Astron* 26:243–251
4. DeMeo FE, Alexander CM, Walsh KJ, Chapman CR, Binzel RP (2015) The compositional structure of the asteroid belt. University of Arizona Press

5. Elvis M, Allen L, Christensen E, DeMeo F, Evans I, DePonte Evans J, Galache J, Konidaris N, Najita J, Spahr T (2013) Linnaeus: boosting near earth asteroid characterization rates. *Asteroids Comets Meteors* 45:208–224
6. Hirayama K (1922) Families of asteroids. *Jpn J Astron Geophys* 1:55
7. Knezević Z, Milani A (2003) Proper element catalogs and asteroid families. *A&A* 403(3):1165–1173
8. Mereta A, Izzo D, Wittig A (2017) Machine learning of optimal low-thrust transfers between near-earth objects. In: *Lecture notes in computer science*. Springer, pp 543–553
9. NASA (2020) JPL small-body database search engine
10. Pasko V (2018) Prediction of orbital parameters for undiscovered potentially hazardous asteroids using machine learning. Springer, Cham
11. Popescu M, Licandro J, Carvano JM, Stoicescu R, de León J, Morate D, Boaca IL, Cristescu CP (2018) Taxonomic classification of asteroids based on MOVIS near-infrared colors. *A&A* 617:A12
12. Pravec P, Vokrouhlick D, Polishook D, Scheeres DJ, Harris AW, Galád A, Vaduvescu O, Pozo F, Barr A, Longa P, Vachier F, Colas F, Pray DP, Pollock J, Reichart D, Ivarsen K, Haislip J, LaCluyze A, Kunirák P, Henych T, Marchis F, Macomber B, Jacobson SA, Krugly YuN, Sergeev AV, Leroy A (2010) Formation of asteroid pairs by rotational fission. *Nature* 466(7310):1085–1088
13. Taghizadeh-Popp M, Kim JW, Lemson G, Medvedev D, Raddick MJ, Szalay AS, Thakar AR, Booker J, Chhetri C, Dobos L, Rippin M (2020) SciServer: a science platform for astronomy and beyond. *Astron Comput* 33:100412
14. Zappala V, Cellino A, Farinella P, Knezevic Z (1990) Asteroid families. I. Identification by hierarchical clustering and reliability assessment. *Am Astron Soc* 100:2030



# Chapter 5

## Detection of the Most Essential Characteristics from Blood Routine Tests to Increase COVID-19 Diagnostic Capacity by Using Machine Learning Algorithms



Faria Rahman and Mohiuddin Ahmad

**Abstract** The regular blood test is the most common and consistent initial test for COVID-19 patients, and the results are quite obtainable within two hours. Due to the high volume of patients admitted to hospitals and the scarcity of medical resources, a regular blood test may be the only way to check for COVID-19 when patients initially visit hospitals. It might be difficult to quickly identify people who are most vulnerable to disease due to improper distribution of RT-PCR-based test equipment. Some tools and resources are required for frequently monitoring patients for optimal treatment. So, keeping up with it regularly is challenging. As a result, a routine blood test allows patients to be monitored daily. In our proposed work, we have attempted to identify the most impacted characteristics that have the strongest effect on the target. So far, we have focused on determining frequently occurring indicators. Then we have used random forest, k-nearest neighbor, decision tree, support vector machine, and naive bayes machine learning approaches and established a stacking technique with those four base learners and one meta learner to effectively justify the outcome. There are four forms of splitting, and each has the best output with an accuracy of more than equal to 84.75%. Based on this, we've discovered that Age and LYMPH are commonly active indicators.

**Keywords** COVID-19 · Features ranking · RT-PCR test · Classification · Routine blood test

---

F. Rahman (✉) · M. Ahmad  
Institute of Information and Communication Technology (IICT), Khulna, Bangladesh  
e-mail: [fariarahman337@gmail.com](mailto:fariarahman337@gmail.com)

M. Ahmad  
Department of Electrical and Electronic Engineering (EEE), Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh

# 1 Introduction

Coronavirus disease 2019, generated by SARS-CoV-2, features a biphasic illness pattern that is thought to be the combined effect of an early viral response phase and a later inflammatory phase. It has a greater reproductive number ( $R_0$ ) than SARS-CoV-1, indicating that it spreads significantly more quickly. SARS-surface CoV-2's proteins have structural changes that allow them to connect to the ACE 2 receptor more effectively and infect host cells more efficiently. SARS-CoV-2 also has a stronger bonding with the upper respiratory tract and conjunctiva, making it easier to infect and conduct airways. The majority of clinical manifestations are mild, and the normal pattern of COVID-19 is more related to influenza-like illness—cough, fever, malaise, headache, myalgia, and smell and taste disturbance than severe pneumonia. COVID-19 has a variety of effects on different persons. Despite this, over a maximum of infected patients experience mild to moderate sickness and recover without the need for hospitalization. In a small percentage of individuals, however, serious symptoms such as loss of speech, shortness of breath, and chest pain may occur. Those who become very ill are more likely to be older and male, with each decade beyond the age of 50 has a greater risk. People with medical issues such as cancer, diabetes, chronic respiratory, and cardiovascular disease are also more likely to acquire serious illnesses [1]. Even though there are no particular treatments for COVID-19, several active clinical trials are examining potential remedies [2]. PCR testing capability is still limited in many hospital settings. Many hospitals lack on-site PCR capabilities and must send samples to centralized labs. The turnaround time can be extended by up to 96h due to transportation delays and queues. The decision-making process in the clinical setting is slowed, and valuable personal protective equipment (PPE) is wasted. Machine learning algorithms were used to detect COVID-19 from lung CT images with 90% sensitivity and a 0.95 AUROC [3, 4]. Although chest CTs have been linked to high sensitivity for the diagnosis of COVID-19 [5], this type of test is rarely used for screening activities due to the high radiation doses, less equipment available, and associated operating costs.

In our proposed work, we have tried to figure out which features are the most influenced and have the most impact on the target. We've concentrated on identifying commonly occurring signs. Then, to effectively justify the observation, we have applied four machine learning methodologies and built a stacking methodology with four base learners and one meta learner. This paper contributes in the following aspects:

1. To differentiate frequently regulating highly active attributes that relate to the target most.
2. To construct an ensemble technique to verify the output most accurately.
3. To specify the highlighted indicators that ensure the early detection of COVID-19.
4. To optimize cost, time, and tools for regular monitoring COVID-19 patient.

## 2 Related Work

Luo et al. [6] have suggested a multi-criteria decision-making (MCDM) algorithm that uses the technique for order of preference by similarity to ideal solution (TOPSIS) and the Naive Bayes (NB) classifier. However, Age, WBC, LYMC, and NEUT were the most important attributes in predicting COVID-19 severity. Alakus et al. [7] have used deep learning and laboratory data to develop clinical predictive models that indicate which patients are at risk of developing COVID-19 disease. To build predictions, they have analyzed data such as hemoglobin, hematocrit, red blood cells, platelets, basophils, leukocytes, eosinophils, lymphocytes, and others. Goodman-Meza et al. [8] have employed a combination of seven machine learning models for the final diagnostic evaluation. In comparison with the SARS-CoV-2 PCR, they have discovered that their machine learning approach has provided superior diagnostic metrics. They have tried to explain that, in hospitals where PCR testing is limited or non-existent, it could be used as a screening tool. Cabitza et al. [9] have used the overall OSR dataset (CBC, biochemical, coagulation, hemogas analysis, and CO-Oxymetry values, as well as age, gender, and particular symptoms at triage) and two sub-datasets to create machine learning models (COVID-specific and CBC dataset). They have created five machine learning models, with AUCs ranging from 0.83 to 0.90 for the entire OSR dataset, 0.83–0.87 for COVID-specific datasets, and 0.74–0.86 for the CBC dataset. According to their findings, regular blood tests' hematochemical values can be a quicker alternative for coronavirus disease identification. Ferrari et al. [10] have observed the plasma levels of 207 individuals who were examined for RT-PCR after being hospitalized. WBC, CRP, AST, ALT, and LDH all showed statistically significant changes in their studies. Using empirical AST and LDH thresholds, 70% of COVID-19 positive and negative individuals have been identified based on routine blood test findings. AlJame et al. [11] have applied deep forest (DF), a new ensemble-based technique that uses numerous classifiers at multiple levels to encourage variation and increase performance. In some places where testing is rare, their approach has functioned as a quick screening tool for COVID-19 patients using clinical and/or routine laboratory data. Jiang et al. [12] have proposed that a decently raised alanine aminotransferase (ALT) (a liver enzyme), the presence of myalgia's (body aches), and rising hemoglobin (red blood cells) are the most predictive clinical characteristics. The predictive algorithms use historical data to assist in forecasting who may get ARDS, a significant result in COVID-19.

### Justification

Luo et al. [6] in their work have suggested a multi-criteria decision-making algorithm, which uses the technique for order of preference by similarity to ideal solution and the Naive Bayes classifier to pick effective indicators from patients' initial blood test results where they have gained AUC of 0.93 and an accuracy of 0.82.

In our proposed work, we have used the same dataset where applied rank base five feature selection methods and individually run attribute by applying machine

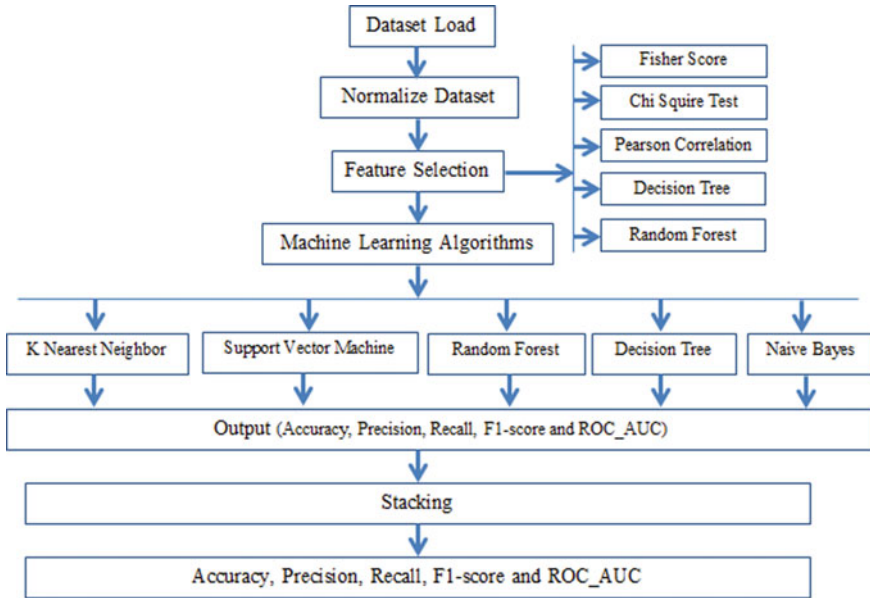


**Fig. 1** Flowchart of the working procedure

learning algorithms to find out the individual impact on the target class. We have extracted {Age, LYMC, LYMPH} subset that has given accuracy 90.00%, precision 86.67%, recall 93.10%,  $f1$ -score 88.60%, and roc\_auc 93.10%.

### 3 Proposed Work

In our proposed work first, we have collected the dataset [6] and then applied five feature selection methods Fisher score, the chi-square test, Pearson's correlation, decision tree, and random forest. Before applying any classifiers, we have normalized the dataset by using MinMaxScaler. Then have applied five classifiers random forest (RF), k-nearest neighbor (KNN), decision tree (DT), naive bayes (NB), support vector machine (SVM), and stacking for further checking of the performances. Stacking explains that we should not rely on a single decision, but rather make several and choose based on the total outputs [13]. We have also taken the help of these classifiers and ensemble techniques (stacking) for checking the performances of individual features how nearly each stand to the target class. In this case, we have run algorithms for checking individual involvement by fivefold classification. We have split our dataset for the classification process both 90% training-10% testing, 80% training-20% testing, 70% training-30% testing, and 60% training-40% testing (Fig. 1).

### **3.1 Dataset Description**

The data was collected at Wuhan Red Cross Hospital, a COVID-19 treatment center in Wuhan, China, from March 15 to March 20, 2020. From February 1, 2020, to March 15, 2020, where 196 COVID-19 patients were diagnosed using WHO guidelines [14] and the following inclusion criteria were: COVID-19 pneumonia has been detected, according to WHO guidelines are given on January 28, 2020, and essential medical data records, such as initial blood test results when patients first come to a fever clinic in a community hospital and patient severity, are commonly accessible. Patients who were discharged within 24 h of their hospitalizations were not included. The dataset includes Gender, age, Lymphocyte Ratio (LYMPH), Neutrophil to Lymphocyte Ratio (NLR), Lymphocyte Count (LYMC), White Blood Cell Count (WBC), Neutrophil Ratio (NEU), Neutrophil Count (NEUT), and Severity of these attributes.

### **3.2 Feature Selection**

The technique of minimizing the number of random features under consideration by obtaining a collection of major or relevant features is known as dimensionality reduction. The reduction of data allows the computer to develop the model with optimal effort as well as speed up the learning and generalization process in machine learning. There are frequently too many characteristics on which the final prediction is made. The more characteristics there are, the more difficult it is to understand the training set and subsequently work on it. Many of these characteristics are sometimes connected or redundant in situations where dimensionality reduction methods are more reliable and effective. When we need to minimize the number of resources required for processing without sacrificing critical information, the feature selection procedure is helpful. Feature selection can also help us cut down on the amount of redundant data in our research.

In our proposed work, we have suggested the Fisher score, chi-square test, Pearson's correlation, decision tree, and random forest feature selection methods, those are discussed below for further clarification.

#### **3.2.1 Fisher Score**

One of the most comprehensively used supervised feature selection approaches is the Fisher score. The technique we have employed returns the ranks of the variables in descending order depending on Fisher's score. The variables can then be chosen according to the circumstances.

### 3.2.2 Chi-Square Test

The chi-square test is used to determine the significance of features in a dataset. After calculating the chi-square between each feature and the target [15], we have determined the desired number of features with the highest chi-square scores.

### 3.2.3 Pearson's Correlation

Using Pearson's correlation coefficient-based technique, the optimal characteristics are selected while the redundant features are removed and the most connected features to the target are taken. Dependency measurements are one of the criteria for selecting features. Many strategies based on dependency have been proposed where the correlation-based technique is the main focus. Pearson's correlation technique is used to determine the relationship between features and the target class. We have also used kNN, SVM, RF, DT, NB classifiers, and the ensemble technique (Stacking) with fivefold cross-validation to examine individual feature performances and see how close they are to the desired class.

$$ML = \text{mean}(\text{KNN, SVM, RF, DT, NB, Stacking}) \quad (1)$$

### 3.2.4 Decision Tree

Calculating the best predictive feature is one of the key steps in building a decision tree. Tree-based models determine feature importance because the best-performing features must be kept as close to the tree's root as possible. In our research, the relevance of features in tree-based models is measured using the Gini Index value.

### 3.2.5 Random Forest

Random forests are effective because they have strong detection ability, low overfitting and are simple to comprehend in general. The fact that it is simple to calculate the relevance of each factor in the tree decision contributes to its interpretability. Calculating how much each variable contributes to the decision is simple. Random forest feature selection belongs to the area of embedded techniques.

## 3.3 Classifiers

Classifiers are the intersection of advanced machine theory and real-world application. Classification algorithms are more than just a way to arrange or 'map' unlabeled data into distinct classifications. Classifiers include a unique collection of dynamic

rules, including an interpretation mechanism for dealing with ambiguous or unknown values, all of which are suited to the type of inputs under consideration. The majority of classifiers also use probability estimations, which allow end-users to adjust data classification using utility functions. In our proposed work, we have applied five machine learning algorithms. Those are KNN, SVM, RF, DT, and NB.

### **3.3.1 K-Nearest Neighbor**

The KNN algorithm predicts the values of new data points based on ‘feature closeness,’ which implies that the new data point will be assigned a value depending on how similarly it matches the points in the training set. In our proposed work, we have applied KNN with five neighbors and weights as uniform parameters.

### **3.3.2 Support Vector Machine**

The support vector machine, or SVM, is mostly utilized in machine learning for classification difficulties. The SVM algorithm’s goal is to find the better line or decision boundary for categorizing  $n$ -dimensional space into classes so that new data points can be easily placed in the appropriate category in the future. In the proposed work, we have applied SVM with a linear kernel for classification purposes.

### **3.3.3 Random Forest**

Random forest is a supervised learning technique that can be used to classify and predict data. A forest, as we all know, is made up of trees, and more trees equal a healthier forest. Similarly, the random forest method constructs decision trees from data samples, extracts predictions from each, and then votes on the best option. In our proposed work, we have run random forest classification with hundred numbers of trees in the forest and the ‘gini’ criterion.

### **3.3.4 Decision Tree**

The aim of using a decision tree is to create a training structure that can be used to predict the target variable’s class or value using basic decision rules learned from previous data. In this case, we used a decision tree with the ‘gini’ criterion.

### **3.3.5 Naive Bayes**

The Naive Bayes is a supervised learning algorithm that uses Bayes’ theorem to solve classification problems. Bayes’ theorem, commonly referred to as Bayes’ rule or

Bayes’ law, is a mathematical formula for determining the probability of a hypothesis based on previous knowledge.

We have trained our dataset further by applying the ensemble technique (Stacking) with KNN, SVM, RF, DT, and NB as a base learner and logistic regression as meta learner.

### 4 Outcome

This work has been implemented using the Python 3 programming language and the Scikit package. Before using five different feature selection procedures in our proposed work, we have preprocessed our data. Fisher score, chi-square test, Pearson’s correlation, decision tree, and random forest feature selection methods have all been used to find features. We have also individually run attributes by applying five classifiers with a fivefold cross-validation process. Next, we have utilized five different classification algorithms after normalizing the dataset. We have used a stacking ensemble strategy with KNN, SVM, RF, DT, and NB as the base learner to improve the model’s efficiency. Based on rank, features from all attribute selection methods are listed below. The order of features is displayed in Table 1.

We have taken the top three features from all attribute selection methods and the top three attributes that have been individually checked by applying classifiers. All are shown in Table 2.

**Table 1** Rank-based features

|                       |            |        |        |        |        |       |        |        |        |
|-----------------------|------------|--------|--------|--------|--------|-------|--------|--------|--------|
| Chi-square test       | Attributes | Age    | NLR    | LYMPH  | NEU    | NEUT  | WBC    | LYMC   | Gender |
| Pearson’s correlation | Score      | 273.55 | 262.55 | 256.65 | 117.34 | 32.54 | 12.56  | 10.41  | 0.7    |
|                       | Attributes | Age    | LYMPH  | NEU    | LYMC   | NEUT  | NLR    | WBC    | Gender |
| Decision tree         | Score      | 0.57   | 0.44   | 0.43   | 0.42   | 0.27  | 0.24   | 0.19   | 0.08   |
|                       | Attributes | Age    | NLR    | NEU    | WBC    | LYMPH | NEUT   | LYMC   | Gender |
| Random forest         | Score      | 0.53   | 0.12   | 0.11   | 0.09   | 0.06  | 0.05   | 0.03   | 0.02   |
|                       | Attributes | Age    | NLR    | LYMC   | LYMPH  | NEUT  | NEU    | WBC    | Gender |
| Fisher score          | Score      | 0.31   | 0.14   | 0.13   | 0.12   | 0.10  | 0.09   | 0.08   | 0.02   |
|                       | Attributes | NEUT   | WBC    | LYMPH  | Age    | LYMC  | NEU    | Gender | NLR    |
| ML                    | Rank       | 1      | 2      | 3      | 4      | 5     | 6      | 7      | 8      |
|                       | Attributes | Age    | LYMC   | LYMPH  | NEU    | NLR   | Gender | WBC    | NEUT   |
|                       | Rank       | 1      | 2      | 3      | 4      | 5     | 6      | 7      | 8      |



**Table 2** Rank-based top three features

| Rank | Chi-square test | Pearson's correlation | Decision tree | Random forest | Fisher score | ML    |
|------|-----------------|-----------------------|---------------|---------------|--------------|-------|
| 01   | Age             | Age                   | Age           | Age           | LYMPH        | Age   |
| 02   | NLR             | LYMPH                 | NLR           | NLR           | WBC          | LYMC  |
| 03   | LYMPH           | NEU                   | NEU           | LYMC          | NEUT         | LYMPH |

**Table 3** Performances using top three features using (60-40)% split

| Training 60% and Testing 40% (Accuracy %) |                       |                 |               |               |              |       |
|-------------------------------------------|-----------------------|-----------------|---------------|---------------|--------------|-------|
|                                           | Pearson's correlation | Chi-square test | Random forest | Decision tree | Fisher score | ML    |
| DT                                        | 77.22                 | 75.95           | 78.48         | 78.48         | 59.49        | 79.75 |
| KNN                                       | 84.81                 | 81.01           | 81.01         | 77.22         | 69.62        | 77.22 |
| SVM                                       | 77.22                 | 78.48           | 82.28         | 79.75         | 65.82        | 79.75 |
| RF                                        | 78.48                 | 81.01           | 81.01         | 81.01         | 64.56        | 81.01 |
| NB                                        | 75.95                 | 77.22           | 78.48         | 77.22         | 59.49        | 77.22 |
| Stacking                                  | 77.22                 | 79.75           | 82.28         | 78.48         | 75.95        | 79.75 |

**Table 4** Performances using top three features using (70-30)% split

| Training 70% and Testing 30% (Accuracy %) |                       |                 |               |               |              |       |
|-------------------------------------------|-----------------------|-----------------|---------------|---------------|--------------|-------|
|                                           | Pearson's correlation | Chi-square test | Random forest | Decision tree | Fisher score | ML    |
| DT                                        | 80.00                 | 80.00           | 75.00         | 82.50         | 67.80        | 80.00 |
| KNN                                       | 84.75                 | 84.75           | 83.05         | 77.97         | 76.27        | 79.66 |
| SVM                                       | 81.36                 | 81.36           | 84.75         | 84.75         | 67.80        | 83.05 |
| RT                                        | 81.36                 | 81.36           | 83.05         | 79.66         | 74.58        | 79.66 |
| NB                                        | 76.27                 | 79.66           | 79.66         | 79.66         | 59.32        | 81.36 |
| Stacking                                  | 81.36                 | 83.05           | 83.05         | 79.66         | 74.58        | 81.36 |

By using Pearson's correlation feature selection, we have extracted the top three features Age, LYMPH, and NEU. Then, we have selected the best three characteristics Age, NLR, and LYMPH using the chi-square test feature selection. From the random forest feature selection, we have chosen the top three features Age, NLR, and LYMPH. Age, NLR, and NEU are all factors to consider using the DT. We have retrieved the top three features LYMPH, WBC, and NEUT using Fisher score feature selection. Machine learning has been used to discover the top three characteristics, Age, LYMC, and LYMPH.

To explore how the dataset responds to taking a lesser amount of training set, we have performed a 60%-40%, 70%-30% split. Tables 3 and 4 show the performances below.

**Table 5** Performances using top three features using (90-10)% split

| Training 90% and Testing 10% (Accuracy %) |                       |                 |               |               |              |       |
|-------------------------------------------|-----------------------|-----------------|---------------|---------------|--------------|-------|
|                                           | Pearson's correlation | Chi-square test | Random forest | Decision tree | Fisher score | ML    |
| DT                                        | 75.00                 | 85.00           | 80.00         | 8500          | 75.00        | 80.00 |
| KNN                                       | 90.00                 | 90.00           | 85.00         | 85.00         | 75.00        | 85.00 |
| SVM                                       | 90.00                 | 90.00           | 85.00         | 90.00         | 65.00        | 85.00 |
| RT                                        | 80.00                 | 85.00           | 85.00         | 85.00         | 70.00        | 80.00 |
| NB                                        | 80.00                 | 85.00           | 85.00         | 85.00         | 75.00        | 85.00 |
| Stacking                                  | 80.00                 | 85.00           | 95.00         | 85.00         | 80.00        | 90.00 |

**Table 6** Accuracies of all classifiers for each feature selection

| Training 80% and Testing 20% (Accuracy %) |                 |                       |               |               |              |       |
|-------------------------------------------|-----------------|-----------------------|---------------|---------------|--------------|-------|
|                                           | Chi-square test | Pearson's correlation | Decision tree | Random forest | Fisher score | ML    |
| DT                                        | 80.00           | 80.00                 | 82.50         | 75.00         | 65.00        | 80.00 |
| KNN                                       | 87.50           | 85.00                 | 82.50         | 82.50         | 75.00        | 85.00 |
| SVM                                       | 87.50           | 87.50                 | 87.50         | 90.00         | 70.00        | 87.50 |
| RT                                        | 80.00           | 77.50                 | 80.00         | 85.00         | 67.50        | 80.00 |
| NB                                        | 82.50           | 80.00                 | 79.66         | 80.00         | 60.00        | 90.00 |
| Stacking                                  | 85.00           | 82.50                 | 85.00         | 82.50         | 77.50        | 82.50 |

From Table 3 we can see that using Pearson’s correlation feature selection KNN has performed well where the accuracy of 84.81%, a precision of 82.75%, a recall of 85.74%, a *f*1-score of 83.66%, and a *roc\_auc* of 85.74%. Table 4 shows that both Pearson’s correlation and chi-square test feature selection behaved well, with an accuracy of 84.75%, precision of 82.53%, recall of 87.47%, *f*1-score of 83.53%, and *roc\_auc* of 87.46% obtained from KNN. We’ve also executed all of the procedures with the (90-10)% split to see how it behaves.

Table 5 shows that stacking has worked well utilizing random forest feature selection, with accuracy 95.00%, precision 92.86%, recall 96.43%, *f*1-score 94.30%, and *roc\_auc* 96.42%. We have also checked all the performances by applying 80% training and 20% testing split.

From Table 6, we can observe that random forest feature selection with SVM and ML with Gaussian Naïve Bayes both have given 90% accuracy but ML with Gaussian Naïve Bayes overall perform better than others. From the above results, we can define easily that using 80% training and 20% testing split, ML with Gaussian Naïve Bayes has given an overall better result where the accuracy of 90.00%, the precision of 86.67%, recall of 93.10%, *f*1-score of 88.60%, and *roc\_auc* of 93.10%, applying. Table 7 shows a comparison between the existing work and our proposed work (Figs. 2 and 3).

**Table 7** Comparison of the proposed method with previous work

|                        | Related paper [6]                          | Proposed work                |
|------------------------|--------------------------------------------|------------------------------|
| Most important factors | Age, WBC, LYMC, and NEUT                   | Age and LYMPH                |
| Subsets                | 1. {Age, WBC, LYMC, NEUT}                  | {Age, LYMC, LYMPH}           |
|                        | 2. {Age, WBC, LMYC}                        |                              |
|                        | 3. {Age, NEUT, LYMC}                       |                              |
| Split                  | Training = 80%                             | Training = 80%               |
|                        | Test = 20%                                 | Test = 20%                   |
| Algorithms             | MCDM algorithm with TOPSIS and Naive Bayes | ML with Gaussian Naive Bayes |
| Performances           | Accuracy = 82.00%                          | Accuracy = 90.00%            |
|                        | roc_auc = 93.00%                           | roc_auc = 93.10%             |

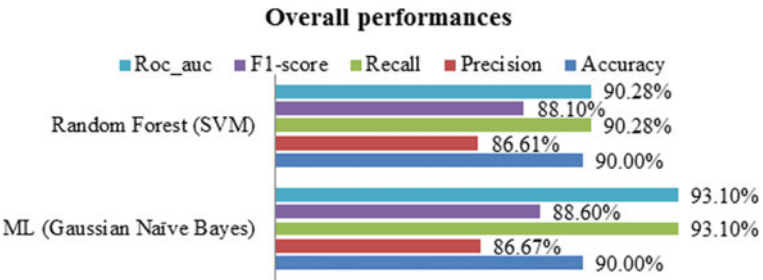


**Fig. 2** Overall performances of ML (Gaussian Naive Bayes) and random forest (SVM) using 20% testing

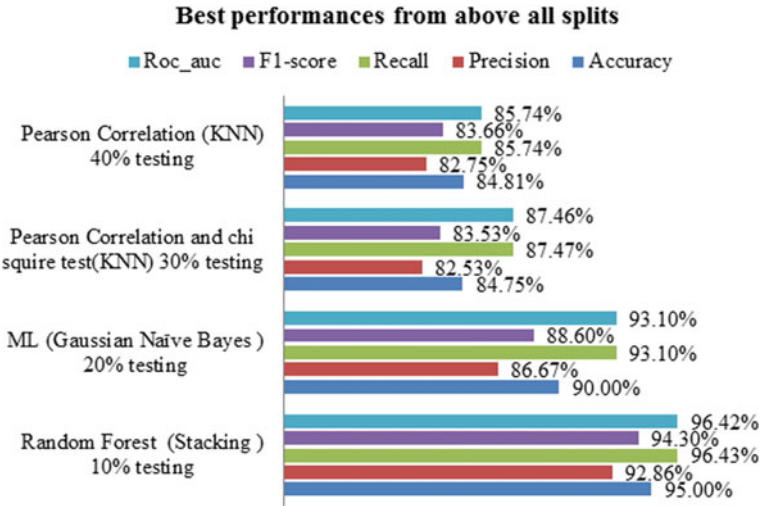


**Fig. 3** Performances among all top outcomes

## 5 Conclusions

We have employed six different feature selection methods to find the far more prevalent qualities that can be traced with a decent sign across the output we've proposed. The model is divided into four stages: first, we have taken relevant data; second, we have applied six methods for selecting features to describe the far more similar characteristics that are mapped with a decent marker over the output; third, we have used random forest, k-nearest neighbor, decision tree, support vector machine, naive bayes; and fourth, we have utilized stacking using all of those five classifiers to check the outcome even more thoroughly. From above all four types of split, we can easily explain that in each split there have best performances where the accuracy is greater than or equal to 84.75%, based on this we can find out that Age and LYMPH are frequently active indicators. Because we have identified four subsets that perform well {Age, LYMPH and NEU}, {Age, NLR, and LYMPH}, {Age, LYMC and LYMPH}, and {Age, NLR, and LYMC}.

It may be challenging to effectively identify the most susceptible people to the disease due to inappropriate distribution of RT-PCR test equipment where blood testing may be a preferable option. In the future, additional data can be collected to explore the work.

## References

1. Coronavirus (n.d.) <https://www.who.int/westernpacific/health-topics/coronavirus>. Accessed 5 Apr 2022
2. Wang F, Nie J, Wang H, Xiao Y, Wang H, Liu X et al (2020) Characteristics of peripheral lymphocyte subset alteration in 2019-nCoV pneumonia. SSRN Electron J. <https://doi.org/10.2139/ssrn.3539681>
3. Gozes O, Frid-Adar M, Sagie N, Zhang H, Ji W, Greenspan H (2020) Coronavirus detection and analysis on chest CT with deep learning. [arXiv:2004.02640](https://arxiv.org/abs/2004.02640) [cs, eess]. Accessed 5 Apr 2022
4. Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B et al (2020) Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy. *Radiology* 296(2):E65–E71. <https://doi.org/10.1148/radiol.20200905>
5. Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W et al (2020) Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. *Radiology* 296(2):E32–E40. <https://doi.org/10.1148/radiol.20200642>
6. Luo J, Zhou L, Feng Y, Li B, Guo S (2021) The selection of indicators from initial blood routine test results to improve the accuracy of early prediction of COVID-19 severity. *PLoS One* 16(6):e0253329. <https://doi.org/10.1371/journal.pone.0253329>
7. Alakus TB, Turkoglu I (2020) Comparison of deep learning approaches to predict COVID-19 infection. *Chaos Solitons Fractals* 140:110120. <https://doi.org/10.1016/j.chaos.2020.110120>
8. Goodman-Meza D, Rudas A, Chiang JN, Adamson PC, Ebinger J, Sun N et al (2020) A machine learning algorithm to increase COVID-19 inpatient diagnostic capacity. *PLoS One* 15(9):e0239474. <https://doi.org/10.1371/journal.pone.0239474>
9. Cabitza F, Campagner A, Ferrari D, Di Resta C, Ceriotti D, Sabetta E et al (2021) Development, evaluation, and validation of machine learning models for COVID-19 detection based on routine

- blood tests. *Clin Chem Lab Med (CCLM)* 59(2):421–431. <https://doi.org/10.1515/cclm-2020-1294>
10. Ferrari D, Motta A, Strollo M, Banfi G, Locatelli M (2020) Routine blood tests as a potential diagnostic tool for COVID-19. *Clin Chem Lab Med (CCLM)* 58(7):1095–1099. <https://doi.org/10.1515/cclm-2020-0398>
  11. AlJame M, Imtiaz A, Ahmad I, Mohammed A (2021) Deep forest model for diagnosing COVID-19 from routine blood tests. *Sci Rep* 11(1):16682. <https://doi.org/10.1038/s41598-021-95957-w>
  12. Jiang X, Coffee M, Bari A, Wang J, Jiang X, Huang J et al (2020) Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. *Comput Mater Continua*. <https://www.techscience.com/cmc/v63n1/38464>. Accessed 5 Apr 2022
  13. Rahman F, Mehejabin T, Yeasmin S, Sarkar M (2020) A comprehensive study of machine learning approach on cytological data for early breast cancer detection. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT). Proceedings of ICCCNT, Kharagpur, India. IEEE, pp 1–6. <https://doi.org/10.1109/ICCCNT49239.2020.9225448>
  14. World Health Organization (2020) Clinical management of severe acute respiratory infection when novel coronavirus (2019-nCoV) infection is suspected: interim guidance, 28 Jan 2020 (No. WHO/nCoV/Clinical/2020.3). World Health Organization. <https://apps.who.int/iris/handle/10665/330893>. Accessed 5 Apr 2022
  15. Rahman F, Ashiq Mahmood Md (2022) A comprehensive analysis of most relevant features causes heart disease using machine learning algorithms. In: Proceedings of the international conference on big data, IoT, and machine learning, vol 95. Springer, Singapore, pp 63–73. [https://doi.org/10.1007/978-981-16-6636-0\\_6](https://doi.org/10.1007/978-981-16-6636-0_6)

# Chapter 6

## Usability Evaluation of Adaptive Learning System Rhapsode™ Learner



Md. Saifuddin Khalid , Tobias Alexander Bang Tretow-Fish ,  
and Amalie Roark

**Abstract** From a usability perspective, designing and evaluating an adaptive learning system involve complexities associated with adaptivity and the diverse requirements from content designers, educators, and students. Moreover, students and educators are increasingly getting subjected to educational quality studies including the evaluation of digital learning technologies, which the desired participants typically do not perceive as valuable. Furthermore, very few case studies report the results of the usability evaluation of adaptive learning systems. Thus, this study addresses the research questions: (1) Considering low participation and time constraint, which established usability evaluation methods might be applied for mixed-methods study of adaptive learning systems? (2) What is the students' satisfaction score of Rhapsode™ Learner and implications for redesign? The result is the selection and application of system usability scale (SUS) and co-Discovery with concurrent interviews for the evaluation of Rhapsode™ Learner. The average score of 53.8 ( $n = 15$ ) shows low marginal acceptability with good internal consistency (10 items, Cronbach's  $\alpha = 0.863$ ). Analyses of the co-discoveries using 12 interaction design principles enable identifying the scopes for redesigning technological factors. For the evaluation of pedagogy and content aspects, further instruments should be developed.

**Keywords** Adaptive learning · Usability evaluation · System usability scale · Co-discovery · Design principles

---

Md. S. Khalid (✉) · T. A. B. Tretow-Fish · A. Roark  
Technical University of Denmark, Lyngby, Denmark  
e-mail: [skhalid@dtu.dk](mailto:skhalid@dtu.dk)

T. A. B. Tretow-Fish  
e-mail: [tabtr@dtu.dk](mailto:tabtr@dtu.dk)

A. Roark  
e-mail: [s184227@dtu.dk](mailto:s184227@dtu.dk)

# 1 Introduction

Adaptive learning systems are integrated into study programs for improving educational quality and students' satisfaction. From usability perspective, designing and evaluating an adaptive learning system involve complexities associated with adaptivity and the diverse requirements from content designers, educators, and students. The flexibilities of adaptive learning platforms result in distinctive design and interaction opportunities for each of the courses. The company Area9 has developed three interconnected and role-based adaptive learning platforms: Rhapsode™ Curator, Rhapsode™ Educator, and Rhapsode™ Learner, for both educational institutions and employee training. The Curator platform is for the design of content and activities, Educator platform is for providing insights to and interaction by educators, and the Learner platform is for students. Usability evaluation of each of the applications must be conducted from the respective roles, for instance, learners should evaluate the Rhapsode™ learner platform. The content, interaction, and experience of the same role may vary across courses, and thereby, the satisfaction score will vary even if the platform's effectiveness and efficiency measures [1, p. 256] are satisfactory from experts' evaluation. Due to the customized design of each of the courses and associated learning resources, each of the courses should be separately evaluated by the students. Because, satisfaction factors depend on customized design of content and perceived experience.

Usability evaluation provides satisfaction measures, which are defined by the ISO 9241 part 11 and allows determining system acceptability. The satisfaction measures achieved by quantitative methods enable further studies to improve the design and implementation of the platform, content, and pedagogy. This study conducts a usability evaluation of the Rhapsode™ Learner platform as part of a three-year project on the integration of the platform in the Nursing Education program of University College Absalon in Denmark. The satisfaction measures associated with the adaptive learning platform are part of both usability matrices and the quality of the nursing education. In the learning technology integration process, students and educators are invited to participate in overwhelming amounts of educational quality studies, which are typically not perceived as valuable from the participants' perspective. Researchers, e-learning consultants, and the management roles do not want to burden the educators and students with large questionnaires and other time-consuming evaluation methods. Furthermore, low participation is a challenge for researchers in the individualistic culture of Scandinavia. Thus, this study shares the dilemma associated with research reliability due to low participation and contributes by demonstrating the evaluation of an adaptive learning platform by applying two usability evaluation methods that take less time from participants.

## 2 Related Work

The original definition of usability is that systems should be easy to use, easy to learn, flexible and should engender a good attitude in people [2]. Benyon [1] documented various methods for the evaluation of usability and the broader scope of user experience. Cooperative evaluation, participatory heuristic evaluation, co-discovery, and controlled experiments are categorized as participant-based evaluation [1]. Some of the validated usability questionnaires are Questionnaire for User Interface Satisfaction (QUIS) from the University of Maryland, Software Usability Measurement Inventory (SUMI) from the University of Cork, User Experience Questionnaire (<http://www.ueq-online.org>), and the system usability score (SUS) [1].

### 2.1 Usability Questionnaires and SUS for e-Learning

The SUMI questionnaire consists of 50 statements which the users are asked to answer on a 3-point Likert-scale. Fulfillment can be achieved with as few as 12 respondents, according to the creator Dr. Jurek Kirakowski and the method provides researchers with valid and reliable measurement of users' perception of product usability [3, 4]. "SUMI has been used effectively to: assess new products during product evaluation make comparisons between products or versions of products set targets for future application developments" [3]. Despite the advantages, the average time required per participant for the 50-statement SUMI is higher as opposed the 10-statement SUS questionnaire. An experienced user of a software takes about 10 min, and a first-time user takes 20 min to more than an hour depending on the complexity of the software and task [5]. The commercially available SUMI analysis service is free for academic purposes.

SUS is a widely used survey developed in 1986 by John Brooke as a "quick and dirty" usability evaluation method [6]. The statements are alternately worded positively and negatively, and the users are asked to rate their agreements on a 5-point Likert scale. Benyon [1, p. 260] states that "There are several standard ways of measuring usability, but the best known and most robust is the system usability scale." SUS is widely applied for the evaluation of e-Learning platforms. A research including eleven studies on students' ( $n = 769$ ) perceived usability evaluation of learning management systems eClass and Moodle results a mean score 76.27 [7]. Furthermore, the analysis of the results also shows the validity and reliability of SUS for LMSs' evaluation, and that it remains robust even for small sample sizes [7]. For the evaluation of a co-active e-learning platform koaLA, the average SUS score for teachers was 72, learners was 56, and overall was 64 [8]. The Social Personalized Adaptive E-Learning Environment named Topolor got the SUS score 75.75, and Cronbach's alpha was 0.85 [9]. It was found that SUS ratings for learning management systems (with adaptive learning functionality) can provide a reliable evaluation decision even with a sample size of 6–14 participants [7]. Multiple studies estab-



lished that students' gender does not significantly affect their SUS score [7, 10]. During the COVID-19 pandemic, responses from Jordanian universities show that SUS score for Zoom was 67 ( $n = 21$ ) and Microsoft Teams was 58.9 ( $n = 54$ ) [11].

In this study, we selected SUS for reducing the time required for the students, established appropriateness in the case of low number of responses, gender independence, and being established method for the usability evaluation of e-learning systems.

## 2.2 *Qualitative Usability Evaluations*

While Benyon [1] used the generalization “participant-based evaluation”, Sharp et al. [12] use the heading “direct observations in controlled environments” for describing the qualitative and experimental usability evaluation techniques like think-aloud test and cooperative evaluation. For the same pair-wise think-aloud test protocol, where two participants who know each other participate in a think-aloud testing session accompanied with a concurrent or retrospective interview, the testing method is called constructive interaction [12, 13] or Co-discovery [1, p. 252], [14]. “Inevitably, asking specific questions skews the output toward the evaluator’s interests, but it does help to ensure that all important angles are covered” [1, p. 252]. Furthermore, studies proved that concurrent think-aloud protocols result in identifying more problems [15]. Thus, considering the time constraint and to build on the insights, an open-ended question is included in the SUS questionnaire and co-discovery was applied with concurrent interviews.

## 2.3 *Design Principles*

Don Norman, Jacob Nielsen, and others have developed many principles of good interactive system design. The design principles can guide the design, ideation, prototyping, and the interactive product evaluation processes. Benyon [1, p. 116–122] grouped the design principles into three main categories: (1) Learnability—Principles 1–4 are concerned with access, ease of learning and remembering, (2) Effectiveness—Principles 5–7 are concerned with ease of use, and principles 8 and 9 with safety, (3) Accommodation—Principles 10–12 are concerned with accommodating differences between people and respecting those differences. The 12 principles are: (1) visibility, (2) consistency, (3) familiarity, (4) affordance, (5) navigation, (6) control, (7) feedback, (8) recovery, (9) constraints, (10) flexibility, (11) style, and (12) conviviality. In this study, the 12 design principles are used as the conceptual framework to analyze the qualitative responses. Existing research lacks case studies on the application of interaction design and usability principles for evaluating and defining scopes for redesign.

### 3 Context

The University College Absalon (UCA), Area9—the proprietor of Rhapsode™ platform, Holbaek municipality, and the Technical University of Denmark (DTU) are collaboratively integrating the platform with the goal of improving the quality of the professional nursing education. The transformation process includes three iterations, and this usability study is part of the second iteration. In Fall 2021, students of a specific semester (not specified here for anonymizing) are divided into cohort A with Rhapsode™ as the learning resource and cohort B with the former practice of using the textbook.

The Area9 team and the educators of the professional nursing education program collaboratively defined measurable learning objectives, activities, appropriated and digitalized the textbook content, and created the online learning environment using the Rhapsode™ Curator platform. In Fall 2021, the cohort A students were expected to use the Rhapsode™ Learner platform for preparing themselves before four lectures. For prioritizing the emphasis on the different concepts covered in the lectures, the course instructors were expected to consider the analytics about the students' learning progression and engagement in the Rhapsode™ Educator platform.

The students already participated multiple interviews and surveys as part of the development project, and further studies would have put strain on the students. Therefore, the usability study had to be designed for taking minimum possible time from the students but find sufficient insights to make further development.

### 4 Methods

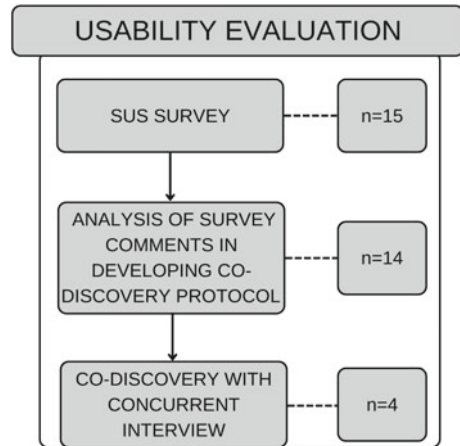
In this study, we conducted a usability evaluation of the Rhapsode™ Learner platform by applying SUS, including one open-ended question in the questionnaire, and the insights from the open-ended question's responses were used to define the tasks and interview questions for the co-discovery session with two pairs of students (see Fig. 1).

#### 4.1 Overview of Data Collection and Analysis Methods

The survey questionnaire in Microsoft forms was distributed to the students through the course instructor, and anonymous responses were collected. The procedure was as follows:

- Translating standard SUS questions to Danish and verifying the translation by the authors.
- Tabulating the response scores, testing reliability of using Cronbach's alpha, and calculating the SUS score.

**Fig. 1** Research progress and number of participants for each phase



- Analyzing the open-ended question's responses, based on which developing protocol for co-discovery.
- Conducting the co-discovery and analyzing the observations and concurrent interview responses using the interaction design principles.

## 4.2 SUS Questionnaire

We used Brooke's [6] standard scoring method for converting the individual statement ratings to a 0–4 value according to the alternating statement wording, which were then multiplied by 2.5 to give a score of 0–100 for each respondent. The overall SUS score for the system was then calculated as the summation of the individual scores.

The SUS questionnaire [6] was translated into Danish, and the translation procedure was as follows:

1. The questions were translated from English to Danish by one of the researchers.
2. The English (original) and Danish (translated) versions were compared to examine the accuracy of the translation and resolve any ambiguities in the wording of the translation. An experienced translator was consulted regarding the translation of sentences and concepts.
3. The Questionnaire in Danish was translated back to English using Google Translate, provided an opportunity to critically appraise differences in wording and meaning based on the chosen Danish phrasing. No deviation was found.
4. The Google-translated questionnaire was compared with a translated version in Danish, which was conducted by a co-author. Inconsistencies were discussed and addressed, and the last version of the translated questionnaire was obtained.

Only 15 of the 30 cohort A students (third semester) participated in the SUS survey. Due to the busy class schedule and absence of clear value for participation,

the students lacked motivation for participating, hence the low rate of participation (50%). Furthermore, the students were all female, and about all were in their twenties except one in her fifties.

### ***4.3 Co-discovery with Concurrent Interview***

The SUS responses were downloaded from Microsoft Forms and analyzed in Microsoft Excel. A table (not included here) was created to further analyze the data. Through thematic analysis [16], the qualitative data from the open-ended question in the SUS survey were used for the development of co-discovery tasks and concurrent interview questions. A table (not included) was made to map the questions and identified themes. To create the questions, we used the four themes developed from the thematic analyses of the qualitative responses to the question in the SUS survey. The themes were: (1) I would rather do something else, (2) Simple and Natural Dialog, (3) Feedback, and (4) Speak the user's language.

The application of concurrent interview during the co-discovery seeks to qualitatively evaluate Rhapsode™ Learner and identify the students' usability issues using the design principles as the theoretical framework. The goals of the concurrent interview are therefore:

- To identify the how and why aspects of the usability of Rhapsode™ learner and examine why Rhapsode™ Learner scored low marginally acceptable (51–68) SUS score.
- To gain new insights on users and their needs.

Due to the time constraints associated with the students' tight study schedule, we only had the option of recruiting four volunteering students in between two lectures. For the 45-min co-discovery with concurrent interviews, we had 11 tasks and questions on the use of Rhapsode™ for the students' preparation activity and 4 questions on how students navigated in Rhapsode™ Learner (table not included). The students attended in pairs, used the Rhapsode™ Learner, and responded to the questions. One of the authors facilitated the tasks and conducted the interviews while another author observed and took notes. Qualitative data from the co-discovery session were analyzed by using the design principles [1, pp. 116–122].

## **5 Results**

This section reports the overall SUS score, reliability of the responses, and the scope of improvement of the Rhapsode™ Learner adaptive learning system.

**Table 1** SUS score calculation table

| SUS question      | User's statement rating <sup>a</sup> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------------|--------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                   | 1                                    | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| Q1                | 3                                    | 2  | 2  | 2  | 3  | 4  | 4  | 3  | 1  | 4  | 4  | 4  | 5  | 1  | 3  |
| Q2                | 3                                    | 5  | 4  | 4  | 5  | 2  | 2  | 3  | 2  | 3  | 3  | 1  | 2  | 5  | 4  |
| Q3                | 4                                    | 2  | 3  | 3  | 1  | 4  | 3  | 4  | 5  | 4  | 4  | 3  | 4  | 1  | 4  |
| Q4                | 2                                    | 1  | 2  | 2  | 4  | 2  | 1  | 4  | 2  | 1  | 3  | 4  | 2  | 5  | 2  |
| Q5                | 4                                    | 3  | 2  | 3  | 2  | 3  | 4  | 3  | 4  | 4  | 3  | 4  | 4  | 2  | 4  |
| Q6                | 3                                    | 5  | 4  | 4  | 5  | 3  | 4  | 4  | 4  | 3  | 3  | 2  | 3  | 5  | 3  |
| Q7                | 4                                    | 5  | 4  | 2  | 3  | 4  | 5  | 3  | 3  | 4  | 4  | 3  | 4  | 3  | 4  |
| Q8                | 1                                    | 4  | 4  | 3  | 4  | 2  | 2  | 2  | 2  | 2  | 3  | 2  | 2  | 4  | 3  |
| Q9                | 4                                    | 4  | 3  | 2  | 1  | 2  | 3  | 2  | 2  | 4  | 3  | 4  | 4  | 1  | 4  |
| Q10               | 2                                    | 2  | 2  | 4  | 2  | 3  | 1  | 2  | 5  | 2  | 3  | 4  | 2  | 3  | 2  |
| Individual scores | 73                                   | 48 | 45 | 38 | 25 | 63 | 73 | 50 | 50 | 73 | 58 | 63 | 75 | 15 | 63 |
| Average           | 53.8                                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

<sup>a</sup> strongly agree = 5, strongly disagree = 1

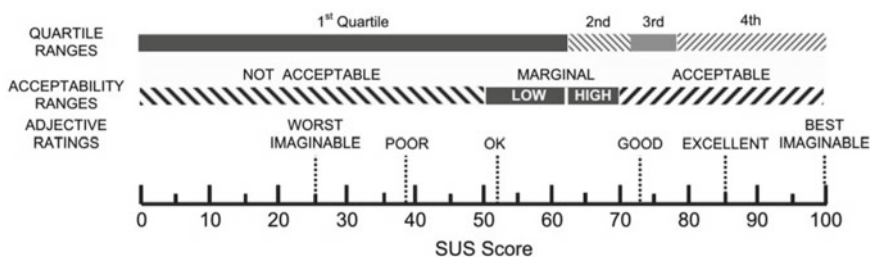


**Fig. 2** A comparison of mean system usability scale (SUS) scores by quartile, adjective ratings, and the acceptability of the overall SUS score [10, p. 592]

## 5.1 SUS Results

The scores in the system usability scale are shown in Table 1, and responses to the open-ended question are analyzed in a table (not presented). Color-coding in Table 1 is done to visualize the responses following the example of [17]. The most positive responses for each of the questions have been colored as green, neutral responses as yellow, and negative responses as orange.

Bangor et al. [10] present a scale for acceptable SUS scores (see Fig. 2) and reviewed that “it is easier to show that a product is unacceptable than it is to show that it is acceptable—if one wishes to make the claim with a high degree of confidence. This must be kept in mind when determining the acceptability of a product based on SUS scores, unless uncommonly large numbers of participants are tested.” These ranges will be used to evaluate the SUS results.

Two out of 15 respondents evaluated the system to be within the range of what is considered acceptable and 13 respondents assessed Rhapsode™ Learner as “not acceptable”. Evaluators 5 and 14 have extremely low scores of 25 and 15 out of 100, respectively. Examining tendencies between the respondents, they fall into two groups. The first pattern is shared by evaluators 2, 3, 4, 5, 8, 9, and 14. These evaluators gave the lowest score—the average score being 50 or below. The second pattern is shared by evaluators 1, 6, 7, 10, 13, and 15 agree on most questions. Evaluators 11 and 12 diverge from these two patterns.

The numbers of orange and green scores are fairly even and dominating the one-fourth yellow (neutral) answers. The overall scores for respondents, on a scale from 0 to 100, range from as low as 15, while only four total scores (73 and 75) lie within the thresholds for an acceptable score, 65–84 [17]. The average score obtained is 53.8, which is below an acceptable average of 68.

From the qualitative responses, four themes were identified through thematic analysis where responses from evaluators fall into more than one theme. The themes were categorized based on the design principles reported by Benyon [1]. Evaluator no. 1 was the only exception, stating “Nothing to add”.

**I would rather do something else:** Evaluators 2, 6, 9, and 12 are not critiquing Rhapsode™ on its own terms. The statements indicate the effect of temporal elements on the experience and evaluation of digital systems in general.

**Simple and Natural Dialog:** Students’ (5, 7, 8, and 10) comments fell into the design principle of simple and natural dialog as presenting the necessary information when and where it is needed according to users’ needs. The theme was used to create the questions on students’ preparation with Rhapsode™ and on how students navigated.

**Feedback:** To follow the feedback design principle, a system should continuously inform users on system status and how user input is interpreted. Ideally, feedback should arrive preemptively before error occurs. Students’ (3, 7, 8, and 11) comments fell into this design principle and were used to create questions (partially overlapping previous set) on students’ preparation with Rhapsode™ and on how students navigated.

**Speak the user’s language:** This theme leans on the design principle of speaking the user’s language. The students (4, 11, 13, 14, and 15) comments were related to both the nursing profession and the learning content-complexity level. Comments were also on the didactic of how the material is structured. Interview questions prepared based on the comments partially overlapped the previous sets.

## 5.2 SUS Validity and Reliability

Many studies on software testing have shown that SUS is highly reliable. Since reliability is widely measured using Cronbach’s alpha, the method was applied to investigate the internal consistency of SUS. Zach [18] summarized that reliability is

acceptable if  $0.7 \leq \alpha < 0.8$  and good if  $0.8 \leq \alpha < 0.9$ . The results show that SUS responses are reliable with good internal consistency (10 items,  $\alpha = 0.863$ ).

### 5.3 Co-discovery Results

Analysis of the observations and interview notes from the co-discovery session was conducted based on the analysis of the open-ended question's responses and are grouped under the following usability principles.

**Inconsistency:** Design should “be consistent in the use of language and design features and be consistent with similar systems and standard ways of working” [1, p. 117]. The readout function does not always resume at the same place where it was interrupted (E10). In addition, wrong answers to questions about the module can result in the user getting caught in an infinite sequence of texts and questions. Furthermore, if the module is not completed upon time, the users' progress and position in the module will be lost. None has shown any action demonstrating that any of these issues are caused by undesired use.

**Unclear Affordances:** “Design things so it is clear what they are for—for example, make buttons look like push buttons so people will press them. [...] Affordances are culturally determined” [1, p. 117]. One of the issues evaluators brought up was the limitations of the search function. It only searches the current module and not across all modules. This is also seen in the SUS comments where the term encyclopedia was brought up several times.

**Familiarity:** Designers should “[u]se language and symbols that the intended audience will be familiar with” [1, p.117]. During co-discovery, the meaning of features such as “Gnist” points were cluttering up the GUI experience of the students.

**Navigation and Visibility:** Navigation should “[p]rovide support to enable people to move around the parts of the system: maps, directional signs and information signs” [1, p. 117]. For visibility, “Try to ensure that things are visible so that people can see what functions are available and what the system is currently doing” [1, p. 117]. The co-discovery participants stated that the table of content is not easy for the students to find in the modules. Furthermore, the GUI was cluttered and made navigation difficult.

## 6 Conclusion and Scope of Future Work

As part of experimental research toward adopting the adaptive learning system Rhapsode<sup>TM</sup> Learner, the learners' satisfaction was measured by applying SUS due to its reliability and to reduce the time required to conduct empirical evaluation. While the average score of 53.8 ( $n = 15$ ) shows low marginal acceptability with good internal consistency (10 items,  $\alpha = 0.863$ ), the scope of design improvement cannot be determined. Thus, the adapted SUS questionnaire with an additional open-ended

question resulted in defining the tasks and questions for a subsequent co-discovery session with two pairs of students. The accompanying concurrent interview and observation of the co-discovery session data were analyzed by using the 12 interaction design principles [1]. SUS scores of the studies on digital learning technologies scored higher than Rhapsode™ Learner and the need for redesign has been justified. We identified that the design principles, namely consistency, affordance, familiarity, navigation, and visibility should be taken into further consideration for improving the design. For the evaluation of learning experience, the design principles focused on technology only and not sufficient for analyzing the evaluation of the pedagogy and content dimensions [19]. For further evaluation of the pedagogical and content dimensions of the adaptive learning platform's user experience, questionnaires should be developed based on the frameworks from the field of information systems, information quality, and pedagogy. Moreover, the effect of temporal elements in the user experience was identified in the responses to the open-ended question, and those elements might have affected the SUS score. The benefit of one open-ended question as part of SUS is warranted, and inclusion of a question investigating the effect of temporal elements might be appended in the questionnaire but relevant for calculating the overall system evaluation score.

The flexibility of the adaptive learning platform bring along an increased need for evaluating each of the course or module separately and collectively. We invite academics to develop broader learner experience design frameworks that are not limited to the interaction design of digital products but the broader user or learner experience. Furthermore, the platforms for content curation and teachers' use, in our case of Rhapsode™ Curator and Rhapsode™ Educator, should be evaluated sufficiently to establish acceptability. Considering the importance of analytics and the increasing opportunities for artificial intelligence methods in providing "nudging" with insights to individual learners and educators, further evaluation studies are required. Finally, we reemphasize the importance of taking as little time as possible from the students and educators while conducting empirical evaluations of digital learning technologies.

**Acknowledgements** Supported by NURSEED Project, which is funded by Innovation Fund Denmark (IFD).

## References

1. Benyon D (2019) Designing user experience. Pearson, UK
2. Shackel B (1990) Human factors and usability. In: Human-computer interaction. Prentice Hall Press, USA, pp 27–41. ISBN: 978-0-13-444910-4 (Visited on 01/27/2022)
3. Kirakowski J. SUMI questionnaire homepage. The official home page for the online SUMI questionnaire. <https://sumi.uxp.ie/> (visited on 01/27/2022)
4. Kirakowski J (1996) The software usability measurement inventory: background and usage. In: Jordan PW et al (eds) Usability evaluation in industry. CRC Press, Boca Raton, pp 169–178. ISBN: 978-0-7484-0460-5



5. Arh T, Blažič BJ (2008) A case study of usability testing: the SUMI evaluation approach of the EducaNext portal. *WSEAS Trans Inf Sci Appl* 5(2):175–181. ISSN: 1790-0832
6. Brooke J (1996) SUS: a “quick and dirty” usability scale. In: Jordan PW et al (eds) *Usability evaluation in industry*. CRC Press, Boca Raton, pp 189–194. ISBN: 978-0-7484-0460-5
7. Orfanou K, Tselios N, Katsanos C (2015) Perceived usability evaluation of learning management systems: empirical evaluation of the system usability scale. *Int Rev Res Open Distance Learn* 16(2):227–246. ISSN: 1492-3831. <https://doi.org/10.19173/irrodl.v16i2>
8. Blecken A, Bruggemann D, Marx W (2010) Usability evaluation of a learning management system. In: 2010 43rd Hawaii international conference on system sciences, pp 1–9. ISSN: 1530-1605. <https://doi.org/10.1109/HICSS.2010.422>
9. Shi L, Awan MSK, Cristea AI (2013) Evaluating system functionality in social personalized adaptive E-learning systems. In: Hernández-Leo D et al (eds) *Scaling up learning for sustained impact. Lecture notes in computer science*. Springer, Berlin, pp 633–634. ISBN: 978-3-642-40814-4. [https://doi.org/10.1007/978-3-642-40814-4\\_87](https://doi.org/10.1007/978-3-642-40814-4_87)
10. Bangor A, Kortum PT, Miller JT (2008) An empirical evaluation of the system usability scale. *Int J Human-Comput Interact* 24(6):574–594. ISSN: 1044-7318. <https://doi.org/10.1080/10447310802205776>
11. Abushamleh H, Jusoh S (2021) Usability evaluation of distance education tools used in Jordanian universities. In: 2021 innovation and new trends in engineering, science and technology education conference (IETSEC), May 2021, pp 1–5. <https://doi.org/10.1109/IETSEC51476.2021.9440491>
12. Sharp H, Preece J, Rogers Y (2019) *Interaction design: beyond human-computer interaction*. Wiley. ISBN: 978-1-119-54730-3
13. (Hans) Kemp JAM, van Gelderen T (1996) Co-discovery exploration: an informal method for the iterative design of consumer products. In: *Usability evaluation in industry*. CRC Press, Boca Raton, 8 p. ISBN: 978-0-429-15701-1
14. Miyake N (1986) Constructive interaction and the iterative process of understanding. *Cogn Sci* 10(2):151–177. ISSN: 0364-0213. [https://doi.org/10.1016/S0364-0213\(86\)80002-7](https://doi.org/10.1016/S0364-0213(86)80002-7). <https://www.sciencedirect.com/science/article/pii/S0364021386800027> (visited on 01/28/2022)
15. van den Haak M, De Jong M, Schellens PJ (2003) Retrospective vs. concurrent think-aloud protocols: testing the usability of an online library catalogue. *Behav Inf Technol* 22(5):339–351. ISSN: 0144-929X, 1362-3001. [https://doi.org/10.1080/0044929031000110001100011000110001](https://doi.org/10.1080/00449290310001100011000110001). <http://www.tandfonline.com> (visited on 05/08/2015)
16. Braun V, Clarke V (2006) Using thematic analysis in psychology. *Qual Res Psychol* 3(2):77–101. ISSN: 1478-0887. <https://doi.org/10.1191/1478088706qp0630a>. <http://www.tandfonline.com> (visited on 04/09/2015)
17. Kulkarni R et al (2013) Usability evaluation of PS using SUMI (Software usability measurement inventory). In: 2013 international conference on advances in computing, communications and informatics (ICACCI), pp 1270–1273. <https://doi.org/10.1109/ICACCI>
18. Zach (2021) How to calculate Cronbach’s alpha in R (with examples). *Statology*. <https://www.statology.org/cronbachs-alpha-in-r/> (visited on 04/06/2022)
19. Koh JHL, Chai CS, Tay LY (2014) TPACK-in-action: unpacking the contextual influences of teachers’ construction of technological pedagogical content knowledge (TPACK). *Comput Educ* 78:20–29. ISSN: 0360-1315. <https://doi.org/10.1016/j.compedu.2014.04.022>. <https://www.sciencedirect.com/science/article/pii/S0360131514001134> (visited on 02/05/2022)

## Chapter 7

# A Machine Learning and Deep Learning Approach to Classify Mental Illness with the Collaboration of Natural Language Processing



**Md. Rafin Khan, Shadman Sakib, Adria Binte Habib,  
and Muhammad Iqbal Hossain**

**Abstract** The most alarming yet abstained issue of our so-called ‘Generation Z’ is mental health. In many developing countries, it is unfortunately treated as a mere joke by a majority of the population. The only way to tackle this is to find out the correct mental illness associated with an individual and provide a systematic solution as early as possible. In this paper, the authors emphasized the category of a disease rather than just generalizing it as depression. Four highly anticipated mental health statuses were selected which were Schizophrenia, PTSD, Bipolar Disorder, and Depression. This research proposes to identify which of these mental illnesses a person is most likely to be diagnosed with. It has been done by multiple classification algorithms and the language patterns of such self-reported diagnosed people from a corpus of Reddit posts to discover better outcome.

**Keywords** Schizophrenia · PTSD · Bipolar Disorder · Depression · Tokenization · Lemmatization · TF-IDF · Count-vectorizer

## 1 Introduction

In many countries, mental illnesses are thought of as imaginary things, and most families are ashamed if one of their family members gets affected with mental illness, so we were thinking what if there was a more discreet way for people to diagnose themselves without the fear of social stigma. Despite the fact that there are billions of individuals, there are just a few clinical psychiatrists [1]. The article also mentions that it can be difficult to know who to turn to for help when there is a scarcity of facilities, hospitals, or psychiatric services. According to the authors, there are limited laboratory tests for diagnosing most forms of mental illness, and the major source of diagnosis is the patient’s self-reported experience or behaviors recorded by

---

Md. R. Khan · S. Sakib · A. B. Habib · M. I. Hossain (✉)  
Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh  
e-mail: [iqbal.hossain@bracu.ac.bd](mailto:iqbal.hossain@bracu.ac.bd)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_7](https://doi.org/10.1007/978-981-19-7528-8_7)

family or friends [2]. This is where the strength of advanced algorithms of Machine Learning comes into play which will forecast what type of mental illness the person has so that they can diagnose themselves from home.

As we all know that mental health has been one of the noteworthy issues in healthcare and plays a vital impact on one's quality of life, therefore we must find a way to quickly detect and diagnose it. In most cases, it is very difficult to express one's true emotions in front of family and friends, hence individuals tend to express themselves on social media platforms hoping to engage with other fellow victims for compassion and/or methods to ease the suffering or even just share their experiences. The authors stated that the ability to discuss mental health issues anonymously on the Internet motivates people to reveal personal information and seek help [3]. As a result, social media such as Reddit, Twitter and many others become a significant resource for mental health researchers in studying mental health. Although data from platforms on the internet such as Twitter, Facebook or Reddit is readily available, labeled data to study mental illness is confined [4]. Due to scarce information, it has been difficult to understand and address the different challenges in this domain, hence information retrieved from social networks not only provides medical assistance to the users in need but also expands our awareness of the common conditions of mental disorders.

The goal of this study is to determine whether or not a person has PTSD, Bipolar Disorder, Schizophrenia, or Depression with the help of Natural Language Processing and Machine learning algorithms from the corpus of any social media post. Prior work [5] has inspired us to improve existing models and test new ones by working on the SMHD dataset which was collected from Georgetown University, in order to improve prediction accuracy and overall precision, recall, and  $f1$  performance. We have used a large-scale Reddit dataset, collected via an agreement, to conduct our research. Reddit is an open-source forum where members of the community (redditors) can submit content (such as articles, comments, or direct links), vote on submissions, and organize content by topics of interest (subreddits). The following is a breakdown of the paper's structure: First, an explanation is provided on dataset collection and preprocessing techniques. After that, various classification algorithms are implemented. Then the accuracy, precision, recall, and  $f1$  score are measured. Finally, concluded with a discussion of plausible future works in this area.

## 2 Related Work

In the recent era of 5G, social media has a drastic impact on our everyday lives. It is a platform for human beings to keep in touch or update their daily lives through posts, pictures, opinions, etc. However, who knew that the opinions and thoughts of the social media users would amount to such valuable research study. Previous works of researchers who used twitter posts as datasets to identify depression and other mental disorders have left us valuable findings that we can use and enhance our results. In 2013, the authors had suggested a strategy for detecting depressed users

from Twitter posts where they used crowd sourcing to compile the Twitter users [2]. They measured data such as, user engagement and mood, egocentric social graphs and linguistic style, depressive language use and antidepressant medication mentions on social media. They then contrasted the activities of depressed and non-depressed users, revealing indicators of depression such as decreased social activity, increased negative feeling, high self-attentional focus, increased relationship and medical concerns and heightened expression of religious views. They used multiple sorts of signals to create an MDD classifier that can predict if an individual is sensitive to depression ahead of the beginning of MDD. As time proceeded, the works of identifying the type of mental health conditions were emphasized more rather than carrying out surveys. For instance, the authors have analyzed four types of mental illness (Bipolar, Depression, PTSD, SAD) from 1200 twitter users using Natural Language Processing. The diagnosis statement from their tweets such as 'I was diagnosed with depression' was conducted through an LIWC tool and the deviations were measured from a control group against those four groups [6]. Then, they used an open vocabulary analysis to collect language use that is related to mental health in addition to what LIWC catches. Subsequently in 2014, the researchers conducted an elaborated work to classify PTSD users from twitter and found out an increasing rate among them, especially targeted among US military soldiers returning from prolonged wars [7]. Similarly, the LIWC tool was used to investigate the PTSD narrations used by these self-reported users for language comparison. Other than that LIWC was also used in finding linguistic patterns between users to classify ten mental health conditions from Twitter posts [4]. Nonetheless the paper [8] analyses tweets to figure out symptoms of depression like negative mood, sleep disturbances, and energy loss. The paper [9] shows further study investigated on the mental health-related social media text categorization generated from Reddit. Although studies to date witnessed CNN to be a better model in terms of performance, the authors implemented hierarchical RNN architecture to address the problem. Because of the fact that computational costs are higher and removal of unnecessary contents makes the model faster, they also employed attention mechanisms to determine which portions of a text contribute the most to text categorization. Even though twitter posts are a significant source of data for language usage, long forums and contents are also pivotal for a valid dataset. In this case, Reddit users are applied for building a corpus which gives newer insights to linguistics. The paper [5] mentions that unlike Twitter, which has a post limitation in terms of word count, Reddit platform has no such constraints. Also this dataset they have used, contains posts of diverse mental health condition patients along with mentally healthy Reddit users also known as control users. Their data was compiled with the use of high-precision diagnosis patterns. The filtering of control users in the dataset was rigid. For example, any Reddit user who never had any post related to mental health as well as having no more than 50 posts were not included. The paper also mentioned that control users tend to post twice as much as any diagnosed user and their posts are comparably shorter. They looked at how different linguistic and psychological signs indicated disparities in language usage between those with mental illnesses (diagnosed users) and others who were not (control users). To identify the diagnosed users, several text categorization algorithms were tested, with

FastText proving to be the most effective overall. In 2016, the authors used Reddit posts and comments and paired 150 depressed users with 750 control users to find out the language distinction between users who are depressed and those who are not [10]. They have outlined the methods they used to generate a test collection of textual encounters made by depressed and non-depressed persons. The new collection will help researchers look into not only the differences in language between depressed and non-depressed persons, but also the evolution of depressed users' language use. The authors [3] applied self-reported diagnoses to a broader set of Reddit users, yielding the Reddit Self-reported Depression Diagnosis (RSDD) dataset, which had over 9000 users with depression and over 100,000 control users (using an improved user control identification technique). Posts in the RSDD dataset were annotated to check that they contained assertions of a diagnosis. Similar kind of task was done by one of the authors in their paper [4], who were also able to authenticate self-reports by deleting jokes, quotes and false statements.

### 3 Data Analysis

#### 3.1 Dataset Overview

The Self-reported Mental Health Diagnoses (SMHD) dataset is collected from Georgetown University [5]. SMHD had conditions corresponding to branches in [11] a total of 9 conditions and only 4 from these are listed in Table 1. These are Schizophrenia, Depression and Bipolar are top-level DSM-5 disorders whereas PTSD is one level lower (Fig. 1).

#### 3.2 Dataset Formation

The data is in the form of json lines (.jl) format, which basically means that each line of the files were in json format. Each json line of the files represented one user, it contained an id, the label of the user's mental health condition and all the posts that the user wrote with the time and date when each comment was posted. Since the dataset is enormous; approximately 50GB, we loaded it using the ijson library of Python, which loads and parses files using iterators to load data lazily. As a result, if an unnecessary key passes by it can be simply ignored and the created object will be removed from memory. This helped avoid exceeding memory usage constraints set by Google Colab runtime. A data frame was created where each row consists of only the labels and the posts all concatenated per user. Another effective measure was taken to optimize memory usage was to pre-process the data, which has a more in-depth discussion in Sect. 4.1, before concatenating the posts iteratively. Lastly, the Pandas Dataframe was converted to a csv file to access it at ease.

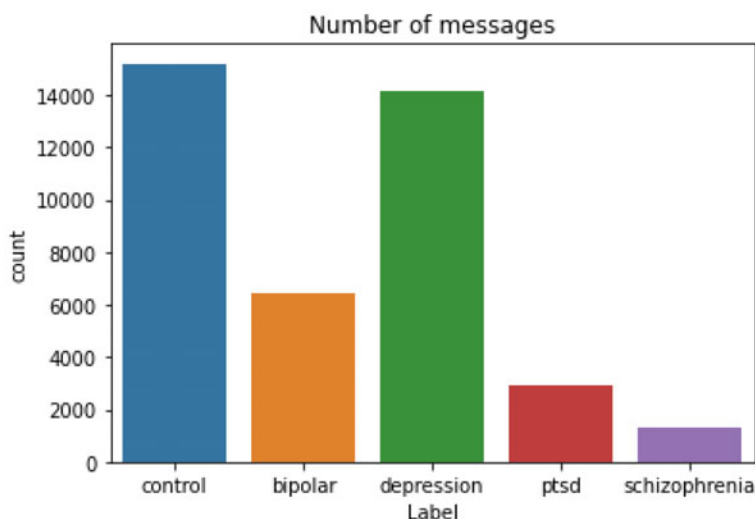


**Fig. 1** Visual representation of number of messages by the self-reported diagnosed users per condition

## 4 Methodology

### 4.1 Data Preprocessing

In order to apply machine learning algorithms in the text data, the data must be clean. In other words, algorithms perform better with numbers rather than text [12]. Thus the dataset was loaded into a pandas dataframe and then ‘feature engineered’. To bring all the values of different numerical range into a standard region standardization, log normalization and feature scaling was used. To make the dataset structured, the data was filtered by removing dirty data such as missing row values, NaN type or mixed data such as emoticons. At first, punctuation marks and stopwords were removed, i.e., connecting words ‘i’, ‘we’, ‘me’, ‘myself’, ‘you’, ‘you’re’ which form a meaningful sentence, from the text since it does not add any value to classification models. Besides, lemmatization was also used which reduces the inflected words to its root word [13]. The reason behind doing lemmatization instead of stemming is because lemmatization always reduces to a dictionary word although it is computationally expensive. Despite the fact that stemming is faster than lemmatization, it simply chops off the end of a word using heuristics while lemmatization uses more informed analysis. Previous works also suggest that truncation of post length improves classification performance [5]. For dealing with text data, each of the words need to give unique numbers since machine learning models can not be trained on text data. The fit and transform method of the TF-IDF class was used from the Sci-Kit learn library. TF-IDF is a simple tool for tokenizing texts and creating a vocabulary of

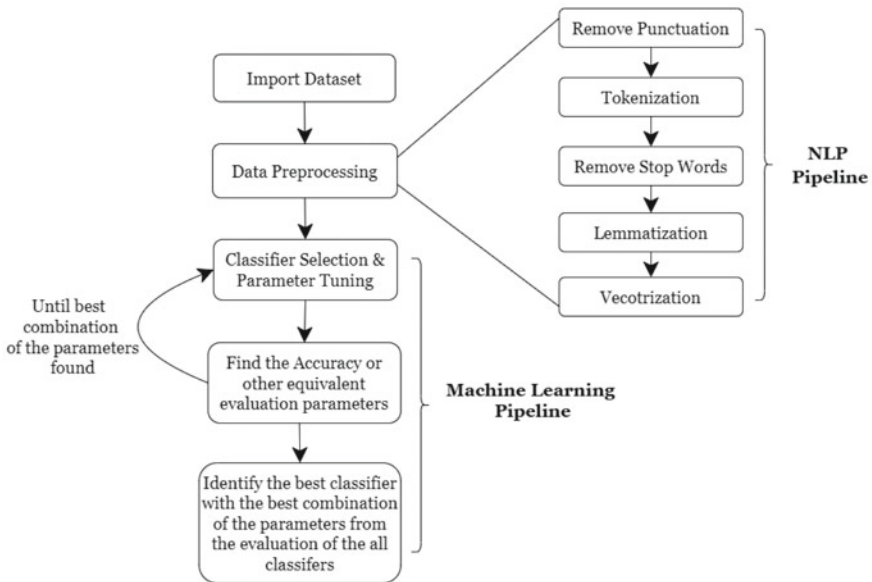


**Fig. 2** Proposed pipeline

known terms, as well as encoding new texts with that vocabulary [14]. Tokenization basically refers to splitting up the raw text into a list of words which helps in the comprehension of the meaning or the creation of the NLP model and Python does not know which word is more important, hence we need further preprocessing. TF-IDF creates a document term matrix where columns are individual unique words and the cells contain a weight which signifies how important a word is for an individual text which means if the data is unbalanced, it will not be taken into consideration. In other terms, words that appear often in the text, for example ‘what’, ‘is’, ‘if’, ‘the’, are scored low since they will have little meaning in that particular document. However, if a word appears frequently in a document but not in others, then it is likely to be relevant. By this way, the TF-IDF algorithm sorts the data into categories which assist our proposed models to work faster and bring outstanding results.

$$W_{i,j} = TF_{i,j} \log(N/DF_i) \quad (1)$$

The dataset was divided into train (70%) and test (30%) groups to make the classification model more precise, then investigated model construction and employed some machine learning algorithms, optimized it, and evaluated the performance of each model. To make our workflow more streamlined, a pipeline was used which allowed us to sequentially apply a list of transformations. This allowed us to implement a few classifiers in a short amount of time.

4.2 Proposed Pipeline

After the data were processed, it is time to fit it into the appropriate estimator. For various data types, estimators perform differently, therefore picking the proper estimator might be tricky. The followed approach was to classify all mental health conditions against our control group. Essentially a one-to-one approach. The reason it was done due to the text data between the mental health conditions being quite similar; as it seems it should be. A person that is diagnosed with PTSD is highly probable to be suffering from depression as well. Thus we want to classify between a person either being mentally ill or healthy. Following much investigation, we have arrived at the conclusion that the models listed below should be used in our work (Fig. 2).

5 Experimentation

5.1 Support Vector Machine

In order to select the best hyper-parameter combination GridSearchCV was used. It is the process of determining the ideal settings for a model’s hyper-parameters. The value of hyper-parameters has a substantial impact on a model’s performance. It’s worth noting that there’s no way to know ahead of time what the best values for hyper-parameters are, therefore we should try all of them to find the best ones, because manually adjusting hyper-parameters would take a significant amount of time and resources. Though it is computationally expensive, we decided it would be worth it for better results. The parameters that we tuned were  $C$ : cost parameter to all points that violate the constraints, gamma: defines how far the influence of a single training example reaches, and the Kernel. Having a low value of  $C$  creates a smooth hyperplane surface whereas a high value tries to fit all training examples correctly at the cost of a complex surface. Having run the GridSearchCV the best parameters were found. It is given in Table 1.

Table 1 Optimized parameters

| Algorithms          | Parameters                                                                                                                       |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------|
| SVM                 | $C = 80$ , Gamma = 0.01, Kernel = rbf                                                                                            |
| Logistic regression | Algorithm = newton cg, top $k = 3$ , median rate ratio = 0.8959                                                                  |
| GRU                 | Hidden layer = 100, activation function = ReLu, dropout = 0.3, loss function = binary crossentropy, optimizer = adam, Epoch = 10 |
| BERT                | Batch size = 16, Epoch = 5                                                                                                       |



## 5.2 Logistic Regression

Logistic regression is not only simple and easy to model, it is also very efficient which is extremely useful in our case due to our data being very large. Even though our dataset is fairly big it is very simple consisting of only two columns (Label and Text), therefore logistic regression is perfect since it produces better accuracy for simpler data. Like any other model logistic regression can also overfit but chances of overfitting is low, still to avoid this the authors have implemented the l2 regularization within the model which basically operates as a force that removes a little portion of the weights in each iteration causing the weights to never reach zero. We implemented newton-cg algorithm for optimization and used a max iteration of 2000 which is basically the number of iterations taken for the optimizer to converge (Fig. 3).

## 5.3 Gated Recurrent Unit (GRU)

For the model a GRU layer has been utilized consisting of 100 units with activation of 'relu' and dropout of 0.3 followed by a dense layer of 1000 units with activation 'relu' and dropout 0.7, this is repeated twice followed by an output dense layer of unit 1 with activation 'sigmoid'. the max token length has been taken to be 600 and padded any text length that is less than the given max token length. To minimize overfitting, dropouts were introduced, and the loss function was binary cross entropy using the 'adam' optimizer to optimize weights and learning rate, which helped reduce loss. Finally, the model was trained for 10 epochs for each sickness. Since

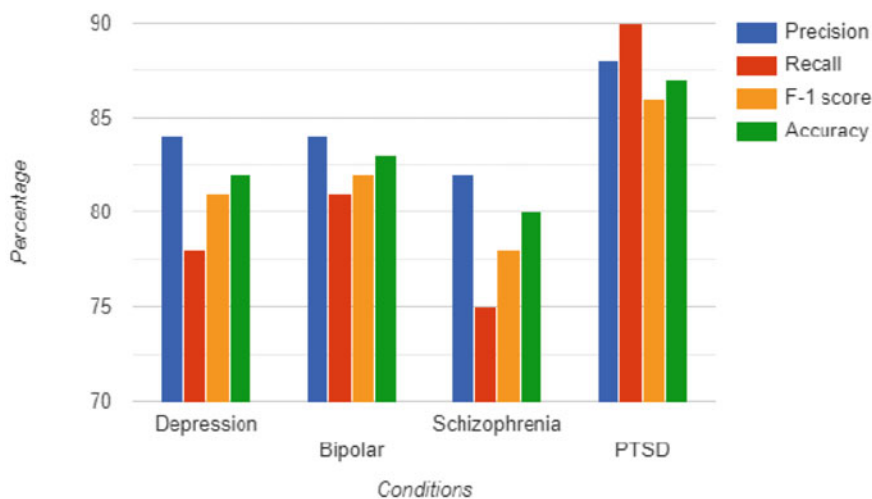


**Fig. 3** Performance analysis of SVM

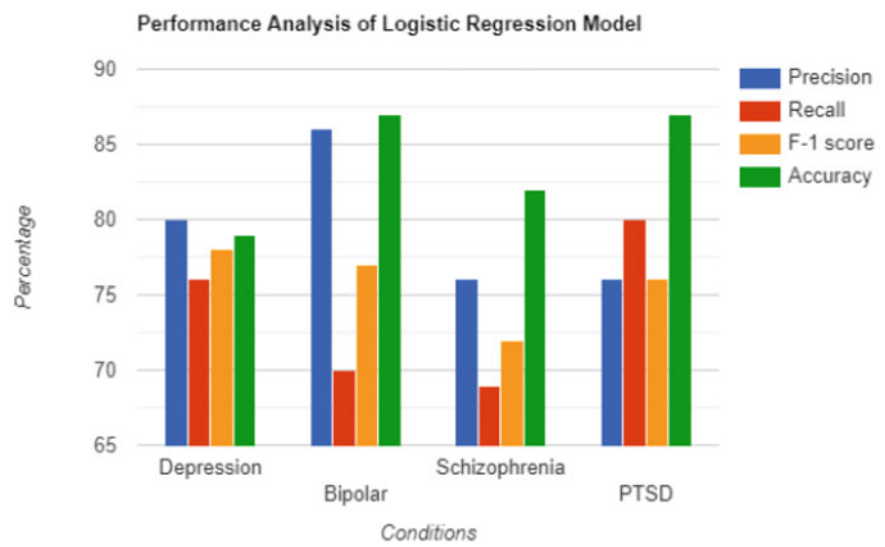


Fig. 4 Performance analysis of logistic regression

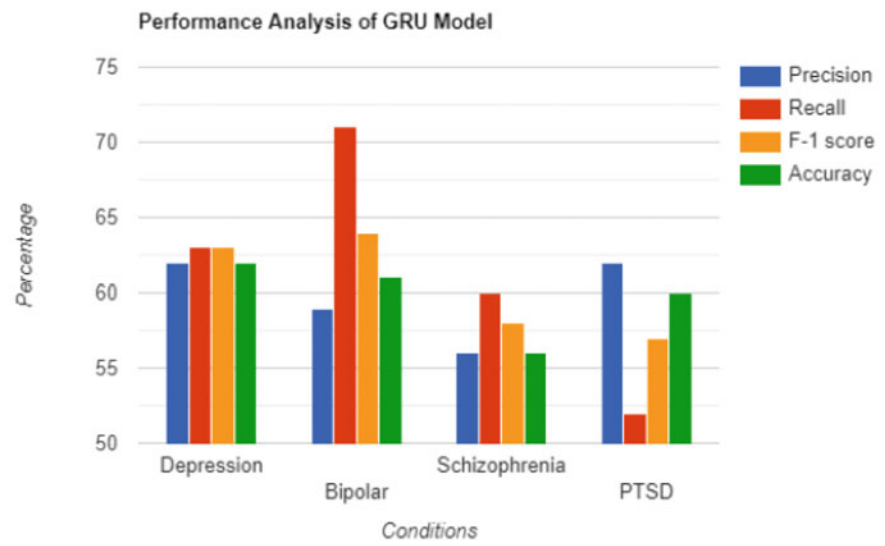


Fig. 5 Performance analysis of GRU

ours is binary classification, sigmoid is the best for output layer activation since it gives a result between 0 and 1 which can be inferred as how confident the model is in an example being in a particular class. And binary cross entropy basically compares two probability distributions and calculates the difference between them which is perfect for our binary classification.

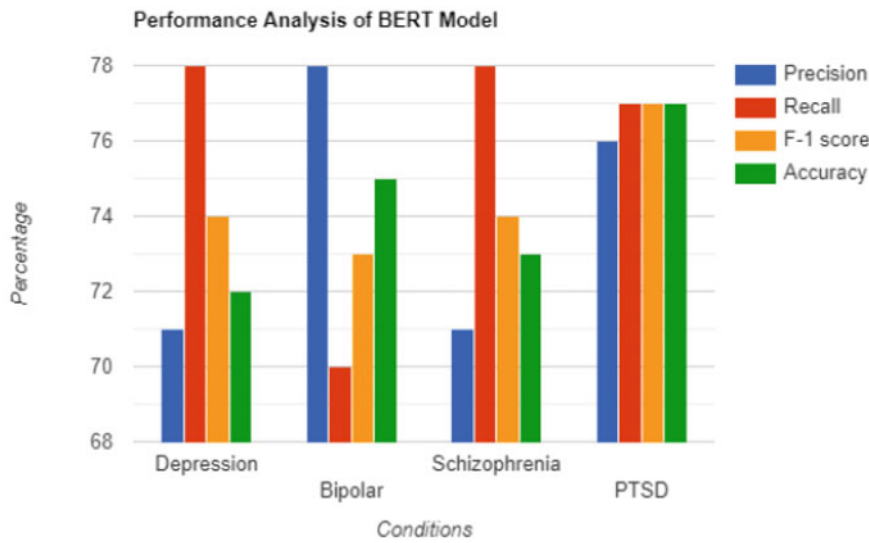


**Fig. 6** Performance analysis of BERT

**5.4 Bidirectional Encoder Representations from Transformers (BERT)**

To make the model even more lightweight ktrain was used. Ktrain is a keras library that aids in the creation, training, debugging, and deployment of neural networks. Ktrain allowed to easily employ the distilBERT pre-trained model and estimate an optimal learning rate. To utilize Ktrain, the ‘get learner’ function was used to wrap our data and model, in this case ‘distilbert-base-uncased,’ inside a ktrain learner object. A batch size of 16 was used for faster performance. The Learner object allows training in various ways. One of the most crucial hyper-parameters to configure in a neural network is the learning rate. Default learning rates for various optimizers, such as Adam and SGD, may not always be appropriate for a given problem. The training of a neural network requires minimizing a loss function. If the learning rate is too low, training will be postponed or halted. If the learning rate is too high, the loss will not be reduced. Both of these conditions are detrimental to the performance of a model. The author says that when graphing the learning rate versus the loss, the greatest learning rate associated with a dropping loss is a preferable choice for training. Thus a learning rate of  $1 \times 10^4$  was used inferring from the graph shown in Fig. 4. The model was trained on a maximum of 5 epochs and a minimum of 3 for the larger datasets (Figs. 5 and 6).

**Table 2** Result analysis of classification models

| Algorithms                           | Depression                        | Bipolar                           | Schizophrenia                     | PTSD                              |
|--------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| SVM and bow features                 | $P = 84, R = 78, F1 = 81, A = 82$ | $P = 84, R = 78, F1 = 81, A = 82$ | $P = 84, R = 78, F1 = 81, A = 82$ | $P = 84, R = 78, F1 = 81, A = 82$ |
| Logistic regression and bow features | $P = 80, R = 76, F1 = 78, A = 79$ | $P = 86, R = 70, F1 = 77, A = 87$ | $P = 76, R = 69, F1 = 72, A = 82$ | $P = 76, R = 80, F1 = 76, A = 87$ |
| GRU                                  | $P = 62, R = 63, F1 = 63, A = 62$ | $P = 59, R = 71, F1 = 64, A = 61$ | $P = 56, R = 60, F1 = 58, A = 56$ | $P = 62, R = 52, F1 = 57, A = 60$ |
| BERT                                 | $P = 71, R = 78, F1 = 74, A = 72$ | $P = 78, R = 70, F1 = 73, A = 75$ | $P = 71, R = 78, F1 = 74, A = 73$ | $P = 76, R = 77, F1 = 77, A = 77$ |

## 6 Result Analysis

The results from the classification models can be found in Table 2. As shown in the table, all of our models produced overall balanced outcomes, with SVM producing the greatest overall values while BERT and GRU models had higher recall than precision in the majority of the illnesses. As discussed earlier SVM performs remarkably well in high dimensional data, whereas GRU and BERT had word embedding limitations due to scarce GPU capability. 512 tokens were feeded for BERT and 600 tokens for GRU and Logistic Regression to reduce computational cost, whereas 4000 max weighted features were fed into SVM. This enabled the model to learn more types of words used by the patients. Even so SVM performs a lot faster than any neural network and is capable of predicting results faster. Also risk of overfitting is less in SVM over Logistic Regression. A very important Key Performance Indicator was noticed to be “Recall” other than just Precision. Precision is basically the number of times the model was correct when the classifier predicted the “True” class whereas recall is the number times the classifier got it correct when the class was actually “True” in short higher recall value means lower type II error, which is why the authors were focusing more on recall than precision since someone who has the illness but is misdiagnosed as negative will be in more danger of the illness progressing than someone who does not have the illness but is misdiagnosed as positive.

## 7 Conclusion

The relentless advancement of Machine Learning over the years is truly astonishing. This work is dedicated to detect mental illness with the assistance of machine learning and deep learning models. The performance analysis of the models showed that the models are performed really good. Among all the models, the model built with BERT algorithm performed well for all the diseases. An overall balanced Key Performance Indicator (KPI) is achieved among all the models that have used but looking forward

to improving those further. This research is limited due to our lack of access to high-end gear. Further work needs to do with larger versions of the proposed models like BERT LARGE and LSTM with embedding techniques.

## References

1. Tackling mental health stigma in Bangladesh. ADD International (n.d.) Retrieved 23 Mar 2022. from <https://add.org.uk/tackling-mental-health-stigma-bangladesh>
2. De Choudhury M, Gamon M, Counts S, Horvitz E (2013) Predicting depression via social media. In: Proceedings of the international AAAI conference on web and social media, vol 7(No 1)
3. Yates A, Cohan A, Goharian N (2017) Depression and self-harm risk assessment in online forums. arXiv preprint [arXiv:1709.01848](https://arxiv.org/abs/1709.01848)
4. Coppersmith G, Dredze M, Harman C, Hollingshead K (2015) From ADHD to SAD: analyzing the language of mental health in Twitter through self-reported diagnoses. In: Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality
5. Cohan A, Desmet B, Yates A, Soldaini L, MacAvaney S, Goharian N (2018) SMHD: a large-scale resource for exploring online language usage for multiple mental health conditions. arXiv preprint [arXiv:1806.05258](https://arxiv.org/abs/1806.05258)
6. Coppersmith G, Dredze M, Harman C (2014) Quantifying mental health signals on Twitter. In: Proceedings of the workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality
7. Coppersmith G, Harman C, Dredze M (2014) Measuring posttraumatic stress disorder in Twitter. In: Proceedings of the international AAAI conference on web and social media
8. Mowery DL, Park YA, Bryan C, Conway M (2016) Towards automatically classifying depressive symptoms from Twitter data for population health. In: Proceedings of the workshop on computational modeling of people's opinions, personality, and emotions in socialmedia (PEOPLES)
9. Ive J, Gkotsis G, Dutta R, Stewart R, Velupillai S (2018) Hierarchical neural model with attention mechanisms for the classification of social media text related to mental health. In: Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic, pp 69–77
10. Losada DE, Crestani F (2016) A test collection for research on depression and language use. In: International conference of the cross-language evaluation forum for European languages, Springer, Cham, pp 28–39
11. Diagnostic and statistical manual of mental disorders (DSM-5-TR). DSM-5 (n.d.) Retrieved 6 Apr 2022. from <https://www.psychiatry.org/psychiatrists/practice/dsm>
12. Pykes K (2020, May 1) Feature engineering for numerical data. Medium. Retrieved 6 Apr 2022. from <https://towardsdatascience.com/feature-engineering-for-numerical-data-e20167ec18>
13. Stemming and lemmatization in python. Data camp community (n.d.) Retrieved 6 Apr 2022. from <https://www.datacamp.com/community/tutorials/stemming-lemmatization-python>
14. Brownlee J (2020, June 27) How to encode text data for machine learning with scikit-learn. Machine learning mastery. Retrieved 6 Apr 2022. from <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/>

# Chapter 8

## Data Governance and Digital Transformation in Saudi Arabia



Kholod Saaed Al-Qahtani and M. M. Hafizur Rahman

**Abstract** Data governance and management solutions are experiencing a continuous data complexity, which make the process very expensive. The sophisticated data usage in many businesses has been a driving factor for new demands in the data handling, which has led to business requiring different ways of managing their data. Many businesses have resolved to the implementation of effective data governance as the most possible way to solve the data issue. Initially, many businesses attempted governing data but the process was not successful due to IT drive and the rigid processes that affected the activities that were carried out on system by system basis. The process has also suffered inadequate support from the organizations and structures for implementation. Data governance has highly recognized significances, but still no advanced research and development directed toward the achievement of the process. Due to the minimum research, this provides a room to advance research in the field in order to widen practice. In this paper, a mini literature review has been conducted to provide a structured and methodical approach for understanding aspects of research in data governance. The main objective is to provide an intellectual foundation for emerging scholars in the field of data governance—principles, models, challenges of data governance and digital transformation, and key dimensions of data governance strategy in Saudi Arabia. The study will incorporate IBM and agility data governance tools and model in integrating the useful information and for establishing and implementing the high level policies and procedures in the study. The mini literature review covered the importance of data governance, data governance principles, data governance models, the challenges facing data governance and digital transformation and the key dimensions in data governance strategy.

**Keywords** Digital transformation · Data governance · Data governance model

---

K. S. Al-Qahtani (✉) · M. M. H. Rahman  
Department of Computer Networks and Communications, CCSIT, King Faisal University,  
Al Hassa 31982, Saudi Arabia  
e-mail: [222453715@student.kfu.edu.sa](mailto:222453715@student.kfu.edu.sa)

M. M. H. Rahman  
e-mail: [mhrahman@kfu.edu.sa](mailto:mhrahman@kfu.edu.sa)

## 1 Introduction

The level of data usage in organizations has increased tremendously in the recent years, which plays a very crucial role in the operations of the various businesses in the country. Data has an influence on the strategic and operational decisions making functions in an organization, thus it is important to critically govern the data in the institutions. In the information system field, data government is an emerging concept requiring much attention [1]. As per the literature, researchers view data governance as the promising approach for the businesses in improving the level of data quality in their operations. According to [2], data governance requires organizations to set and enforce the priorities for managing and using data in their operations as a strategic asset. It is a combination of processes, tools, techniques and methods to ensure that companies have high quality data for effective operations, areas that are critical in the implantation of digital transformation in the country.

In essence, data governance forms the essential parts for the overall corporate governance strategy across sectors, thus follows the principles of corporate governance in place. A quality data governance strategy plays a great role within organizations to achieve clarity, maintain scope, establish accountability, increase confidence of using the organizational data, keep the focus and define a measurable success [3]. Currently, there is no particular approach for the implementation for succession, thus the literature argues that setting clear data governance strategy to govern the data, will play a critical role for the planned digital transformation in the country. There is little research on data governance particularly in cloud computing in Saudi Arabia. This mini literature review focuses on the contribution to the information system community in the country through the identification of the role digital transformation plays in the achievement of the Saudi Arabian vision 2030.

## 2 Review of the Literature on Data Governance

The study has undertaken an informative mini literature review in helping us and readers to understand the research landscape of data governance. The study will be very instrumental in developing a clear and guided data governance and data governance in digital transformation in cloud computing of Saudi Arabia [4]. According to...the research study followed the systematic literature guidelines and protocol, comprising a customized search, selection of study process and the all the applicable criteria. The study used the Saudi Digital Library and Google Scholar in conducting the search on the topic, by using the term data governance term as point of reference in covering all the relevant publications. The study focused on developed that have been recorded in data governance and data in digital transformation of Saudi Arabia for the period between 2003 and 2021 [5]. From the study, little literature is available of data governance in the cloud services in the country. All the literature confirms that data governance is a vital component for businesses that intend to accurately guide their information.

In addition, researchers described some challenges and issues that relate to data transformation to cloud, a set up outside the business premises, which are legal, technological and organizational. Reference [6] suggests that data accountability is a major component of data governance that contributes to best practices and mechanisms for a quality data governance process in an organization [7]. Accountability composes the main roles in data governance process that focus on the customers, providers, the subject, brokers, auditors, data carriers and data supervisory authority. Addressing the requirements that relate to confidential and personal information for cloud data governance is discussed in [2]. Data governance has been overlooked by many organizations in Saudi Arabia, leaving it for the recognized leaders in the areas such as Microsoft Corporation and Cloud Security Alliance [2]. An evaluation of the existing work on data governance shows that minimal effort has been made in the area. The area lacks standards; this study will bring valuable attributes and contributions on data governance in organization in Saudi Arabia.

### 3 Importance of Data Governance in Saudi Arabia

The Saudi Arabian government has established the National Data Management Office under Saudi Authority for Data (NDMO). The authority has recently started releasing regulations on data governance [8]. The regulations lay the important foundation for the Saudi Arabian society, which is empowered, enabled and securely protected data under clear and transparent frameworks. The move is an important strategy that aims at impacting the technological and other sectors of the Kingdom of Saudi Arabia.

Data governance ensures that individuals have a full control over their personal data. Governments, public and private sectors in the world have recognized that individuals own their personal data and follow them, an aspect that is relevant the organizations and individuals in the kingdom [9]. Data governance contains cookies that enable the user to access information in the website. The cookies protect and give permission for access of personal data.

Data governance and regulations in Saudi Arabia aims at contributing toward achieving a robust digital ecosystem for driving national development and economic diversifications across the sectors. Establishment of transparency and openness culture in the technology sector will enable monitoring the deployment of data intelligent solutions that are necessary for digital transformation. Cybersecurity and privacy protection comprise the foundation of the shared future of Saudi. They present the challenges encountered by data governance stakeholders and the possible solutions needed to address the issues for all the parties to work together [10]. Therefore, governments and companies in Saudi Arabia have to engage the industry-leading security technology in effective data governance and improve user experience.

Proper data governance leads to trusts from the users in the society. Therefore, ensuring accurate data protection calls for a shared responsibility and a unified standards and an independent verification process. All data facts must be subjected to verification by a unified standard across the Kingdom. This will enable organizations



adopt use of products that pass the security verification criteria for data governance and improve the digital transformation in the country [11]. The adoption of data protection will increase intelligent in the society and enable various industries in the society work together. Bringing together the public and private sectors will create a true value of data usage and accelerate the digital transformation of Saudi Arabia.

## **4 Principles of Data Governance in Saudi Arabia**

The principles stipulate overarching guidelines and norms for the policies and actions. They help in governing the design process for new data systems and define the changes to the existing sets of data and collection procedures, for successful data governance across sectors.

### ***4.1 Ownership and Accountability***

Data ownership and accountability need to be clearly defined in any successful data governance process. It is important for the organizations to have someone who takes the ownership of data governance to avoid sensitive information being ineffective, blunt and directionless [12]. Ownership and accountability should cut across the all the departments rather than centralization to a particular department, achievable through establishment of data governance council comprising of representatives from all the departments. The team formulate policies and rules for governing the information and have an authority to oversee adherence in the organization.

### ***4.2 Rules and Regulation Standardization***

Data governance process in the country should adhere and follow set of standardized rules and regulations across the sectors to help in data protection and ensure usage according to the relevant external regulations. Important data aspects such as access, definition, security standards and privacy policies have to be incorporated in the rules and guidelines to enable strictly enforcement in the organizations.

### ***4.3 Strategic and Valued Company Asset***

Data are a critical asset for any organization. They contain tangible, real and measurable value, which form the foundation for decision making in the organizations [13]. Thus, organizations have to recognize and appreciate data as among the val-

ued enterprise assets in all the programs and operations. Organization should set clear rules and process-driven ways on data assets in terms of controlling and accessibility.

#### 4.4 *Quality Standards*

Data quality requires a consistent management from the onset. Having a high quality and trustworthy data governance systems enable an effective stakeholder's decision making process in an organization [14]. This calls for periodic testing of enterprise data against the set and defined quality standards especially from the established technologies.

#### 4.5 *Data Transparency*

Organizations in the country have a duty to ensure that data governance implementation considers utmost transparency levels [15]. The organizations have a duty to retain the permanent records for all the relevant data governance processes and actions, showing reasons for data usage. Transparency protects organizations in cases of data breach and provides a learning experience for the stakeholders on ways of data usage throughout their organization.

### 5 Data Governance Model

Figure 1 presents a practical design of data governance model suitable for the National Data Management Office (NDMO) in developing and strengthening data governance maturity across Saudi Arabian organizations [16]. The model constitutes components vital for any data governance program. The model covers the following data governance activities.

- **Strategy and planning**—NDMO clearly defines the vision, values and mission of the data governance program and formulate a strategy alignment of the business for governing data, overseeing data transmission and management of these valuable organizational asset.
- **Data management**—the NDMO establishes their data governance programs that have oversight functions for data management in terms of data storage, quality and security and business insights [17]. This involves of the logistics of data in reference to the processes involved from the initial stages of data creation to the final stages of retirement of data.



Fig. 1 Data governance model

- **Structures, roles and responsibilities of the organization**—the data agency body ensure the assignment and formalizing of data accountability and decision making on data related issues to all levels of organizations.
- **Enablers in the organizations**—the government agencies that appropriate organizational environment are enablers for great data governance. This creates a strong motivation for having a sustained investment from organizational leadership suitable for attaining a strong culture in organizational data governance across the country [18]. In addition, it ensures that companies have the requisite in regard to appropriate technologies, tools and workforce capabilities, which are suitable in achieving good data governance.

**Fig. 2** Gartner golden triangle for data governance



## 6 Model Interpretation

A component in the model illustrates in detail a conclusive summary of what the component entails their significances in data governance, the goals of good data governance practices and explains the suitable ways of achieving these practices [19]. The model is suitable in expansion of the practical elements for the framework applicable for NMDO to adapt in appropriate data governance. In addition, the model is in alignment with the Golden Triangle, according to Gartner (2017) who suggests that People, Process and Technology, Data being at the center, are the appropriate elements for effective data governance.

People execute effort and assume the accountability functions in data governance. Processes are repeated in throughout the process to ensure the efforts directed become attainable. Technology provides the appropriate support for the implementation of the components in data governance process (Fig. 2).

## 7 Challenges for Data Governance Implementation

### 7.1 Technological Challenges

Security of data remains a major challenge for organizations are worried about their sensitive data during digital transformation to cloud environment. Software and hardware areas require much attention in digital transformation, thus the security of data in motion and cloud data storage is difficult to govern [18]. Privacy and protection aspects are major concerns for companies interested in the implementation of data governance and digital transformation. Many organizations in the country are not quite sure of the full control of their data especially when stored in the cloud. They

lack the guarantee of data confidentiality in the entire process. Performance bars organizations from fully adapting data governance and digital transformation.

The uncertainties of reliability, efficiency and scalability make it hard for organizations in the country to implement the process. Cost related to data governance and digital transformation is a setback for full implementation of the process [20]. For effective data governance, organizations have to install multiple servers, more technological expertise and research on the transformation process, which translates to a cost implication in business operations. Therefore most of the organizations fear directing resources in adaption of the new technical strategy, which hinders the overall data governance process in Saudi Arabia.

## **7.2 Organizational Challenges**

Data governance and digital transformation supports business to embrace new business opportunities in order to improve their current business performance and in responding to crisis situations. Data governance strategy supports the business functions in organizations [21]. The factors that hinder data implementation in businesses include, lack of support from top management, size of the organization and the readiness of the organization to embrace the technology. The literature review notes a significant gap for appropriate and effective data governance and digital transformation strategies in Saudi Arabia.

## **7.3 Legal Challenges**

Data governance in the digital transformation into clouds requires legal contracts between the organizations and the cloud actors. Companies find it complicated to understand the agreements in such legal and regulatory implications in relation to their business operations [22]. Saudi Arabia among other countries still lacks the compulsory regulatory support for data governance, protection and privacy. The legal factors affecting the implementation of data governance and digital transformation in organizations include; compliance with regulations, statutory regulations and physical location. They hinder full data governance in an organization thus forming a setback for the achievement of Saudi Arabian data governance vision 2030.

# **8 Key Dimensions of Data Governance**

Many reviews have recognized the need of implementing data governance and digital transformation in clouds. A data governance dimension provides an investigative



**Fig. 3** Key dimension for data governance in Saudi Arabia

effort that involves different stakeholders across sectors and the support they provide in achievement of the transformation strategy [23] (Fig. 3).

**8.1 Cloud Deployment Model**

This is a crucial aspect in data governance and digital transformation. The models here are public, private, community and hybrid [12]. In addressing the data governance, organizations must take into considerations the complexity and level of risks for reach of the deployment model.

## **8.2 *Service Delivery Model***

Services under digital transformation of data can be categorized into platform as a service, software as a service and Infrastructure as a service [24]. The data governance teams have the duty to take into account the characteristics for the service delivery model by defining appropriate policies that will enforce the roles and responsibilities on data governance. Actors refer to the people and organizations that take part in data governance and digital transformation processes. They play special roles and responsibilities in data governance as spelled by the data governance leaders.

## **8.3 *Service Level Agreement—SLA***

This is an agreement that exist between the consumers and the service providers, which states the services to be provided, the way it will be provided and the expected outcome if it fails to meet the intended functions [25]. It is an important element in data governance a it provides room for negotiation for the parties to have an agreement that protect both the parties.

## **8.4 *Organizational***

Data governance and digital transformation requires change management in the business to enable commitment from the IT staffs, senior level executive, and the management [26]. Support from top management enables a smooth implementation of the plan. Staffs in the companies need to learn the data governance functions for a skillset in the overall strategy.

## **8.5 *Technology***

This is a crucial component for data governance success. An appropriate technological aspect hinders successful data governance in organizations. They need to assess all the technological features that are available in the organization to enable effective data governance implementation.

## 8.6 *Environmental*

These factors comprise the external environmental aspects such as data protection act and government legislation [27]. During the design of data governance functions, the teams have to consider all the required environmental aspects and ensure that data governance functions comply with the environment. Through this, organizations will develop and build a strong data governance thus improving their business operations across all sectors of Saudi Arabia.

## 9 Conclusion

This study presents a useful contribution to the growing relevant research works in data governance in Saudi Arabia. A review on data governance literature reveals that there is little research activities that explores on governing data, an area where more organizations in the country consider data as their valuable asset. However, there are some studies that enable the understanding of data governance, which reveals the various progresses in exploring the functions that need governing data in the organizations. The study have analyzed a number of researches that focus on data governance functions where the importance, principles, models, challenges and key dimensions for data governance have been discussed in depth. Therefore, it is suggested that more studies on data governance to focus on data governance activities in regard to implantation and monitoring.

**Acknowledgements** This work was supported by the Deanship of Scientific Research, Vice 292 Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No.2040], through its KFU Research Summer initiative. The authors would like to sincerely thank to anonymous scholastic reviewers for their insightful comments and suggestions improve the quality and clarity of the paper.

## References

1. Abraham R, Schneider J, Vom Brocke J (2019) Data governance: a conceptual framework, structured review, and research agenda. *Int J Inf Manage* 49:424–438
2. Yallop AC, Gica OA, Moisescu OI, Coros MM, Seraphin H (2021) The digital traveller: implications for data ethics and data governance in tourism and hospitality. *J Consum Mark*
3. Al-Ruithe M, Benkhelifa E, Hameed K (2019) A systematic literature review of data governance and cloud data governance. *Pers Ubiquit Comput* 23(5):839–859. *Intr* (Al-Ruithe et al. 2019)
4. Benfeldt O, Persson JS, Madsen S (2020) Data governance as a collective action problem. *Inf Syst Front* 22(2):299–313
5. Alnofal FA, Alrwisan AA, Alshammari TM (2020) Real-world data in Saudi Arabia: current situation and challenges for regulatory decision-making. *Pharmacoepidemiol Drug Saf* 29(10):1303–1306
6. Janssen M, Brous P, Estevez E, Barbosa LS, Janowski T (2020) Data governance: organizing data for trustworthy artificial intelligence. *Gov Inf Quart* 37(3):101493



7. Omar A, Almaghthawi A (2020) Towards an integrated model of data governance and integration for the implementation of digital transformation processes in the Saudi universities. *Int J Adv Comput Sci Appl* 11(8):588–593
8. Memish ZA, Altuwaijri MM, Almoen AH, Enani SM (2021) The Saudi data and artificial intelligence authority (SDAIA) vision: leading the kingdom's journey toward global leadership. *J Epidemiol Global Health* 11(2):140
9. Otto B (2011) A morphology of the organisation of data governance
10. De Prieelle F, De Reuver M, Rezaei J (2020) The role of .. ecosystem data governance in adoption of data platforms by internet of-things data providers: case of dutch horticulture industry. *IEEE Trans Eng Manage*
11. Cheong LK, Chang V (2007) The need for data governance: a case study. In: *ACIS 2007 proceedings*, p 100
12. Brous P, Janssen M, Vilminko-Heikkinen R (2016, Sept) Coordinating decision-making in data management activities: a systematic review of data governance principles. In: *International conference on electronic government*, Springer, Cham, pp 115–125
13. Carroll SR, Garba I, Figueroa-Rodríguez OL, Holbrook J, Lovett R, Materechera S, Parsons M, Raseroka K, Rodriguez-Lonebear D, Rowe R, Sara R (2020) The CARE principles for indigenous data governance
14. Paskaleva K, Evans J, Martin C, Linjordet T, Yang D, Karvonen A (2017, Dec) Data governance in the sustainable smart city. In: *Informatics*, vol 4(no 4), Multidisciplinary Digital Publishing Institute, p 41
15. Khatri V, Brown CV (2010) Designing data governance. *Commun ACM* 53(1):148–152
16. Micheli M, Ponti M, Craglia M, Berti Suman A (2020) Emerging models of data governance in the age of datafication. *Big Data Soc* 7(2):2053951720948087
17. Weber K, Otto B, Osterle H (2009) One size does not fit all-a contingency approach to data governance. *J Data Inf Qual (JDIQ)* 1(1):1–27
18. Wende K (2007) A model for data governance-Organising accountabilities for data quality management
19. Korhonen JJ, Melleri I, Hiekkänen K, Helenius M (2014) Designing data governance structure: an organizational perspective. *GSTF J Comput (JoC)* 2(4)
20. Elouazizi N (2014) Critical factors in data governance for learning analytics. *J Learn Analytics* 1(3):211–222
21. Tse D, Chow CK, Ly TP, Tong CY, Tam KW (2018, Aug) The challenges of big data governance in healthcare. In: *2018 17th IEEE International conference on trust, security and privacy in computing and communications/12th IEEE International conference on big data science and engineering (TrustCom/BigDataSE)*, IEEE, pp 1632–1636
22. Winter JS, Davidson E (2019) Big data governance of personal health information and challenges to contextual integrity. *Inf Soc* 35(1):36–51
23. Lis D, Otto B (2021) Towards a taxonomy of ecosystem data governance
24. Juddoo S, George C, Duquenoy P, Windridge D (2018) Data governance in the health industry: investigating data quality dimensions within a big data context. *Appl Syst Innov* 1(4):43
25. Morabito V (2015) Big data governance. *Big Data Analytics* 83–104
26. Dai W, Wardlaw I, Cui Y, Mehdi K, Li Y, Long J (2016) Data profiling technology of data governance regarding big data: review and rethinking. *New Generations, Information Technology*, pp 439–450
27. Al-Ruithi M, Benkhelifa E, Hameed K (2016, Aug) Key dimensions for cloud data governance. In: *2016 IEEE 4th International conference on future internet of things and cloud (FiCloud)*, IEEE, pp 379–386

## Chapter 9

# A Waveguide-Fed Hybrid Graphene Plasmonic Nanoantenna for On-Chip Wireless Optical Communication



Richard Victor Biswas

**Abstract** In this paper, a waveguide-fed hybrid graphene plasmonic nanoantenna operating in the optical frequency spectrum for the point-to-point wireless nanolink has been studied, modeled, and simulated. For the improvement of certain radiation characteristics of the proposed nanoantenna at 193.5 THz, an electrostatic gate bias potential of 0.7 eV has been set at every graphene layer—which in turn allows the transverse electromagnetic surface plasmon polariton to resonate. Gain, directivity, and efficiency of 12.4 dB, 13.3 dBi, and 93.23%, respectively, have been obtained when combining the graphene, aluminum, and SiO<sub>2</sub> layers in the all-around W-shape slotted radiators fed by a waveguide port. Side lobes of −18.7 dB and reflection coefficient of −28.22 dB have been attained with a unidirectional radiation pattern. Contingent upon the performance of the proposed nanoantenna, it can be integrated into the inter- or intra-on-chips for safe optical data transmission at the resonant frequency of 193.5 THz.

**Keywords** Point-to-point wireless optical nanolink · Graphene · Optical frequency spectrum · Surface plasmon polariton

## 1 Introduction

The nanophotonic research community is more inclined to plasmonic optical nano-scale antennas due to their extensive energy localization of the visible frequency spectrum in the subwavelength volume. Owing to their excellent control over electromagnetic fields generating highly oriented radiation patterns, minimal power consumption, and better impedance matching, these optical nanoantennas (NAs) are being widely integrated into several emerging applications, for instance, spectroscopy, nanophotonic circuitry, sensing, non-linear optics, improved photoemission and photodetection, and optical metasurfaces. As compared to the microwave

---

R. V. Biswas (✉)

Department of Electrical and Electronic Engineering, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh  
e-mail: [victor.biswas18@gmail.com](mailto:victor.biswas18@gmail.com)

and radio frequency antennas, optical plasmonic NAs cannot radiate beams over a long distance because of the intrinsic ohmic losses of metals in the visible spectrum of light. One of the feasible areas to utilize the optical NAs is the photonic integrated circuits (PICs) where chip-to-chip data transfer can be accomplished with ease at as high as 10–100 GHz speed. Once the radiating segments of a group of optical NAs receive an in-plane optical signal and subsequently radiate electromagnetic beams toward several on-chip receivers, an intra- or inter-on-chip nanolink for optical communication can then be constructed; for more directive radiation patterns, graphene layers controlled by electrostatic gate bias voltage are taken into account along with other materials suitable for optical applications.

To create an optical wireless nanolink, Alù and Engheta presented a set of complemented dipole NAs which showed satisfactory performance and low impedance mismatch losses as compared to its wired counterparts [1]. In earlier investigations on the optical wireless nanolinks, some types of optical NAs were proposed, namely Yagi-Uda, graphene patch, phase array, cross dipole, dipole-loop, and circular hybrid plasmonic NAs [2–7]. With the intention of acquiring the transmission bandwidth as high as 0.1–100 THz and tunability in optical communications, plasmonic antennas with graphene layers could be one of the promising options. Graphene is an allotrope of carbon, whose atoms are arranged in a two-dimensional hexagonal lattice structure. It has amazing optical as well as mechanical properties and intrinsic tunable electrical conductivity which permits the propagation of high-frequency electrical signals. It has a mean free path of 300–500 nm for ballistic carrier transport and carrier mobility in the range of 8000–10,000  $\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$ . Owing to the high carrier mobility at the optical spectrum, graphene increases the propagation of confined electromagnetic surface plasmon polariton (SPP) waves [8].

A *W*-shape slotted hybrid graphene plasmonic nanoantenna (WSHGPNA) emitting a unidirectional electromagnetic radiation beam with high efficiency of 93.23% at 193.5 THz (wavelength of 1550 nm) has been proposed in this study. The conductivity of graphene has been altered in such a way that the overall radiating structure absorbs the maximum incident power at the wavelength of 1550 nm. At the frequency of interest, the chemical potential of graphene has been tuned to 0.7 eV, essentially varying the electrostatic biasing or chemical doping across the graphene sheets in reality [9]. The directors composed of aluminum, graphene, and  $\text{SiO}_2$  in the proposed NA further ensure the reflection coefficient as low as  $-28.22$  dB and satisfactory radiation efficiency of 93.23%. Moreover, its unidirectional radiation beam directivity of 13.3 dBi is conceivably a notable feature, motivating to realize the point-to-point wireless nanolinks for optical communication.

The next sections of this paper are organized as follows: conductivity, surface impedance, power absorption properties, and the modeling of graphene have been discussed in Sect. 2. In Sect. 3, the performance of a waveguide-fed hybrid graphene plasmonic nanoantenna has been presented. The optimization method for realizing our proposed WSHGPNA has been delineated in detail in Sect. 4. The purpose of various modified structures as well as layers in the radiator and the optimum dimensions of WSHGPNA has been given in Sect. 5. The radiation characteristics of our proposed NA and the comparison table including all reported optical NAs

resonating at 193.5 THz as well as our proposed one have been added in Sect. 6. Eventually, in the “conclusion” section, a summary of this study has been provided.

## 2 Properties of Graphene

### 2.1 Conductivity, Surface Impedance, Power Absorption of Graphene

According to Kubo’s formula,  $\sigma_g(\omega, \mu_c, \Gamma, T)$  is the conductivity of graphene which is a function of inter-band and intra-band transitions as expressed in Eqs. (1), (2), and (3) [10].  $K_B$  and  $T$  are the Boltzmann constant and temperature at 300 K, respectively.

$$\sigma_g = \sigma_{\text{intra}} + \sigma_{\text{inter}} \quad (1)$$

$$\sigma_{\text{intra}}(\omega, \mu_c, \Gamma, T) = -j \frac{e^2 K_B T}{\pi \hbar^2 (\omega - j\Gamma)} \left\{ \frac{\mu_c}{K_B T} + 2 \ln \left( e^{-\frac{\mu_c}{K_B T}} + 1 \right) \right\} \quad (2)$$

$$\sigma_{\text{inter}}(\omega, \mu_c, \Gamma, T) = \frac{-je^2}{4\pi \hbar^2} \ln \left( \frac{2|\mu_c| - (\omega - j\Gamma)\hbar}{2|\mu_c| + (\omega - j\Gamma)\hbar} \right) \quad (3)$$

The characteristic impedance of the supported transverse magnetic (TM) plasmons in the graphene sheet is given as follows [10]:

$$Z_g = Z_C = \frac{1}{\sigma_g} = R_g + jX_g = \frac{k_{\text{SPP}}}{\omega \varepsilon_0 \varepsilon_{r(\text{eff})}} \quad (4)$$

where  $R_g$  and  $X_g$  are the surface resistance and reactance of the graphene sheet, correspondingly;  $\varepsilon_{r(\text{eff})}$  is the effective dielectric constant of the surrounding media. The chemical potential ( $\mu_c$ ) and the electrostatic biasing gate voltage ( $V_g$ ) are strongly linked to each other [11]:

$$\mu_c = E_f = \hbar v_f \sqrt{\frac{\pi \varepsilon_{\text{ox}} \varepsilon_0 V_g}{e t_{\text{ox}}}} = \hbar v_f \sqrt{\frac{\pi C_{\text{ox}} V_g}{e}} \approx \hbar v_f \sqrt{\pi N} \quad (5)$$

where  $v_f$  is the velocity of electrons in Fermi energy level ( $\approx 10^6 \text{ ms}^{-1}$ ),  $\varepsilon_{\text{ox}}$  is the permittivity of  $\text{SiO}_2$  layer,  $t_{\text{ox}}$  is the oxide thickness,  $C_{\text{ox}} (= \frac{\varepsilon_{\text{ox}} \varepsilon_0}{t_{\text{ox}}})$  is the electrostatic gate capacitance per unit area, and  $N$  ( $\approx \frac{C_{\text{ox}} V_g}{e}$ ) is the carrier concentration [12]. Therefore, more chemical potential of graphene is evident for more electrostatic biasing voltage.

## 2.2 Graphene Modeling

Once the electrostatic gate bias voltage is applied, the epsilon-near-zero effect of graphene becomes prominent, resulting in an absorption of graphene power given by Eq. (6) [13].

$$P_g = \frac{\text{Re}(\sigma_g)E^2}{2} \quad (6)$$

Referring to Eq. (6), our hybrid graphene nano-scale structure absorbs a wide range of powers ( $P_g$ ) when the conductivity of the graphene layer and the incident laser photon energy ( $E$ ) is adjusted dynamically. The chemical potential ( $\mu_c$ ) of graphene (0.7 eV) and the charged particle scattering rate ( $\Gamma$ ) of 0.00051423 eV have been chosen with the aim of reconfiguring our proposed NA. Using the Ansys Lumerical software, which follows the finite-difference time-domain method (FDTD), the relationship among chemical potential, frequency, and conductivity of graphene material has been obtained systematically. A direct connection of the real and imaginary parts of  $\sigma_g$  with respect to  $\mu_c$  is demonstrated in Fig. 1a. Even though both parts of conductivity of graphene for  $\mu_c = 0.0$  eV have a proportional relationship with frequency (190–195 THz) as shown in Fig. 1b, a negative correlation sustains between each part of conductivity and frequency in the case of  $\mu_c = 0.7$  eV as illustrated in Fig. 1c. The real parts of graphene conductivity,  $\text{Re}(\sigma_g)$ , fall drastically with the enhancement of chemical potential at 193.5 THz;  $\text{Re}(\sigma_g)$  nearly equals 60.85  $\mu\text{S}$  and 0.1308  $\mu\text{S}$  for  $\mu_c = 0$  eV and 0.7 eV, respectively. Conversely,  $\text{Im}(\sigma_g) \approx -0.04$   $\mu\text{S}$  and  $\text{Im}(\sigma_g) \approx 42.3$   $\mu\text{S}$  for  $\mu_c = 0$  eV and 0.7 eV, exclusively. Hence, the radiation characteristics of NAs depend largely on the impact of  $\text{Im}(\sigma_g)$  as compared to its counterpart. Graphene behaves as a transparent medium for  $\mu_c = 0$  eV, while it acts as an absorptive medium at 1550 nm for  $\mu_c = 0.7$  eV.

## 3 Design of a Waveguide-Fed Hybrid Graphene Plasmonic Nanoantenna

Figure 2a illustrates a waveguide-fed hybrid graphene plasmonic nanoantenna having a volume of  $1000 \times 1000 \times 207.5$  nm<sup>3</sup>. A combination of aluminum, SiO<sub>2</sub>, and graphene layers has been selected for this nano-scale antenna structure. As depicted in Fig. 2b, the  $S_{11}$  of this NA is  $-12.667$  dB at 193.5 THz. The directivity, gain, and efficiency of it are 10.147 dBi, 7.291 dB, and 71.853%, respectively. It is apparent that the radiation characteristics of this NA at the frequency of interest have to be improved further by tuning the antenna parameters.

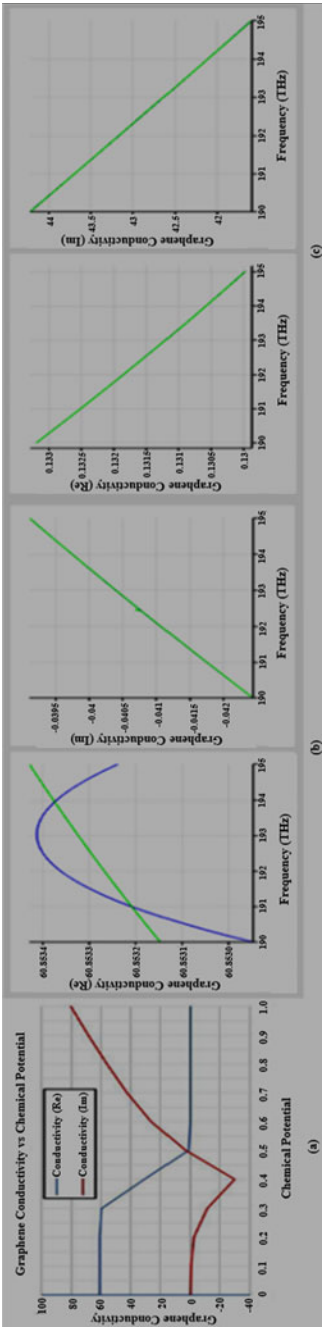


**Fig. 1** a At 193.5 THz, conductivity curves of graphene with respect to chemical potentials for a relaxation time = 0.1 ps and T = 300 K. For **b**  $\mu_c = 0.0$  eV and **c**  $\mu_c = 0.7$  eV, the change in real and imaginary parts of conductivity curves of graphene as functions of frequency (190–195 THz)

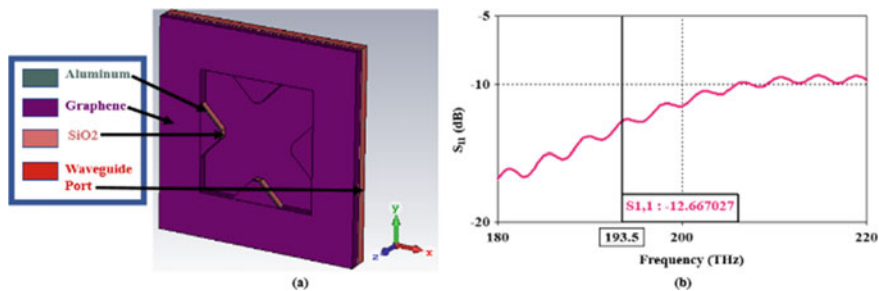


**Fig. 2** **a** Prospective view and **b** return loss ( $S_{11}$ ) at 193.5 THz of the waveguide-fed hybrid graphene plasmonic nanoantenna

## 4 Optimization of the Waveguide-Fed Hybrid Graphene Plasmonic Nanoantenna

In this section, the parametric analysis of the waveguide-fed hybrid graphene plasmonic nanoantenna has been executed by varying the radiator geometry, size, thickness, metal, substrate, and chemical potential of graphene to have optimum radiation characteristics.

### 4.1 Parametric Analysis Based on Radiator Geometry, Size, and Thickness

$S_{11}$  versus frequency plot of three geometries (V-shape slot, V-shape extended slot, and W-shape slot) of the radiating portion has been depicted in Fig. 3a; the radiation characteristics of the three radiating geometries are more or less the same, however, the efficiency of the V-shape extended slot is not satisfactory as compared to others. Since V-shape slots create straight paths along the radiators, magnetic field vectors partially collide with them; hence, the efficiency (78.118%) of this kind is better than the V-shape extended slot. The W-shape slots form the radiating structure in such a manner that both electromagnetic field vectors strike the radiators symmetrically, deriving a unidirectional radiation pattern with excellent efficiency of 81.801% and return loss of  $-29.285$  dB. As a result, for the imminent optimization steps, radiators designed by the W-shape slots have been taken into consideration. Table 1 records the radiation characteristics of every geometry discussed above.

Three WSHGPNA's possessing different sizes ( $750 \times 750 \text{ nm}^2$ ,  $1000 \times 1000 \text{ nm}^2$ , and  $1250 \times 1250 \text{ nm}^2$ ) with aluminum metal have been successfully simulated in order to extract the  $S_{11}$  versus frequency plots. The performance metrics of our WSHGPNA are demonstrated in Table 2. Referring to Fig. 3b, when the footprint size of WSHGPNA enlarges, the return loss reduces but the efficiency essentially improves. The smallest  $S_{11}$  parameter ( $-29.285$  dB) has been achieved for  $750 \times$

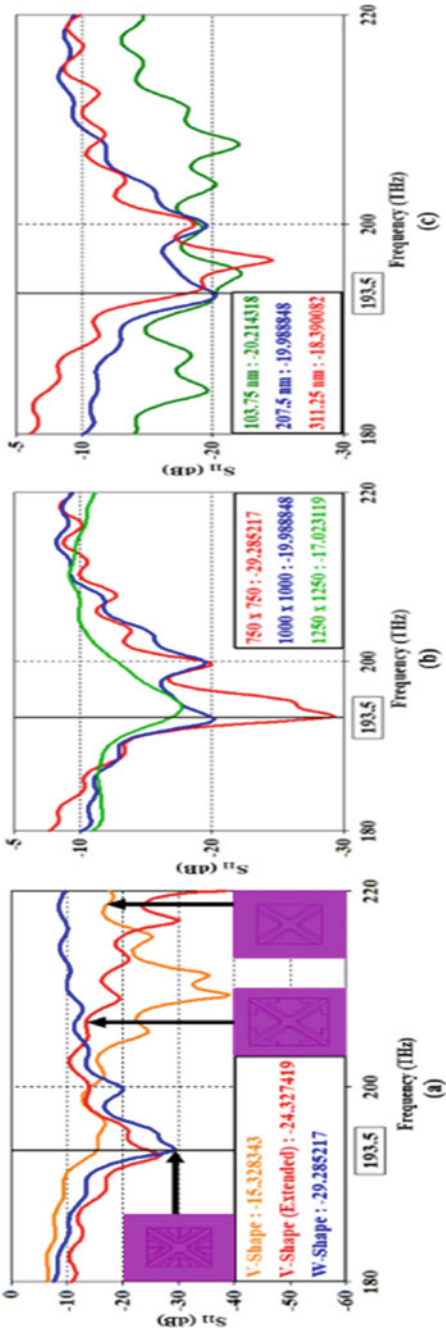


**Fig. 3** Reflection coefficients ( $S_{11}$ ) of NA **a** radiator geometries, **b** sizes, and **c** thicknesses at 193.5 THz



**Table 1** Comparison of the proposed WSHGPNA designs based on radiator geometries

| Radiator geometry  | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|--------------------|---------------|-------------------|-----------|----------------|
| V-shape            | -15.328       | 10.703            | 8.361     | 78.118         |
| V-shape (extended) | -24.327       | 10.524            | 7.708     | 73.242         |
| W-shape            | -29.285       | 11.062            | 9.085     | 81.801         |

**Table 2** Comparison of the proposed WSHGPNA designs based on sizes

| Size (nm <sup>2</sup> ) | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|-------------------------|---------------|-------------------|-----------|----------------|
| 750 × 750               | -29.285       | 11.062            | 9.085     | 81.801         |
| 1000 × 1000             | -19.989       | 13.204            | 11.174    | 84.626         |
| 1250 × 1250             | -17.023       | 14.202            | 11.733    | 82.615         |

**Table 3** Comparison of the proposed WSHGPNA designs based on thicknesses

| Thickness (nm) | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|----------------|---------------|-------------------|-----------|----------------|
| 103.75         | -20.213       | 11.964            | 9.465     | 79.112         |
| 207.5          | -19.989       | 13.204            | 11.174    | 84.626         |
| 311.25         | -18.379       | 12.843            | 10.421    | 81.141         |

750 nm<sup>2</sup> feature size even though its efficiency is not quite significant as opposed to other sizes. For the largest size,  $S_{11}$  does not get reduced further but instead increases noticeably. Consequently, the size of WSHGPNA has been selected between 750 × 750 nm<sup>2</sup> and 1250 × 1250 nm<sup>2</sup> in such a way that minimum reflection coefficient and good efficiency are apparent. In fact, a size of 1000 × 1000 nm<sup>2</sup> for our proposed NA has been chosen which provides much lower  $S_{11}$  (-19.989 dB) and excellent efficiency (84.626%) as compared to other ones.

In Fig. 3c, the return losses with respect to frequency for three WSHGPNA's having different thicknesses [103.75 nm (0.5 × 207.5 nm), 207.5 nm, and 311.25 nm (1.5 × 207.5 nm)] are illustrated. The minimum  $S_{11}$  is achieved as the thickness decreases; however, efficiency degrades. While all the listed thicknesses show comparable return losses, WSHGPNA with a thickness of 207.5 nm has been preferred for its remarkable efficiency of 84.626%. The radiation characteristics of our proposed WSHGPNA have been gathered in Table 3.

#### 4.2 Parametric Analysis in Terms of Metal, Substrate, and Chemical Potential of Graphene

In Fig. 4a, the effect of four metals, namely Ag, Al, Au, and Cu, on WSHGPNA has been portrayed, and in Table 4, its radiation characteristics are reported. Although the

minimum return loss ( $-19.989$  dB) has been obtained for Ag among all the listed ones, its efficiency (84.626%) is less than Al. At 193.5 THz, since the reflection coefficients and efficiency of Au and Cu are not prominent as compared to Al ( $S_{11} = -19.566$  dB and efficiency = 86.169%), Al metal has been sandwiched between the  $\text{SiO}_2$  layer and the graphene in the radiating structure of WSHGPNA.

The effect of different substrate materials (Polymide, Quartz, Rogers RO4003C,  $\text{SiO}_2$ , and Teflon) on WSHGPNA with Al metal is depicted in Fig. 4b, and its radiation characteristics are organized in Table 5. Due to the strong plasmonic response of an amalgamation of  $\text{SiO}_2$  and Al in the optical domain, the best radiation performances in terms of minimum  $S_{11}$  ( $-19.566$  dB) and efficiency (86.169%) have been obtained. Conversely, Polymide, Quartz, and Rogers RO4003C have not obtained high power enhancement factors and lower return losses at 193.5 THz. Teflon may be preferred as it provides the lowest reflection coefficient ( $-19.691$  dB) at the frequency of interest; nevertheless, the efficiency (85.239%) is slightly less than  $\text{SiO}_2$ . Therefore, the  $\text{SiO}_2$  substrate has been used as the ground plane and one of the layers of the radiating structure in our WSHGPNA design.

Table 6 accumulates the values of reflection coefficient, directivity, gain, and efficiency for 0.1 eV, 0.3 eV, 0.5 eV, 0.7 eV, and 0.9 eV chemical potentials of graphene at 193.5 THz. The return losses for different chemical potentials of graphene at 193.5 THz are depicted in Fig. 4c. Even though the lowest return loss of  $-30.13$  dB has been obtained by setting 0.9 eV at the frequency of interest, the radiation characteristics [directivity (13.342 dBi), gain (12.384 dB), and efficiency (92.821%)] observed owing to 0.7 eV at the respective frequency are more acceptable. For 0.1 eV, 0.3 eV, and 0.5 eV, a relatively moderate level of reflection signal is noticeable from the overall structure. In short, for our final optimized WSHGPNA, graphene with 0.7 eV chemical potential has been considered because such biasing voltage brings forth an excellent NA efficiency in contrast to other values.

## 5 Analysis of the Optimized Waveguide-Fed Hybrid Graphene Plasmonic Nanoantenna for On-Chip Wireless Optical Communication

### 5.1 Details of This Proposed Nanoantenna Structures and Dimensions

A homogenous  $\text{SiO}_2$  ground plane (Fig. 5b) and a rectangular graphene structure with a rectangular cavity of 87.5 nm thickness (Fig. 5c) where the composite radiators [graphene, aluminum (Al), and  $\text{SiO}_2$ ] forming all-around W-shape slots (Fig. 5d–f) have been arranged to realize this optimized NA (Fig. 5g) fed by a waveguide port (Fig. 5a). Al has a high density of free electrons (i.e., 3 electrons per atom in its conduction band), a high bulk plasma frequency ( $\omega_p = 15$  eV) leading to substantial surface plasmon responses, a broad resonant frequency spectrum (UV, optical, and

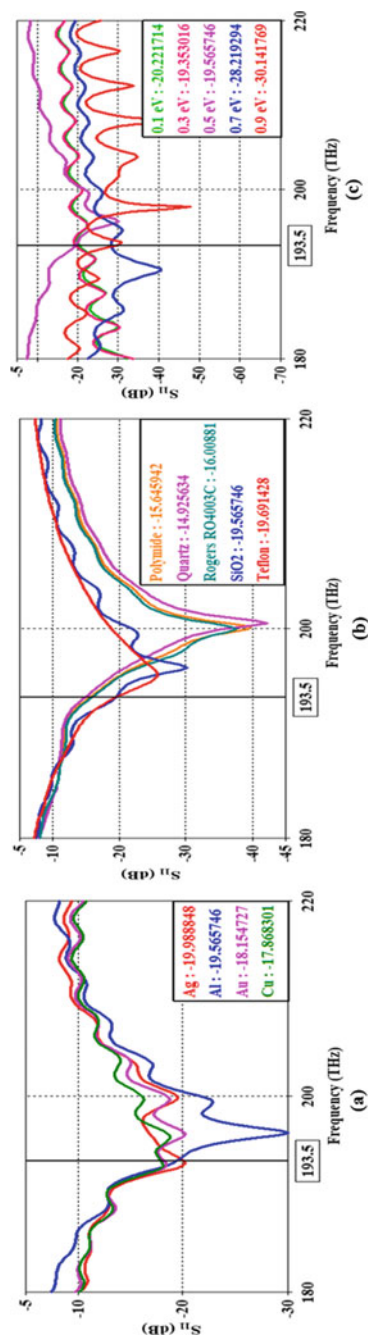


**Fig. 4** Return losses ( $S_{11}$ ) of the optimized W-shape slotted NA **a** metals, **b** substrates, and **c** chemical potentials of graphene at 193.5 THz

**Table 4** Comparison of the proposed WSHGPNA designs based on metals

| Metal | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|-------|---------------|-------------------|-----------|----------------|
| Cu    | −17.872       | 13.412            | 11.423    | 85.169         |
| Al    | −19.566       | 13.174            | 11.352    | 86.169         |
| Au    | −18.159       | 13.033            | 10.791    | 82.798         |
| Ag    | −19.989       | 13.204            | 11.174    | 84.626         |

**Table 5** Comparison of the proposed WSHGPNA designs based on substrate material

| Substrate        | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|------------------|---------------|-------------------|-----------|----------------|
| Polymide         | −15.646       | 13.351            | 11.294    | 84.593         |
| Quartz           | −14.926       | 13.302            | 11.212    | 84.288         |
| Rogers RO4003C   | −16.009       | 13.361            | 11.323    | 84.747         |
| SiO <sub>2</sub> | −19.566       | 13.174            | 11.352    | 86.169         |
| Teflon           | −19.691       | 13.353            | 11.382    | 85.239         |

**Table 6** Comparison of the proposed WSHGPNA designs based on chemical potential of graphene at 193.5 THz

| Chemical potential of graphene (eV) | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|-------------------------------------|---------------|-------------------|-----------|----------------|
| 0.1                                 | −20.227       | 13.353            | 11.762    | 88.085         |
| 0.3                                 | −19.358       | 13.273            | 11.621    | 87.553         |
| 0.5                                 | −19.566       | 13.174            | 11.352    | 86.169         |
| 0.7                                 | −28.217       | 13.342            | 12.384    | 92.821         |
| 0.9                                 | −30.13        | 12.831            | 11.673    | 90.975         |

near-infrared), no bulk oxidization issue, and excellent non-linear optical properties [14]; hence, Al is a perfect choice for our proposed NA. The graphene slabs enclosing the composite radiating structure create a waveguide that directs the unidirectional electromagnetic radiation beam toward the Z direction as illustrated in (Fig. 5c–f).

The motives for forming W-shape slots on the radiating portion around the graphene passage are to diminish the shock waves at the dielectric-aluminum-graphene interfaces at the sub-millimeter and optical frequency regime and to discharge the confined energy from SiO<sub>2</sub> which results in considerable surface wave losses [15]. By applying 0.7 eV electrostatic gate bias potentials of graphene layers, the main lobe direction, absorption power, and intensity of radiation pattern have been regulated. The optimized dimensions of several layers of the proposed NA, listed in Table 7, are shown in Fig. 5h, i. For optical and mid-infrared applications, e.g., multipath intra- or inter-on-chip wireless nanolink, the optimized length and width of the radiating structure have been chosen to be 575 nm and thickness to be 87.5 nm.

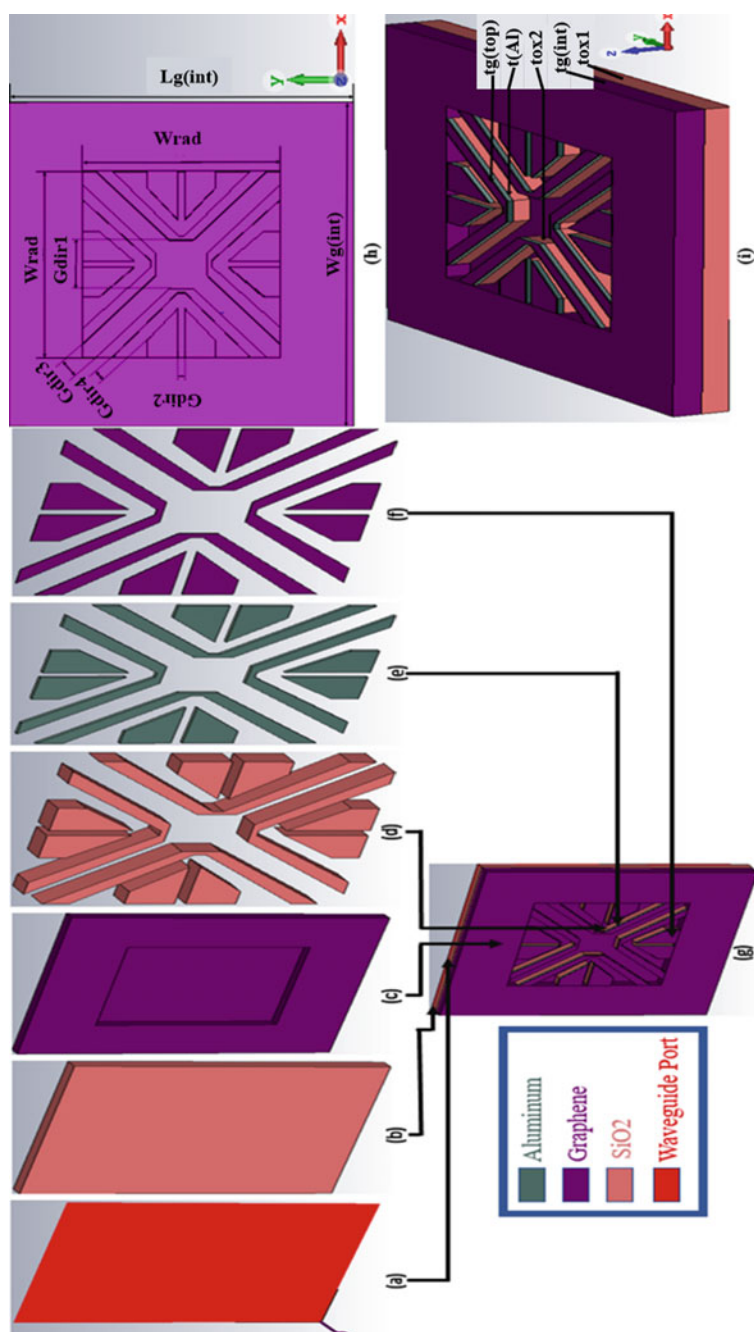


**Fig. 5** **a–g** Modified layers of various materials (aluminum, graphene  $-0.7$  eV, and SiO<sub>2</sub>) to observe the resonance of optimized waveguide-fed plasmonic NA at 193.5THz. **h** Top and **i** side schematic views of the proposed NA

**Table 7** Optimum dimensions of the proposed waveguide-fed hybrid graphene plasmonic nanoantenna

| Dimension                                           | Value (nm) |
|-----------------------------------------------------|------------|
| Length of intermediate graphene ( $L_{g(int)}$ )    | 1000       |
| Width of intermediate graphene ( $W_{g(int)}$ )     | 1000       |
| Length of radiating structure ( $L_{rad}$ )         | 575        |
| Width of radiating structure ( $W_{rad}$ )          | 575        |
| Gap between two director 1 ( $G_{Dir1}$ )           | 300        |
| Gap between two director 2 ( $G_{Dir2}$ )           | 40         |
| Gap between two director 3 ( $G_{Dir3}$ )           | 113.14     |
| Gap between two director 4 ( $G_{Dir4}$ )           | 100        |
| Thickness of bottom SiO <sub>2</sub> ( $t_{ox1}$ )  | 100        |
| Thickness of intermediate graphene ( $t_{g(int)}$ ) | 107.5      |
| Thickness of top SiO <sub>2</sub> ( $t_{ox2}$ )     | 62.5       |
| Thickness of aluminum ( $t_{Al}$ )                  | 20         |
| Thickness of top graphene ( $t_{g(top)}$ )          | 5          |

## 5.2 Simulation Results

For our proposed NA resonating at 193.5 THz, the return loss ( $S_{11}$ ) and  $-10$  dB impedance bandwidth of  $-28.22$  dB and 88.43 THz (171.57 THz–260 THz), respectively, have been achieved as depicted in Fig. 6a. In Fig. 6b, c, the electric field and magnetic field patterns in the  $XY$  plane at 193.5 THz are illustrated, respectively. Intense electric field vectors propagate vertically for a phase of  $90^\circ$ , which have been observed both at the outer edges of all directors as shown in Fig. 6b. However, as the horizontal directors trap the distribution of electric field vectors to some extent, the dominance of more  $E$ -field is noticed here as compared to their counterparts. Because the magnetic field vectors move along  $X$ -axis for a phase of  $180^\circ$ , only horizontal directors experience the strong  $H$ -field effects, referring to Fig. 6c.

The 3D far-field directivity of the proposed NA is demonstrated in Fig. 7a where maximum directivity of 13.3 dBi has been obtained. In Fig. 7b, c, the 2D and 3D far-field plots of gain for this NA at 193.5 THz are depicted, individually. Along the elevation plane, the proposed NA radiates the electromagnetic waves with a maximum gain of 12.4 dB at 1550 nm. From the 2D radiation pattern, gain lies within  $\theta$  of  $-30^\circ$  to  $30^\circ$ . The far-field radiation patterns and parametric values of the total electric field in various planes of the proposed NA are illustrated and accumulated in Fig. 7d and Table 8, respectively. At 193.5 THz, the proposed NA emanates unidirectional electromagnetic beams along the  $XZ$  plane (elevation- $E$  plane) with a main lobe magnitude of 27.1 dBV and negligible side lobe level of  $-18.7$  dB. The 3 dB angular widths or half-power beam widths (HPBW) for the total electric field of this NA in  $XY$  (azimuth- $H$  plane),  $XZ$ , and  $YZ$  planes are  $63.3^\circ$ ,  $35.5^\circ$ , and  $47.1^\circ$ , correspondingly. Along the elevation plane, the main lobe direction of  $0^\circ$

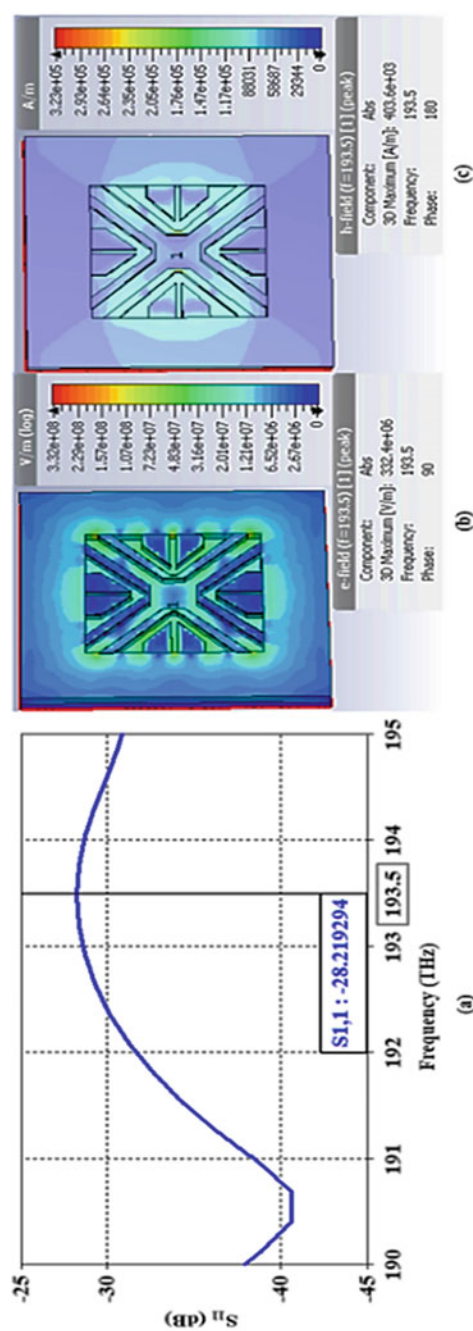


Fig. 6 a Return loss versus frequency plot of the proposed NA and b, c plasmonic behavior of it in the electric and magnetic fields at 193.5 THz

at 1550 nm wavelength implies that the proposed waveguide-fed hybrid graphene plasmonic nanoantenna generates a kind of directive radiation that is impervious to external stimuli (i.e., field and noise), establishing point-to-point on-chip nanolinks in optical systems.

The notable antenna characteristics, in terms of area, reflection coefficient, directivity, gain, and efficiency, of the proposed waveguide-fed hybrid graphene plasmonic nanoantenna have been compared with that of the previously reported optical NAs in Table 9. As opposed to the directivity of the unidirectional radiation pattern of the waveguide-fed hybrid plasmonic patch nanoantenna [16], several directors in our proposed design boost the directivity to more than 7.3 dBi at 1550 nm wavelength. In addition, our efficient NA offers higher gain (12.4 dB), lowest reflection coefficient ( $-28.22$  dB), and smaller size ( $1 \mu\text{m}^2$ ) in contrast to the reported NA in the literature [16]. The optical NAs at 193.5 THz from literatures [6, 18–20] achieve directivities less than 13.3 dBi and return losses greater than  $-28.22$  dB. Even though the hybrid plasmonic horn nanoantenna [20] and the hybrid plasmonic waveguide-fed broadband nanoantenna [21] are somewhat smaller than our proposed NA, these two hybrid plasmonic NAs have reflection coefficients  $> -28.22$  dB and gains  $< 12.4$  dB. Lastly, the directivity and area of both the Yagi-Uda nanoantenna [22] and Yagi-Uda nanoantenna array [23] are larger than the listed ones; conversely, the proposed waveguide-fed hybrid graphene plasmonic nanoantenna has obtained remarkable efficiency of 93.23%.

## 6 Conclusion

Using a commercially available finite element method (FEM) simulator known as CST Microwave Studio, a waveguide-fed hybrid graphene plasmonic nanoantenna resonating at 193.5 THz has been designed and simulated. It achieved a maximum gain and directivity of 12.4 dB and 13.3 dBi, respectively, by optimizing several parameters and the conductivity of graphene through electrostatic biasing. Aluminum,  $\text{SiO}_2$ , and graphene layers in the *W*-shape slotted radiating portion of the proposed NA have been placed in such a manner that a minimum return loss ( $S_{11}$ ) of  $-28.22$  dB and remarkable efficiency of 93.23% are obtained. Because of its excellent radiation characteristics, intra- or inter-on-chip wireless nanolink for secure optical data transfer with low latency can be implemented by considering our proposed optical NAs.



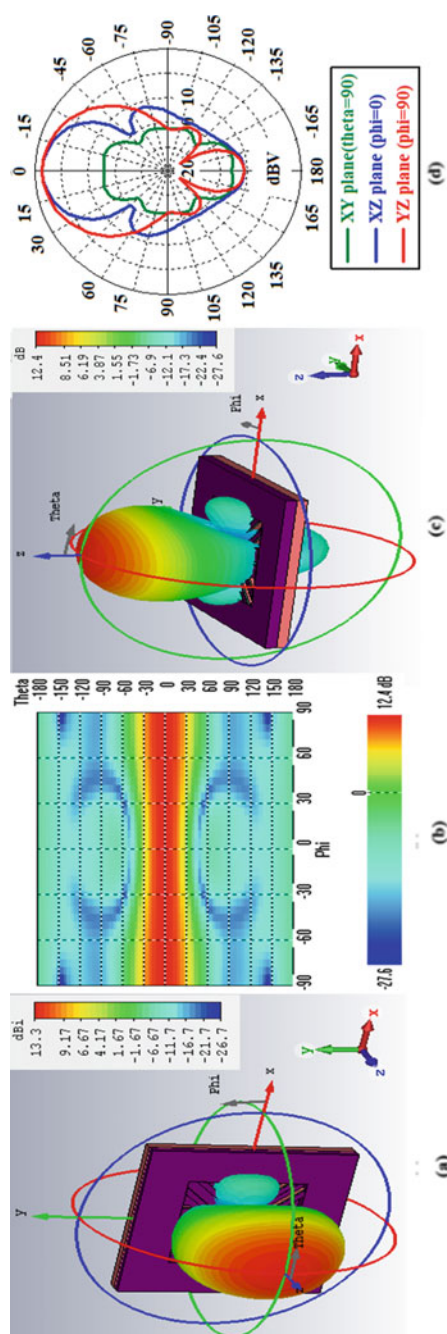


**Fig. 7** **a** 3D plot of far-field directivity, **b** 2D plot of gain, **c** 3D plot of far-field gain, and **d** far-field radiation patterns for the total electric field for the proposed NA resonating at 193.5 THz

**Table 8** Characteristics of far-field radiation pattern of the proposed NA

| Parameter                 | XY plane ( $\theta = 90^\circ$ ) | XZ plane ( $\Phi = 0^\circ$ ) | YZ plane ( $\Phi = 90^\circ$ ) |
|---------------------------|----------------------------------|-------------------------------|--------------------------------|
| Main lobe magnitude (dBV) | 4.74                             | 27.1                          | 27.1                           |
| Main lobe direction       | $-15^\circ$                      | $0^\circ$                     | $0^\circ$                      |
| Angular width (3 dB)      | $63.3^\circ$                     | $35.5^\circ$                  | $47.1^\circ$                   |
| Side lobe level (dB)      | $-6.7$                           | $-18.7$                       | $-18.7$                        |

**Table 9** Comparative analysis among previously reported optical plasmonic NAs with the proposed one

| References  | Size ( $\mu\text{m}^2$ ) | $S_{11}$ (dB) | Directivity (dBi) | Gain (dB) | Efficiency (%) |
|-------------|--------------------------|---------------|-------------------|-----------|----------------|
| [22]        | $10 \times 19.5$         | $-13.2$       | 16.6              | 12.9      | 77.71          |
| [23]        | $90 \times 90$           | –             | 18                | 6         | 33.33          |
| [20]        | $0.45 \times 0.625$      | $-15.7$       | –                 | 4.7       | –              |
| [21]        | $0.55 \times 0.5$        | $-13$         | –                 | 5         | –              |
| [6]         | $3.5 \times 3.5$         | –             | 8.79              | –         | –              |
| [19]        | $0.55 \times 0.5$        | $-12.86$      | 8.79              | 7.792     | 88.64          |
| [18]        | $5 \times 5$             | $-22$         | 8.6               | –         | –              |
| [17]        | $0.7 \times 0.7$         | $-21$         | 6.9               | 6.6       | 95.65          |
| [16]        | $5.6 \times 5.6$         | $-10.56$      | 6                 | 5.6       | 93.33          |
| Proposed NA | $1 \times 1$             | $-28.22$      | 13.3              | 12.4      | 93.23          |

References

1. Alù A, Engheta N (2010) Wireless at the nanoscale: optical interconnects using matched NAs. *Phys Rev Lett* 104:213902

2. Zeng YS et al (2017) All-plasmonic optical phased array integrated on a thin-film platform. *Sci Rep* 7:9959

3. Kullock R, Ochs M, Grimm P, Emmerling M, Hecht B (2020) Electrically driven Yagi-Uda antennas for light. *Nat Commun* 11:115

4. de Souza JL, da Costa KQ (2018) Broadband wireless optical nanolink composed by dipole-loop nanoantennas. In: *IEEE Photon J* 10(2):1–8. Art no. 2200608. <https://doi.org/10.1109/JPHOT.2018.2810842>

5. Kwon SJ et al (2018) Extremely stable graphene electrodes doped with macromolecular acid. *Nat Commun* 9:2037

6. Moshiri SMM, Nozhat N (2021) Smart optical cross dipole nanoantenna with multibeam pattern. *Sci Rep* 11:5047. <https://doi.org/10.1038/s41598-021-84495-0>

7. Rui G, Zhan Q (2014) Highly sensitive beam steering with plasmonic antenna. *Sci Rep* 4:5962. <https://doi.org/10.1038/srep05962>

8. Llatser I, Kremers C, Chigrin D, Jornet JM, LemmeMC, CabellosAparicio A, Alarcón E (2012) Radiation characteristics of tunable graphene in the terahertz band. *Radioeng J* 21(4)

9. Liatser I et al (2012) Graphene-based nano-patch antenna for terahertz radiation. *Photon Nanostruct Fundament Appl* 10:353–358

10. GusyninV SS, Carbotte J (2006) Magneto-optical conductivity in graphene. *J Phys Condens Matter* 19(1–28):026222

11. Zeng C, Liu X, Wang G (2014) Electrically tunable graphene plasmonic quasicrystal metasurfaces for transformation optics. *Sci Rep* 4(5763):1–8
12. Chen PY, Alu A (2011) Atomically thin surface cloak using graphene monolayers. *ACS Nano* 5:5855–5863
13. Lu Z, Zhao W (2012) Nanoscale electro-optic modulator based on graphene slot waveguides. *J Opt Soc Am B* 29:1490–1496
14. Knight MW et al (2014) Aluminum for plasmonics. *ACS Nano* 8(1):834–840
15. Grischkowsky D, Duling IN III, Chen TC, Chi C-C (1987) Electromagnetic shock waves from transmission lines. *Phys Rev Lett* 59(15):1663–1666
16. Yousefi L, Foster AC (2012) Waveguide-fed optical hybrid plasmonic patch nano-antenna. *Opt Express* 20(16):18326–18335
17. Nikoufard M, Nourmohamadi N, Esmaili S (2018) Hybrid plasmonic nanoantenna with the capability of monolithic integration with laser and photodetector on InP substrate. *IEEE Trans Antennas Propag* 66(1):3–8
18. Sethi WT, Vettikalladi H, Fathallah H (2015) Dielectric resonator nanoantenna at optical frequencies. In: 2015 international conference on information and communication technology research (ICTRC), pp 132–135. <https://doi.org/10.1109/ICTRC.2015.7156439>
19. Baranwal AK, Pathak NP (2019) Enhanced gain triangular patch nanoantenna using hybrid plasmonic waveguide on SOI technology. In: 2019 IEEE Indian conference on antennas and propagation (InCAP), pp 1–3. <https://doi.org/10.1109/InCAP47789.2019.9134564>
20. Nourmohammadi A, Nikoufard M (2020) Ultra-wideband photonic hybrid plasmonic horn nanoantenna with SOI configuration. *SILICON* 12:193–198. <https://doi.org/10.1007/s12633-019-00113-9>
21. Saad-Bin-Alam M, Khalil MI, Rahman A, Chowdhury AM (2015) Hybrid plasmonic waveguide fed broadband nanoantenna for nanophotonic applications. *IEEE Photon Technol Lett* 27(10):1092–1095
22. Sethi W, de Sagazan O, Vettikalladi H, Fathallah H, Himdi M (2018) Yagi-Uda antenna for 1550 nanometers optical communication systems. *Microw Opt Technol Lett* 60(9):2236–2242. Available: <https://doi.org/10.1002/mop.31339>
23. Dregely D, Taubert R, Dorfmueller J et al (2011) 3D optical Yagi-Uda nanoantenna array. *Nat Commun* 2:267. <https://doi.org/10.1038/ncomms1268>

# Chapter 10

## Effectiveness of Transformer Models on IoT Security Detection in StackOverflow Discussions



Nibir Chandra Mandal, G. M. Shahariar, and Md. Tanvir Rouf Shawon

**Abstract** The Internet of Things (IoT) is an emerging concept that directly links to the billions of physical items, or “things” that are connected to the Internet and are all gathering and exchanging information between devices and systems. However, IoT devices were not built with security in mind, which might lead to security vulnerabilities in a multi-device system. Traditionally, we investigated IoT issues by polling IoT developers and specialists. This technique, however, is not scalable since surveying all IoT developers is not feasible. Another way to look into IoT issues is to look at IoT developer discussions on major online development forums like Stack Overflow (SO). However, finding discussions that are relevant to IoT issues is challenging since they are frequently not categorized with IoT-related terms. In this paper, we present the “IoT Security Dataset”, a domain-specific dataset of 7147 samples focused solely on IoT security discussions. As there are no automated tools to label these samples, we manually labeled them. We further employed multiple transformer models to automatically detect security discussions. Through rigorous investigations, we found that IoT security discussions are different and more complex than traditional security discussions. We demonstrated a considerable performance loss (up to 44%) of transformer models on cross-domain datasets when we transferred knowledge from a general-purpose dataset “Opiner”, supporting our claim. Thus, we built a domain-specific IoT security detector with an F1-Score of 0.69. We have made the dataset public in the hope that developers would learn more about the security discussion and vendors would enhance their concerns about product security. The dataset can be found at—<https://anonymous.4open.science/r/IoT-Security-Dataset-8E35>.

**Keywords** IoT · IoT security · Transformers · StackOverflow · Machine learning

---

G. M. Shahariar and Md. Tanvir Rouf Shawon are contributed equally to this research.

---

N. C. Mandal (✉) · G. M. Shahariar · Md. T. R. Shawon  
Ahsanullah University of Science and Technology, Dhaka, Bangladesh  
e-mail: [nibir338@gmail.com](mailto:nibir338@gmail.com)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_10](https://doi.org/10.1007/978-981-19-7528-8_10)

125

# 1 Introduction

The Internet of Things (IoT) is a widespread connection of intelligent devices that are bridged through the internet and are rapidly expanding to every corner of the globe. The number of IoT devices will touch a 36.8 billion margin by 2025, which is 107% higher than in 2020 [1]. Because IoT devices carry a lot of information about their users, the growing number of IoT devices presents a lot of security concerns for developers. An unintentional intrusion into the device might cause significant harm to the users. As a result, today's developers are paying close attention to this scenario and attempting to identify the best solution for protecting devices from attackers. In numerous developer communities, there is a lot of discussion about the security of IoT devices. One of the communities that developers use to address security solutions is Stack Overflow (SO). The SO community is quite active and has a lot of IoT security discussions. A great number of works have been done in this research field [2–4]. In this work, we have tried to demonstrate a comprehensive analysis of determining whether an aspect of a discussion is an IoT security aspect or not. The purpose of this study is to assist developers in learning more about security issues in the IoT space, where there is a lot of discussion about various aspects. Furthermore, with the aid of our efforts, vendors may increase the security of their products. We have basically employed some of the popular transformer models like BERT [5], RoBERTa [6], XLNet [7] and BERTOverflow [8] in 3 of our different experiments. For our initial approach, we used a benchmark dataset called Opiner [9], which recorded a variety of aspects of discussions such as performance, usability, security, community, and so on. The field of discussion was also diverse. Instead of a general-purpose dataset, we worked to create a domain-specific dataset focused solely on IoT security issues, which we named the “IoT Security Dataset”. Following the creation of the dataset, we attempted to transfer the knowledge gained from the Opiner dataset to our own IoT dataset, but the results were unsatisfactory. Then we decided to fine-tune the pretrained transformer models using our own dataset, and we achieved a decent result. From these experiments, we discovered that knowledge transfer is not a good technique to get a decent outcome. The best way to identify security issues in developer discussions is to conduct domain-specific experiments using domain-specific data. In summary, we have made the following main contributions in this paper:

- We have created a domain-specific dataset called “IoT Security Dataset” that focuses on security aspects of IoT-related textual developer discussions.
- We further employed multiple transformer models to automatically detect security discussions. Through rigorous investigations, we found that IoT security discussions are different and more complex than traditional security discussions.
- We demonstrated a considerable performance loss (up to 44%) of transformer models on cross-domain datasets when we transferred knowledge from a general-purpose dataset “Opiner”, supporting our claim.

## 2 Background Study

### 2.1 Transformers

A neural network that learns factors in sequential data is referred to as a transformer model. A transformer model provide us with an word embedding which is used as the input of a fully connected neural network for further classification task. In natural language processing transformer models are used frequently as it tracks the meaning of consecutive data by tracking the relationships among them like the words in a sentence. Transformer models were first introduced by Google [10] and are one of the most powerful models till date in the history of computer science. We incorporated four different kinds of transformer models in our work which are RoBERTa, BERT, XLNET and BERTOverflow. Bidirectional Encoder Representations from Transformers or BERT [5] was first established by Jacob Devlin et al. which is a transformer model to represent language. It was modeled in such way that anyone can fine-tune it with a supplementary layer to create new models to solve a wide range of tasks. Yinhan Liu et al. introduced RoBERTa [6] as a replication study of BERT [5]. Authors showed that their model was highly trained than BERT overcoming the limitations and showed a good performance over basic BERT model. Zhilin Yang et al. introduced XLNet [7] which overcomes the constraints of BERT [5] using a universal autoregressive pre-training technique. Maximum expected likelihood was taken into consideration over all arrangements of the factorization order. Authors showed that their model beat BERT by a huge margin on a variety of tasks like sentiment analysis, language inference or document ranking etc. BERTOverflow [8] by Jeniya Tabassum et al. is a kind of transformer model trained for identifying code tokens or software oriented entities inside the area of natural language sentences. Authors introduced a NER corpus of total 15,372 sentences for the domain of computer programming.

### 2.2 Evaluation Matrices

Any kind of model requires some measurement criteria to understand its' efficiency. In this project 4 different kinds of performance matrices were used which are **Precision**, **Recall**, **F1-Score** and **Area Under Curve (AUC)**. The percentage of accurately predicted labels that are literally correct is calculated by a model's **precision** score. It also indicates the caliber of a model to predict the labels correctly. The model's ability to reliably discover positives among existing positives is measured by its **recall** score. We can also state it as a model's capacity to locate all relevant cases within a data set. The precision and recall score are used to generate the **F1-Score**. The F1-score is a metric in which Precision and Recall play an equal role in evaluating the model's performance. A graphical illustration of how effectively binary classifiers work is the Receiver Operating Characteristic (ROC) curve. **(AUC)** score

is calculated using ROC and is a very useful tool for calculating a model's efficiency in an uneven dataset.

### 2.3 Cross Validation

A reshuffling process for assess various algorithms and models on a little amount of data is known as Cross-validation. It comprises an attribute,  $k$ , denoting the quantity of sets a provided sample of data should be partitioned. That is why, the technique is widely known as  $k$ -fold cross-validation. When we give a value to the parameter  $k$ , it means we want to partition the whole data into  $k$  sets.

## 3 The IoT Security Dataset

In this section, we will present details of the IoT security dataset creation process. The whole process is divided into three subprocesses: (1) Collect IoT posts (2) Extract sentences from IoT posts and (3) Label extracted sentences. Each process is discussed in the following subsections.

### 3.1 IoT Posts Collection

As IoT tags do not cover all IoT-related posts, we applied three steps to collect IoT-related posts.

**Step (1) Download the SO dataset:** We used the September 2021 Stack Overflow (SO) data dump, which was the most recent dump accessible at the time of research. The following meta data may be found in each post: (1) *text data and code samples*; (2) *creation and edition times*; (3) *score, favorite, and view counts*; (4) *user information of the post creator*; and (5) *if the post is a question, tags given by the user*. If the user who posted the question marked the response as approved, the answer is flagged as accepted. A question might include anything from one to five tags. All postings from 2008 to September 2021 are included in the SO dataset.

**Step (2) Identify IoT Tagset:** Because not all of the posts in the SO dataset are linked to the IoT, we had to figure out which ones had IoT-related topics. We established the set of tags that may be used to label a discussion as IoT-related using the user-defined tags attached to the questions. We followed the two-step method described in [11]: (1) In SO, we discovered three prominent IoT-related tags. (2) We gathered postings with those three initial tags and studied the tags that were applied to them. There are 78 tags in the final tag collection. We have discussed each step in detail below.

**(a) Identify initial tags:** Using the Stack Overflow (SO) search engine, the three initial tags were chosen. We began our search by looking for queries in SO that were labeled with “iot”, and the SO search engine returned posts that were tagged with “iot” as well as a set of 25 additional tags that were related to these topics, such as “raspberry-pi”, “arduino”, “windows-10-iot-core”, “python”, and so on. In our initial set, we thus considered the following three tags: (a) “iot” or any tag including the term “iot”, such as “windows-10-iot-core”; (b) “arduino”, and (c) “raspberry-pi”.

**(b) Determine final tagset:** To begin, we looked for IoT-related posts using the three initial tags. Second, we retrieved all questions from the SO dataset that were tagged with one of the initial three tags. Finally, we collected all of the tags from the question set. It is possible that not all of the tags in the questions set match to IoT subjects (for example, “python”). As a result, the significance and relevance specified in [11] for each tag were computed to pick filtered and finalized IoT-related tags in the questions set. If a tag’s significance and relevance values exceed certain criteria, it is considered significantly relevant to the IoT. After comprehensive testing, we discovered 78 IoT-related tags. Please see [11] for a more complete discussion of the tagset identification technique.

**Step (3) Collect IoT Posts:** All posts labeled with at least one of the selected 78 tags make up our final dataset. If a SO question is labeled with one or more tags from the final tagset, it is an IoT question. In our SO dump, we discovered a total of 40 K posts based on the 78 tags.

### 3.2 *Sentence Level Dataset Creation*

As security discussions can be buried inside a post, So we followed sentence-level analysis to get a better understanding of the security discussion. We got the sentences as follows: As our main focus is textual analysis, we avoided security discussion inside code snippets and urls. We used the BeautifulSoup library to extract all that extra information such as titles, codes, and urls. We also used the NLTK sentence tokenizer to get all the sentences from those 40 K posts. Finally, we found around 200 K sentences.

### 3.3 *Benchmark Dataset Creation*

As there are not any automated tools to label this large data set, we chose to label them manually. Due to the data size and sparsity, we labeled a statistically significant portion of the data. For our experiment, we randomly selected 7147 (99% confidence with a confidence interval of 1.5) sentences and labeled them. Without knowing each other, two developers labeled this data first. In addition, a third developer was introduced to resolve the conflict. We labeled each sentence by majority



**Table 1** Security data distribution for opiner and IoT security datasets

| Dataset | Size | Security   | Kappa | Agreement score |
|---------|------|------------|-------|-----------------|
| Opiner  | 4522 | 163 (3.6%) | –     | –               |
| IoT     | 7147 | 250 (3.5%) | 0.92  | 98.5%           |

voting. For example, if both developers agreed, we labeled them with the same label they agreed on. Otherwise, we took into account the third developer’s decision. All these developers have industrial experience of more than 2 years and have enough knowledge of the domain. The summary of our dataset can be seen here in Table 1.

## 4 Opiner Dataset

Opiner dataset was created by Uddin et al. [12] to study developers opinion in SO for different APIs. The dataset includes developer discussions in form of text and related aspects of that discussions. Each opinion may have multiple aspects such as “Performance”, “Usability”. One of those aspects is “Security”. In this research, we only took consideration of this aspect. We discarded other aspects and labeled all security aspect related samples as 1 and other samples as 0. We found a total of 163 samples are security-related.

## 5 Methodology

The proposed system for aspect categorization is presented in this section. The key phases of the proposed method for binary aspect classification (*considering usability aspect as an example*) are summarized in Fig. 1 and are further detailed below.

**Step (1) Input Sentence:** Each raw sentence from the dataset is presented to the proposed model one by one for further processing. Before that each sentence goes through some pre-processing steps: all urls and codes are removed.

**Step (2) Tokenization:** Each processed sentence is tokenized using the *BERT Tokenizer*. Each tokenized sentence gets a length of 100 tokens and zero padded when required. In the event of length more than 100, we cut off after 100. The output of this step is a tokenized sentence of size 100.

**Step (3) Embedding:** For word embedding, BERT [5] is used to turn each token in a sentence into a numeric value representation. Each token is embedded by 768 real values via BERT. The input to this step is a tokenized sentence of size 100 and output of this step is an embedded sentence of size  $100 * 768$ .

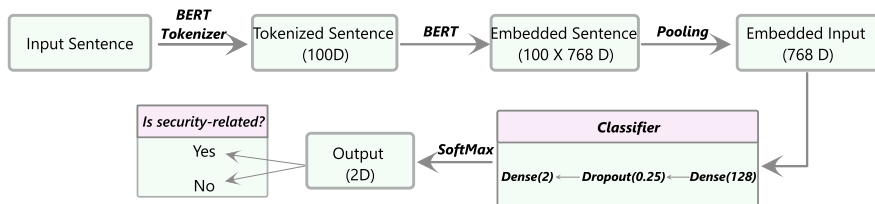


**Fig. 1** Aspects classification process

**Step (4) Pooling:** To reduce the dimension of the feature map ( $100 * 768$ ) for each tokenized sentence in step 3, max pooling is used which provides a real valued vector representation of size 768 per sentence.

**Step (5) Classification:** Aspect classification is accomplished by the use of transfer learning [13]. To classify security and non security aspects, a basic neural network with two dense layers (with 128 and 2 neurons) is utilized to fine-tune the four pre-trained transformer models one by one such as RoBERTa [6], BERT [5], DistilBERT [14], and XLNET [7]. The pretrained weights were initialized using *glorot\_uniform* in the simple neural network. A dropout rate of 0.25 is employed between the dense layers. Finally, we apply a “Softmax” activation to get the prediction.

## 6 Experimental Results

### 6.1 Experiments

At first, we found that Uddin et al.’s Logistic Regression (Logits) model [12] outperformed other rule-based models. Thus, we took this Logits as our benchmark model. In recent times, deep models, specifically pretrained models, have shown a huge improvement over both shallow machine learning models such as SVM and Logistics Regression and classic deep models such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM). Therefore, we decided to use the Transformer models. We applied these already trained models on Opiner dataset to test the IoT dataset which we have prepared for this study to check whether a general purpose dataset can give a good generalization performance for a domain-specific dataset. We did not get a satisfactory result from this experiment as the generalization is not a good idea. Finally, we added domain-specific IoT security data during the tuning of the pretrained models and found acceptable results from the experiment. The performances of all the models on opiner dataset are shown in Table 2. **The result of logistic regression is taken from the experiments [9] of Gias Uddin et al. on the benchmark Opiner dataset.** A comparison of performances among all the 3 experiments on Cross-validation on Opiner dataset, Cross-Domain dataset (Trained on Opiner and tested on IoT dataset)

**Table 2** Performance of 4 different transformer model on *Opiner* dataset

| Type            | Model        | Precision   | Recall      | F1-score    | AUC         |
|-----------------|--------------|-------------|-------------|-------------|-------------|
| Proposed models | BERT         | 0.72        | 0.79        | 0.74        | 0.89        |
|                 | RoBERTa      | <b>0.86</b> | 0.76        | <b>0.79</b> | <b>0.92</b> |
|                 | XLNet        | 0.72        | <b>0.84</b> | 0.77        | 0.91        |
|                 | BERTOverflow | 0.79        | 0.76        | 0.76        | 0.88        |
| Previous work   | Logits [12]  | 0.77        | 0.57        | 0.60        | 0.69        |

and Cross-validation on IoT Dataset for all the transformer models we used are shown in Fig. 2.

## 6.2 Result Analysis

First, We experimented with three generic transformers, BERT, RoBERTa and XLNet, as well as one domain-specific model, BERTOverflow. As these robust pre-trained models can capture underlying implicit contexts, the performance of these models significantly improved over shallow models, although the size of the dataset is comparatively low. We found that the lowest performing transformer model has 23% better F1-Score than the baseline models.

We found that RoBERTa model has the best performance in terms of F1-Score among other transformers. Despite having a low recall, RoBERTa achieves the highest F1-Score of 0.79, followed by XLNet (0.77), BERTOverflow (0.76) and BERT (0.74). This indicates that RoBERTa is the most precise model, but it has lower discoverability. Thus, the generic purpose transformer RoBERTa outperforms both the domain-specific BERTOverflow and other generic purpose models. This happens for two reasons. *First*, RoBERTa is the most optimized transformer model. Due to this optimization, RoBERTa gets an extra edge to outperform its base model, BERT, for our task. *Second*, security discussions in the Opiner dataset are more generic than domain-specific. Although the traditional rule-based approach fails to guess the security discussions as correctly as deep models like the transformer model, SO domain knowledge makes the prediction more intriguing. XLNet has enough insights to identify security discussions but it suffers from precision. As example, we found the following sentences in Opiner dataset: ‘*For instance, messages sent by the client, may be digitally signed by the applet, with the keys being managed on per-user USB drives (containing the private keys).*’ It is clearly visible that the word ‘signed’ is a pivotal security-related word in this sentence. As our SO domain knowledge specific model BERTOverflow is trained on both programming and non-programming data, the model finds ambiguity in such a scenario and thus makes erroneous predictions. However, RoBERTa handles the scenario in a better way.

We found that transformers improved the performance of baseline models by a long way for security aspect detection. This motivated us to dive deeper into security aspects. However, we found that the performance significantly dropped when the transformers trained on Opiner security data were tested on the IoT dataset. We investigated this drop in performance and found that security discussions in the IoT dataset are more implicit compared to the generic security discussion in Opiner. Moreover, the domain knowledge that we need to discern those security discussions is missing in the generic Opiner dataset. For instance, the following sentence includes the IoT domain-specific keyword RFID which is an identification key: ‘Is there any way to restart the screen after using the RFID reader?’. As such discussions are only available in IoT discussions, transformer models that are created and trained on only for general purposes fail to perceive the security context. As a result, the model underperforms in the IoT domain. The performance drop in F1-Score for all the transformers can be seen in Fig. 3. We can clearly see from the charts that the performance of **BERT**, **RoBERTa**, **XLNET** and **BERTOverflow** has dropped by **28%**, **32%**, **43%** and **44%** respectively after testing our own IoT dataset with the transformers trained on Opiner dataset.

As our previous experiment result analysis denotes that the model is missing IoT contextual insights, we included IoT discussions in our dataset to check how these discussions influence the performance. We observed that adding domain-specific data during the tuning of the pretrained model resulted in more robust performance. We found that the performance improved over 50% in terms of F1-score. Like our previous findings, the RoBERTa model has the best F1-Score of 0.69, followed by BERTOverflow (0.66), BERT (0.64), and XLNet (0.54). In addition, RoBERTa also shows the best precision of 0.73 and recall of 0.66. This again shows that RoBERTa is the most effective model for security aspect detection in any domain.

We compared this result with the results of previous experiments. We have showed bar-charts in Fig. 2 for all these four models. We found that all models have lower precision than previous experimental results. However, the models had improved their recall compared to experiment 2 but still lagged behind experiment 1. As a result, the models have a higher F1-Score than experiment 2 but lower than experiment 1. For example, RoBERTa has a 46% better F1-Score than experiment 2, but 13% lower than experiment 1. Based on this observation, we came to two conclusions regarding the detection of IoT security aspects. ***First, IoT security is different from the general type of security we discuss more often.*** We conclude this based on the evidence that the same transformers that can detect security aspects reliably (Experiment-1 results) fail to perform similarly in the IoT domain (Experiment-2 results), but the performance improves while domain knowledge is added to the model (Experiment-3 results). ***Second, IoT security aspects are more complex, sparse, and implicit than normal security aspects.*** The basis of this conclusion is the evidence that even if the transformer models are adopted to IoT domain knowledge, the models fail to perform as well as they do for general security aspect detection (Experiment-3 results). Thus, we believe by incorporating more IoT knowledge (i.e., increasing training samples) the performance of the models can be further improved, which we left as our future work.

## 7 Related Work

Several recent papers have used SO posts to investigate various aspects of software development using topic modeling, such as what developers are talking about in general [2], or about a specific problem like big data [15], concurrency [3], security [4, 16], and trends in block chain reference architectures [17]. All of these projects demonstrate that interest in the Internet of Things is growing, and conversations about it are becoming more common in online developer communities like Stack Overflow (SO). Analyzing the presence, popularity and complexity of certain IoT issues may be gained through understanding these discussions. Uddin et al. [11] investigated at nearly 53,000 IoT-related posts on SO and used topic modeling [18] to figure out what people were talking about. On SO, Aly et al. [19] addressed at questions of IoT and Industry 4.0. They utilized topic modeling to identify the themes mentioned in the investigated questions, similar to the work given by Uddin et al. [11]. Their study concentrated on the industrial issues of IoT technology, whereas Uddin et al. [11] intended to learn about the practical difficulties that developers experience when building real IoT systems. Recent studies [20, 21] explored the connection between API usage and Stack Overflow discussions. Both research discovered a relationship between API class use and the number of Stack Overflow questions answered. But Gias Uddin [12] utilized their constructed benchmark dataset named “OPINER” [12] to carry out the study and noticed that developers frequently provided opinions about vastly different API aspects in those discussions which was the first step toward filling

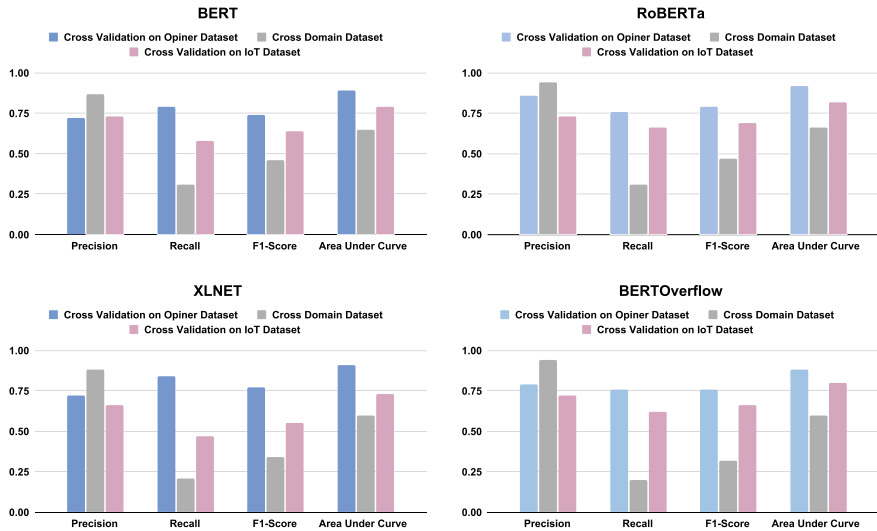


**Fig. 2** A comparison of performance among all the 3 experiments for BERT (top left), RoBERTa (top right), XLNET (lower left) and BERTOverflow (lower right) models

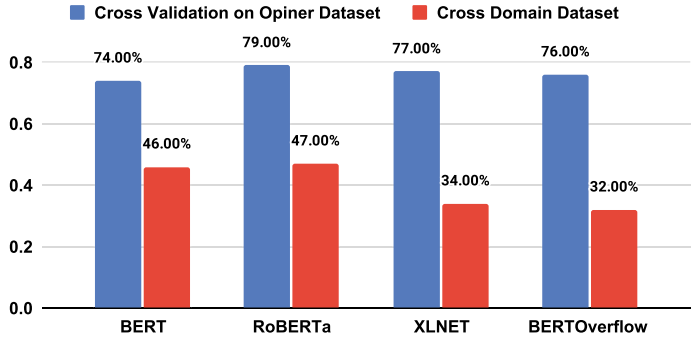


**Fig. 3** A comparison of **F1-Score** between the experiments on cross-validation on opiner dataset and cross-domain dataset

the gap of investigating the susceptibility and influence of sentiments and API aspects in the API reviews of online forum discussions. Uddin and Khomh [12] introduced OPINER, a method for mining API-related opinions and providing users with a rapid summary of the benefits and drawbacks of APIs when deciding which API to employ to implement a certain feature. Uddin and Khomh [22] used an SVM-based aspect classifier and a modified Sentiment Orientation algorithm [23] to comply with API opinion mining. Based on the positive and negative results emphasized in earlier attempts to automatically mine API opinions, as well as the seminal work in this field by Uddin and Khomh [22], Lin et al. [24] introduced a new approach called Pattern-based Opinion MinEr (POME), which utilizes linguistic patterns preserved in Stack Overflow sentences referring to APIs to classify whether a sentence is referring to a specific API aspect (functional, community, performance, usability, documentation, compatibility or reliability), and has a positive or negative polarity.

## 8 Conclusion and Future Works

We found identifying security aspects in IoT-related discussions to be a difficult task since domain-specific datasets on security-related discussions are not commonly available. In our study, we attempted to create a one-of-a-kind dataset in this respect, and we presented a brief comparison between our IoT security dataset experiments and the benchmark dataset on aspect identification. We have come to the conclusion that generalization is not really the best method for identifying security discussions on sites like StackOverflow. Domain-specific knowledge transfer via various transformer models might be a superior alternative to security aspect detection. In the future, we can incorporate other transfer learning methods to improve our performance. The results we obtained are not quite satisfactory as our dataset is not very large. Increasing the number of samples in the dataset is another effort we may undertake in the future to enhance the outcome.

## References

1. Industrial IOT connections to reach 37 billion globally by 2025, as 'smart factory' concept realised. Accessed 9 Apr 2022
2. Barua A, Thomas SW, Hassan AE (2012) What are developers talking about? An analysis of topics and trends in stack overflow. In: Empirical software engineering, pp 1–31
3. Ahmed S, Bagherzadeh M (2018) What do concurrency developers ask about?: A largescale study using stack overflow. In: Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement, vol 30
4. Yang X-L, Lo D, Xia X, Wan Z-Y, Sun J-L (2016) What security questions do developers ask? A large-scale study of stack overflow posts. *J Comput Sci Technol* 31(5):910–924
5. Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
6. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692)
7. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) XLNet: generalized autoregressive pretraining for language understanding. In: Advances in neural information processing systems, vol 32
8. Tabassum J, Maddela M, Xu W, Ritter A (2020) Code and named entity recognition in stack-overflow. arXiv preprint [arXiv:2005.01634](https://arxiv.org/abs/2005.01634)
9. Uddin G, Khomh F (2017) Automatic summarization of API reviews. In: 2017 32nd IEEE/ACM international conference on automated software engineering (ASE). IEEE, pp 159–170
10. Ashish V, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30
11. Uddin G, Sabir F, Guéhenec Y-G, Alam O, Khomh F (2021) An empirical study of IoT topics in IoT developer discussions on stack overflow. *Empirical Softw Eng* 26(6):1–45
12. Uddin G, Khomh F (2017) Mining API aspects in API reviews. Technical report
13. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2019) Exploring the limits of transfer learning with a unified text-to-text transformer
14. Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint [arXiv:1910.01108](https://arxiv.org/abs/1910.01108)
15. Bagherzadeh M, Khatchadourian R (2019) Going big: a large-scale study on what big data developers ask. In: Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering. ESEC/FSE 2019. New York, NY, USA, pp 432–442
16. Mandal N, Uddin G (2022) An empirical study of IoT security aspects at sentence-level in developer textual discussions. *Inf Softw Technol* 150:106970. <https://doi.org/10.1016/j.infsof.2022.106970>
17. Wan Z, Xia X, Hassan AE (2019) What is discussed about blockchain? A case study on the use of balanced LDA and the reference architecture of a domain to capture online discussions about blockchain platforms across the stack exchange communities. *IEEE Trans Softw Eng*
18. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J Mach Learn Res* 3(4–5):993–1022
19. Aly M, Khomh F, Yacout S (2021) What do practitioners discuss about IoT and industry 4.0 related technologies? Characterization and identification of IoT and industry 4.0 categories in stack overflow discussions. *Internet Things* 14:100364
20. Kavalier D, Posnett D, Gibler C, Chen H, Devanbu P, Filkov V (2013) Using and asking: APIs used in the android market and asked about in stackoverflow. In: Proceedings of the international conference on social informatics, pp 405–418
21. Parnin C, Treude C, Grammel L, Storey M-A (2012) Crowd documentation: exploring the coverage and dynamics of API discussions on stack overflow. Technical report, technical report GIT-CS-12-05, Georgia Technology

22. Uddin G, Khomh F (2017) Opiner: an opinion search and Gias Uddin and Foutse Khomh. "Opiner: an opinion search and summarization engine for APIs". In: Proceedings of the 32nd IEEE/ACM international conference on automated software engineering (ASE 2017). IEEE Computer Society, pp 978–983
23. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining (SIGKDD 2004). ACM, pp 168–177
24. Lin B, Zampetti F, Bavota G, Di Penta M, Lanza M (2019) Pattern-based mining of opinions in Q&A websites. In: 2019 IEEE/ACM 41st international conference on software engineering (ICSE). IEEE, pp 548–559



# Chapter 11

## Digital Health Inclusion to Achieve Universal Health Coverage in the Era of 4th Industrial Revolution



**Farhana Sarker, Moinul H. Chowdhury, Rony Chowdhury Ripan, Tanvir Islam, Rubaiyat Alim Hridhee, A. K. M. Nazmul Islam, and Khondaker A. Mamun**

**Abstract** The lack of proper healthcare facilities, resource constraints, and a non-functional referral system hinder Bangladesh's healthcare system offering comprehensive primary and preventive healthcare (PPH) services. To address these issues, cloud-based medical system framework (CMED) health created the digital general practitioner (GP) model for the rural people of Bangladesh with digital health account and structured referral mechanism. In this study, we introduce this digital GP model integrated with digital platforms to ensure PPH with referrals for rural Bangladesh. Overall, the digital GP model consists of applications for users, health workers, GP doctors, and management, for service delivery and monitoring. By utilizing this digital GP model, rural people can get regular doorstep health checkups, track their health conditions, take necessary steps to prevent diseases in early stages, reduce their out-of-pocket expenditure, and consult with GP doctors through telemedicine or

---

F. Sarker · M. H. Chowdhury · R. C. Ripan · T. Islam · R. A. Hridhee · A. K. M. N. Islam · K. A. Mamun (✉)

CMED Health, Dhaka, Bangladesh  
e-mail: [mamun@cse.uiu.ac.bd](mailto:mamun@cse.uiu.ac.bd)

F. Sarker  
e-mail: [farhana.sarker@ulab.edu.bd](mailto:farhana.sarker@ulab.edu.bd)

M. H. Chowdhury  
e-mail: [moin@cmehhealth.com](mailto:moin@cmehhealth.com)

R. C. Ripan  
e-mail: [u1404090@student.cuet.ac.bd](mailto:u1404090@student.cuet.ac.bd)

T. Islam  
e-mail: [tanvir.islam@cmehhealth.com](mailto:tanvir.islam@cmehhealth.com)

A. K. M. N. Islam  
e-mail: [nazmul@cmehhealth.com](mailto:nazmul@cmehhealth.com)

F. Sarker  
Department of CSE, University of Liberal Arts Bangladesh, Dhaka, Bangladesh

K. A. Mamun  
AIMS Lab, Department of CSE, United International University, Dhaka, Bangladesh

physical visit. During the pilot project, this digital GP model served a total of 11,890 people, consisting of 4752 men (39.97%) and 7138 women (60.03%). From our data analysis, 438 (7.02%) adult people were suspected of having diabetes among 6239 blood sugar measurement, 164 (2.56%) adult people were found obese among 6383 BMI measurement, and 1991 (19.41%) adult people were suspected of hypertension among 10,260 blood pressure measurement. In addition, among young people, females were “underweight” at a higher percentage (72, or 22.71%) than males (52, or 20.72%) among 568 BMI measurement. Finally, the digital GP concept is a great method for the government to implement digital health inclusion and move closer to universal health coverage.

**Keywords** Digital GP model · ICT in healthcare · Digital health referral system

## 1 Introduction

A general practitioner (GP) is a doctor who offers primary healthcare for a variety of chronic conditions, administers preventive, and primary care within a catchment area and sends patients to a hospital or specialist following risk assessment [1]. Numerous developed countries have introduced GP models, which have increased the effectiveness of their health systems and reduced the strain on hospitals for basic healthcare concerns. Primary care was developed in the early twentieth century in the United Kingdom, stressing the notion of referral [1]. The United States of America pioneered the physician assistant job in 1960 to address a physician shortage [2]. Norway has a solid primary care system, and its citizens have relied on general practitioners as their primary care physicians since 2001 [3]. In Australia, general practitioners treat common ailments, chronic diseases, and diabetes, as well as administering vaccinations. They offered “telehealth” services during the COVID epidemic [4]. According to research conducted in England, around 15–40% of patients who go to emergency departments (EDs) for treatment may be treated by their primary care physician (GP). Additionally, this research aided in the implementation of a GPED paradigm in which primary care and emergency medicine are merged to provide better care to a larger population [5].

Numerous industrialized nations are attempting to digitalize their health systems to boost the value of co-creation between physicians and patients via the use of a digital health system (DHS). For example, Presti et al. demonstrated via a multiple linear regression analysis that online involvement had a direct impact on the social sustainability of the “paginemediche.it” (an Italian DHS system) and hence had a favorable effect on physician loyalty [6]. Additionally, Rosis et al. discovered that patients’ reactions, trust, and happiness with DHS are quite good [7].

The majority of Bangladesh’s lands (about 61.82% [8]) are rural. Due to a lack of healthcare infrastructure at the rural level, as well as a shortage of competent general practitioners and health workers, Bangladesh’s health industry faces several obstacles. In metropolitan areas, there are 18.2 doctors, 5.8 nurses, and 0.8 dentists per

10,000 inhabitants, whereas rural areas have 1.1, 0.8, and 0.08 physicians, respectively [9]. In addition, according to World Bank report, Bangladesh's Health Spending Per Capita (HSPC) is 123\$, Govt. Health Expense Per C. is 22.9\$, and out-of-pocket expense is 72% [10]. Whereas, Global Health Spending Per Capita (HSPC) is 1467\$, Govt. Health Expense Per C. is 865\$, and out-of-pocket expense is 18% [10]. This shows the scarcity of healthcare facility and suffering of people in Bangladesh.

CMED Health [11] designed and implemented a digital GP model for rural areas to provide comprehensive preventive and primary healthcare, named as the "Rural General Practitioner" (RGP) model, following an examination of the existing healthcare system, the digital health system, and the rural health service situation in Bangladesh. By the help of this digital health inclusion, individuals can access primary care on-demand through CMED's digital health kits and smartphone apps, which are maintained by professional health workers. If health workers identify a risk to the patient's health using the clinical decision support system (CDSS), they either refer the patient to an integrated GP center or facilitate telemedicine services on their phone, where doctors provide additional intervention and digitally prescribe the patient. To achieve universal health coverage, this digital GP model is designed in such a way that rural people can get all the services of this digital GP model by only spending Bangladeshi 100 taka (US 1.20\$) monthly. To experiment this system, CMED health piloted their digital GP model in Nayanagar union of Melandaha upazila in the Jamalpur district, a rural area from Mymensingh division on 26th March 2021. The main objectives of this study are (a) to introduce digital GP model for rural Bangladesh and (b) to outline key findings based on measurement data that are collected from the piloted digital GP model. The rest of the paper is designed according to the following: Sect. 2 provides a comprehensive overview of the digital GP model for rural Bangladesh. We analyze all the measurement data and outline the findings in Sect. 3. In Sect. 4, we discuss our findings and finally conclude this paper in Sect. 5.

## 2 Design and Development of Digital GP Model

### 2.1 System Overview

CMED has developed a digital GP model for rural Bangladesh with comprehensive preventive and primary healthcare delivery utilizing 4th industrial revolution technologies such as IoT-enabled devices and cloud-based platforms. This model provides services in three segments which are health workers (paramedic), telemedicine services, and the GP center. Health workers provide doorstep primary health services using smart health devices and a mobile application and collect various socio-demographic data and health-related data. They can also refer people to telemedicine or the GP center for further intervention if necessary. The overview of the digital GP system is illustrated in Fig. 1.

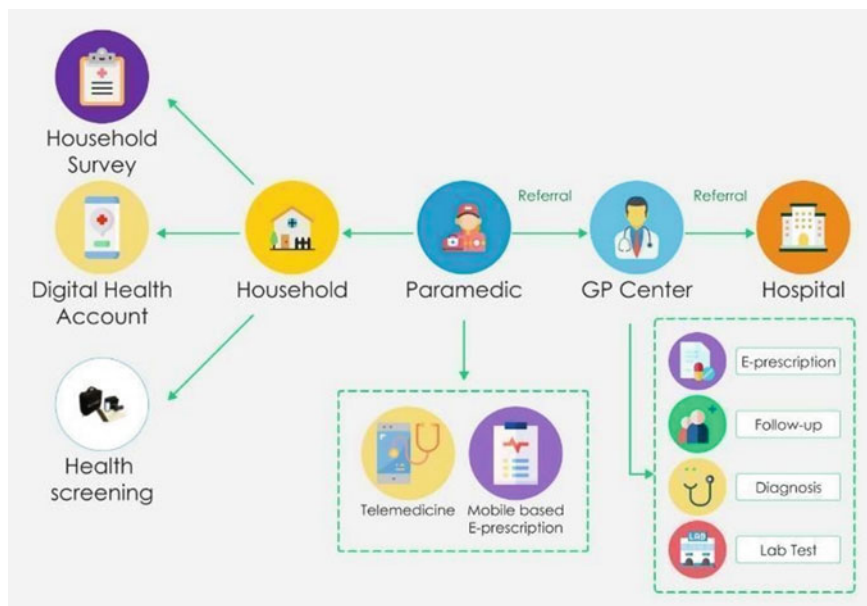


**Fig. 1** Overview of digital GP model

**Health Workers and Doorstep Services.** Health workers provide preventive and primary healthcare services at doorsteps and take socio-demographic and health-related historical data. They use IoT-enabled devices to measure diabetes, blood pressure, pulse rate, weight, and SpO<sub>2</sub> (oxygen saturation). These devices connect with the mobile devices via Bluetooth and automatically insert the data into the mobile application while taking the measurement. Health workers also take heights, body mass index (BMI), temperature, mid upper arm circumference (MUAC) measurements, and inputs into the mobile application manually. The overview of digital GP model services and smart health devices is illustrated in Fig. 2. After performing health checkups of a patient, the system executes risk assessment using CDSS and creates a suggestion of referral if a patient needs to see a doctor. Then, a health worker refers a patient to telemedicine or GP center.

**GP Center.** GP center comes with several facilities such as doctors and lab tests. GP doctors use the GP doctor Web dashboard to view patient health records and provide e-prescriptions or printed prescriptions. Also, patients can do several medical tests such as ECG, blood, and urine in the GP center laboratory for further diagnosis.

**Health Worker Application.** This application's first user interface is a sign up or log in system for health workers. After logging in to the system, they get to see a dashboard that has 3 sections: (1) "household visit and healthcare", (2) "meeting", (3) "referred history". By clicking "referred history", they can see all the patients they have referred so far. They can view all registered households by clicking the

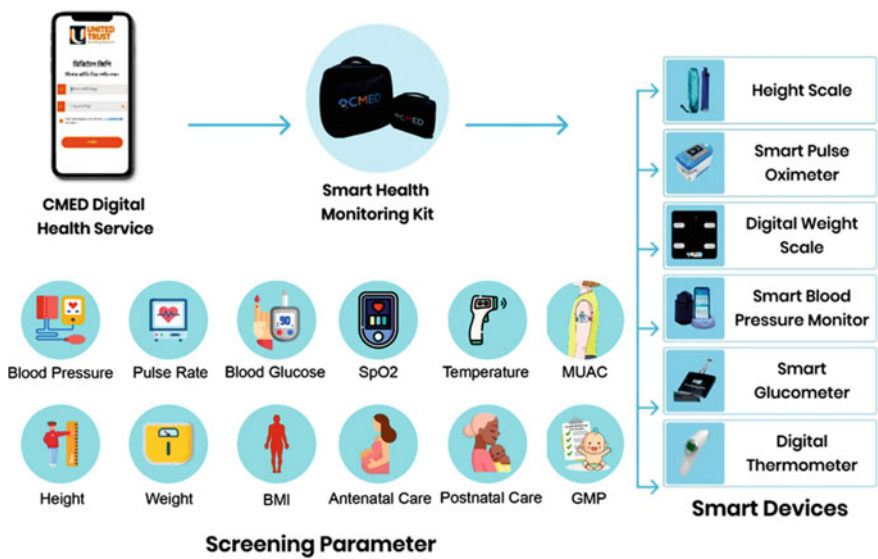


Fig. 2 Services and smart health devices of the digital GP model

“household visit and healthcare” button, and they can also create new households by clicking the “plus” button in the bottom-right corner of the screen. Each household comes with three tabs: (1) “members”, (2) “surveys”, (3) “mothers and child”. When health workers tap on a particular member, they get to see three different tabs: (1) “measurements”, (2) “prescription”, (3) “lab report”. The “measurement” tab displays all that member’s measurement history (sorted by date from most recent to least recent). In addition to that there is also a button at the bottom-right corner of the screen.

When they click on that they are taken to a measurement dashboard with five measurement features and one telemedicine feature. They can see all the digital prescriptions and all the lab reports that member has received through the GP system in the “prescription” tab. Since it is hard to get Internet connections in rural areas, health workers can save all the data offline and then sync all the data back to the cloud by using the “sync” feature, which is in the side menu bar. Some main features of this application are represented in Fig. 3.

**User Application.** When a user logs into the application, the first section, the user can see his or her name, phone number, age, etc., as shown in Fig. 4a. Then in the second section, they can see their gender, blood group, diabetic status, measurements, etc. Then in the third section, there is the “health screening” feature, with which the user can give measurement inputs manually or if they have access to CMED smart health kits. By the “my health” feature, they can see a graphical overview of their health, and by “my health records”, they can see all the previous health records like prescriptions and lab reports, etc., and even download them. A menu bar is located at the bottom of

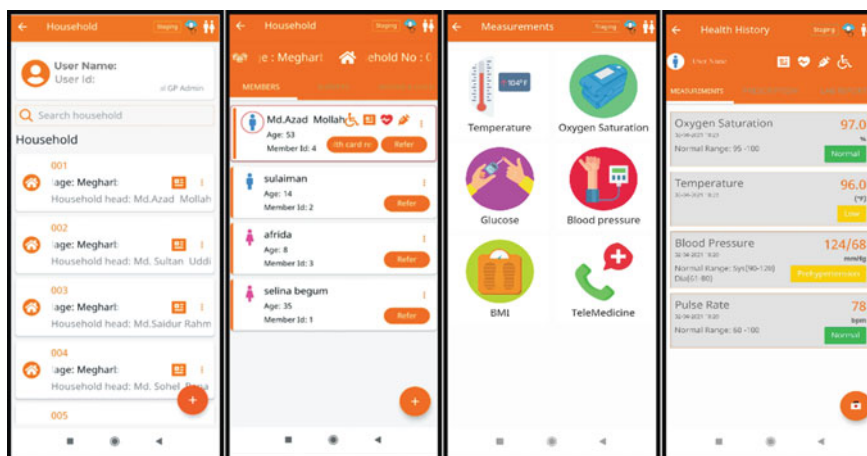


Fig. 3 Health worker's mobile application

the main dashboard to navigate different sections such as “home”, “article”, “vitals”, “report”, and “search”. By article section, users can learn about different preventive and primary healthcare measures that they should take by reading several articles related to healthcare. Through the “report” section, they could download reports of BP, pulse rate, etc., if they measured them in the GP center. They can use “search” to find the nearest hospitals, ATMs, blood banks, and pharmacies. There is also a side bar which includes some basic features like “my account”, “logout”, etc.

**GP Doctor Dashboard.** Only GP doctors have access to the GP doctor dashboard. There is a menu bar which is located on the left side of the dashboard, and it comes with three sections: (1) “home”, (2) “patient list”, (3) “lab report list”. At first glance, on the “home” dashboard, doctors can see the total served patient number, the total served patient number that day, how many patients are waiting, and so on. Also, at the bottom of the dashboard, the doctor can access all the waiting patients’ lists, referral dates, etc. When they click on a particular patient’s name, they can see a dashboard as shown in Fig. 5. On the upper side of this dashboard, they get to see all the details like name, gender, last visit date, etc. In addition, on the upper right-hand side, they can see all the NCDs a particular patient has in picture coded. On the left side, they can store patients’ complaints, comorbidity, etc., see the previous drug history, gynecological history, and can give diagnosis lab tests. In the middle section of the dashboard, they can give all the medicine, medicine dosage, and instructions for each drug and provide advice and suggestions. All the above is later converted into a digital prescription.

**Admin Dashboard.** The admin’s dashboard is created to control, monitor, and provide quality assurance of the digital GP model. When admin logs in to the dashboard, at first glance, they can see all the organizations they are partnered with, how many unions they are serving, how many doctors they have, how many health officers

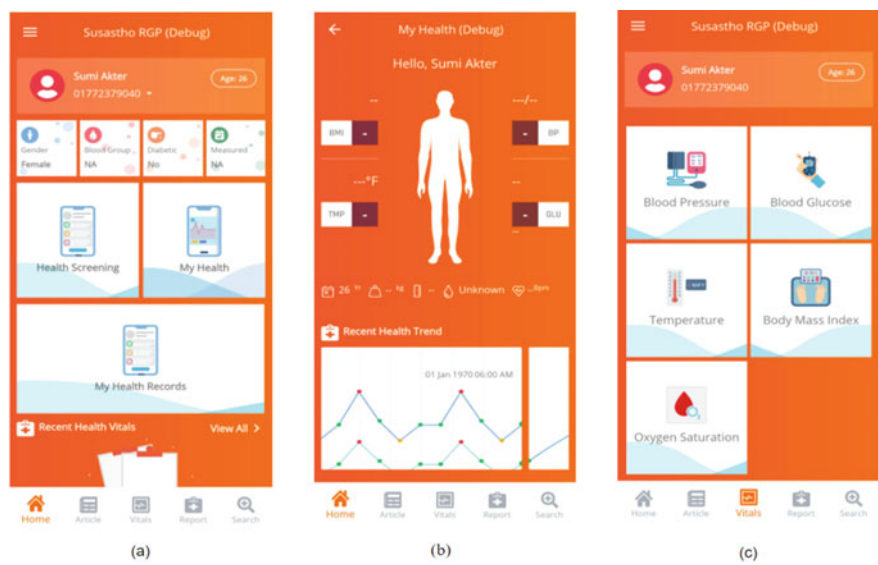


Fig. 4 User application

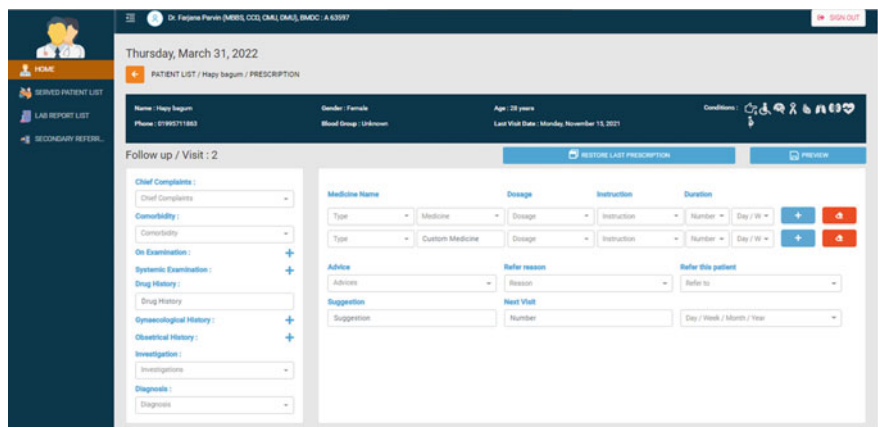


Fig. 5 GP doctor dashboard

they have, total number and percentage of households surveyed, total number and percentage of members served, total number of cards members bought, etc. Also, by scrolling down, they can see several visualization charts of socio-demographic data and health-related data. A simple overview of the admin dashboard can be observed in Fig. 6.



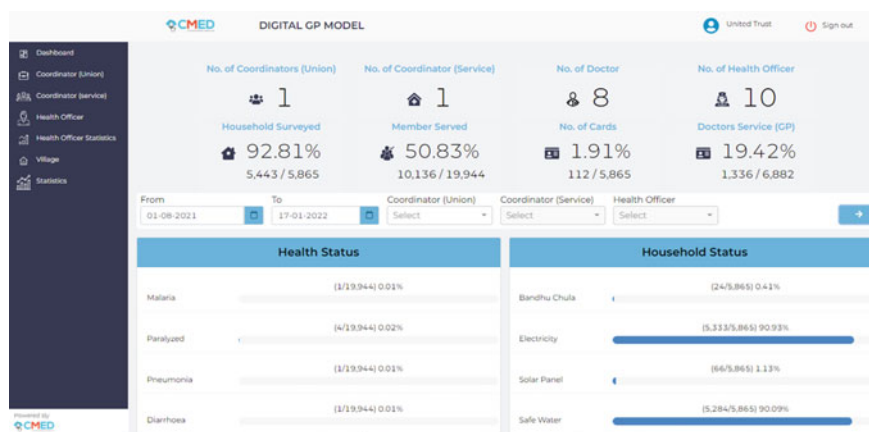


**Fig. 6** Admin dashboard

### 3 Analysis of Collected Data

The pilot project of this model launched on 26th March 2021 in Nayanagar, a union of Melandaha Upazila in Jamalpur district, Mymensingh and ended on 31st December 2021. The health measurements collected during the pilot are analyzed and segregated into age categories, gender, and severity. This section elaborates the outcomes and highlights important findings. Table 1 illustrates the distribution of measured people segregated into different age groups and gender. During the pilot project, the digital system served 4752 (39.97%) male and 7138 (60.03%) female among 11,890 people. From this table, we can observe that our digital GP system has reached to all age class people from child to old. In addition to that, female group is measured around 60.03% compared to the male group. It is because health workers usually visit during the day when most of the male members usually go for work. Also, it is observed that 30–39 age groups were more prevalent than other age groups. In addition, most of the females were from age group 18–29, whereas most of the males were from age group 30–39.

Table 2 represents distribution of overall measured population over age 18 based on blood pressure (BP), BMI, SpO<sub>2</sub>, and blood sugar. From this table, it is observed that most of the people in Nayanagar union were healthy. However, among 6239 blood sugar measurement, 438 (7.02%) people were suspected of having diabetes, and 162 (2.59%) people were in high blood sugar who already had diabetes. Besides, 164 (2.56%) people were found obese among 6383 BMI measurement, and 1991 (19.41%) people were suspected of hypertension among 10,260 blood pressure measurement.

Table 3 represents distribution of overall measured population under age 18 based on blood pressure (BP), BMI, SpO<sub>2</sub>, and blood sugar. It is observed from table that among 1202 people, 12 people was measured blood sugar, 568 people



**Table 1** Population distribution segregated into age and gender

| Age groups | No. of people |               |               |
|------------|---------------|---------------|---------------|
|            | Male          | Female        | Total         |
| Total      | 4752 (39.97%) | 7138 (60.03%) | 11,890        |
| <18        | 579 (4.87%)   | 623 (5.24%)   | 1202 (10.11%) |
| 18–29      | 667 (5.61%)   | 1675 (14.09%) | 2342 (19.7%)  |
| 30–39      | 946 (7.96%)   | 1674 (14.08%) | 2620 (22.04%) |
| 40–49      | 839 (7.06%)   | 1137 (9.56%)  | 1976 (16.62%) |
| 50–59      | 752 (6.32%)   | 1070 (9%)     | 1822 (15.32%) |
| 60–69      | 599 (5.04%)   | 650 (5.47%)   | 1249 (10.5%)  |
| ≥70        | 370 (3.11%)   | 309 (2.6%)    | 679 (5.71%)   |

was measured BMI, 186 people was measured blood pressure, and 183 people was measured MUAC. From the MUAC measurement, 26 (14.21%) people were facing severe malnutrition, and among BMI measurement 62 (10.92%), people were found to be obese. Even though the female population is larger in BMI measurement, 40 (15.29%) males were obese, significantly more than their female counterparts (22, or 6.94%). However, females were “underweight” at a greater rate (72, or 22.71%) than males (52, or 20.72%). Besides, in terms of MUAC measurement, females (18, or 20.69%) were more severe malnourished than males (8, or 8.33%).

## 4 Discussion

Overall, at first, we describe the digital GP model for rural Bangladesh concept and digital platforms that incorporating digital GP model. By utilizing this digital GP model, rural people can get constant doorstep health checkups, track their health records, take necessary steps to prevent diseases in early stages, reduce their out-of-pocket expenditure, consult with GP doctors through telemedicine and so on. All these services are maintained using four digital platforms (health worker application, user application, GP doctor dashboard, and admin dashboard). During the pilot project, the digital system served 4752 (39.97%) male and 7138 (60.03%) female among 11,890 people. In addition to that, female group is measured around 60.03% compared to the male group. It is because health workers usually visit during the day when most of the male members usually go for work. From Table 2, it is observed that among blood sugar measurement, 438 (7.02%) people were suspected of having diabetes, 164 (2.56%) people were found obese among BMI measurement, and 1991 (19.41%) people were suspected of hypertension. Besides, from Table 3, it is observed that females were underweight at a greater rate (72, or 22.71%) than males (52, or 20.72%). Besides, in terms of MUAC measurement, females (18, or 20.69%) were more severe malnourished than males (8, or 8.33%). We claim that this

**Table 2** Distribution of population over age 18 based on different health measurements

| Characteristics                  | No. of people |               |               |
|----------------------------------|---------------|---------------|---------------|
|                                  | Male          | Female        | Total         |
| Total population (age $\geq$ 18) | 4173 (39.04%) | 6515 (60.96%) | 10,688        |
| Blood sugar                      | 2026 (32.47%) | 4213 (67.53%) | 6239          |
| Low (hypoglycemia)               | 8 (0.39%)     | 25 (0.59%)    | 33 (0.53%)    |
| Normal                           | 1771 (87.41%) | 3769 (89.46%) | 5540 (88.8%)  |
| Pre-diabetic                     | 32 (1.58%)    | 34 (0.81%)    | 66 (1.06%)    |
| Diabetic (need confirmation)     | 143 (7.06%)   | 295 (7.00%)   | 438 (7.02%)   |
| HIGH (borderline)                | 13 (0.64%)    | 11 (0.26%)    | 24 (0.38%)    |
| High                             | 59 (2.91%)    | 79 (1.88%)    | 138 (2.21%)   |
| BMI                              | 1949 (30.53%) | 4434 (69.47%) | 6383          |
| Underweight                      | 341 (17.5%)   | 726 (16.37%)  | 1067 (16.72%) |
| Normal                           | 1367 (70.14%) | 2979 (67.19%) | 4346 (68.09%) |
| Overweight                       | 220 (11.29%)  | 599 (13.51%)  | 819 (12.83%)  |
| Obesity                          | 24 (1.23%)    | 114 (2.57%)   | 138 (2.16%)   |
| Highly Obesity                   | 2 (0.1%)      | 18 (0.41%)    | 20 (0.31%)    |
| Morbid obesity                   | 0 (0%)        | 6 (0.14%)     | 6 (0.09%)     |
| Blood pressure                   | 3929 (38.29%) | 6331 (61.71%) | 10,260        |
| Low                              | 315 (8.02%)   | 638 (10.08%)  | 953 (9.29%)   |
| Normal                           | 1416 (36.04%) | 2614 (41.29%) | 4030 (39.28%) |
| Prehypertension                  | 1433 (36.47%) | 1853 (29.27%) | 3286 (32.03%) |
| Mild high                        | 508(12.93%)   | 803 (12.68%)  | 1311 (12.78%) |
| Moderate high                    | 187 (4.76%)   | 320 (5.05%)   | 507 (4.94%)   |
| High                             | 1 (0.03%)     | 1 (0.02%)     | 2 (0.02%)     |
| Severe high                      | 69 (1.76%)    | 102 (1.61%)   | 171 (1.67%)   |
| Pulse rate                       | 3929 (38.29%) | 6331 (61.71%) | 10,260        |
| Low                              | 183 (4.66%)   | 107 (1.69%)   | 290 (2.83%)   |
| Normal                           | 3524 (89.69%) | 5541 (87.52%) | 9065 (88.35%) |
| High                             | 222 (5.65%)   | 683 (10.79%)  | 905 (8.82%)   |
| SpO <sub>2</sub>                 | 996 (37.4%)   | 1667 (62.6%)  | 2663          |
| Very low                         | 18 (1.81%)    | 21 (1.26%)    | 39 (1.46%)    |
| Low                              | 70 (7.03%)    | 47 (2.82%)    | 117 (4.39%)   |
| Normal                           | 908 (91.16%)  | 1599 (95.92%) | 2507 (94.14%) |

**Table 3** Distribution of population under age 18 based on different health measurements

| Characteristics             | No. of people |              |              |
|-----------------------------|---------------|--------------|--------------|
|                             | Male          | Female       | Total        |
| Total population (age < 18) | 579 (48.17%)  | 623 (51.83%) | 1202         |
| Blood sugar                 | 5 (25.00%)    | 15 (75.00%)  | 20           |
| Low                         | 1 (20.00%)    | 0 (0%)       | 1 (5.00%)    |
| Normal                      | 3 (60.00%)    | 14 (93.33%)  | 17 (85.00%)  |
| High                        | 1 (20.00%)    | 1 (6.67%)    | 2 (10.00%)   |
| BMI                         | 251 (44.19%)  | 317 (55.81%) | 568          |
| Underweight                 | 52 (20.72%)   | 72 (22.71%)  | 124 (21.83%) |
| Normal                      | 146 (58.17%)  | 195 (61.51%) | 341 (60.04%) |
| Overweight                  | 15 (5.98%)    | 28 (8.83%)   | 43 (7.57%)   |
| Obesity                     | 40 (15.94%)   | 22 (6.94%)   | 62 (10.92%)  |
| Blood pressure              | 70 (37.63%)   | 116 (62.37%) | 186          |
| Low                         | 25 (35.71%)   | 58 (50.00%)  | 83 (44.62%)  |
| Normal                      | 32 (45.71%)   | 45 (38.79%)  | 77 (41.4%)   |
| High                        | 13 (18.57%)   | 13 (11.21%)  | 26 (13.98%)  |
| MUAC                        | 96 (52.46%)   | 87 (47.54%)  | 183          |
| Normal                      | 79 (82.29%)   | 57 (65.52%)  | 136 (74.32%) |
| Moderate                    | 9 (9.38%)     | 12 (13.79%)  | 21 (11.48%)  |
| Severe                      | 8 (8.33%)     | 18 (20.69%)  | 26 (14.21%)  |
| Pulse rate                  | 70 (37.63%)   | 116 (62.37%) | 186          |
| Low                         | 3 (4.29%)     | 1 (0.86%)    | 4 (2.15%)    |
| Normal                      | 60 (85.71%)   | 107 (92.24%) | 167 (89.78%) |
| High                        | 7 (10.00%)    | 8 (6.9%)     | 15 (8.06%)   |
| SpO2                        | 299 (49.1%)   | 310 (50.9%)  | 609          |
| Very low                    | 12 (4.01%)    | 17 (5.48%)   | 29 (4.76%)   |
| Low                         | 10 (3.34%)    | 11 (3.55%)   | 21 (3.45%)   |
| Normal                      | 277 (92.64%)  | 282 (90.97%) | 559 (91.79%) |

discrimination occurs because of rural people's propensity for providing more food to male family members than to female family members [12]. In addition, we have several limitations such as female studied population were more than male population. So, our results could be skewed toward female population. Also, the digital GP model only implemented in one union. So, our findings cannot be interpreted for whole rural scenario.

## 5 Conclusion and Future Work

The people in Bangladesh are still far away from receiving the desired healthcare benefits. Due to the lack of effective monitoring and evaluation system, it is difficult to ensure service availability and delivery in the community clinics. As there is no structured referral system, most of the rural patients directly visit Upazila Health Complexes and/or Private Clinics/Hospitals. This increases the patient load massively on Upazila Health Complex and increases out-of-pocket expenditure for the citizens as well. Through digital health inclusion, the digital GP model ensures a structured healthcare referral system that helps people to receive appropriate medical services with preventive and primary care. By utilizing and adapting this model, most diseases can be identified at an early stage. As a result, people do not have to go to higher-level health facilities for primary healthcare services thus out-of-pocket expenditure can be reduced. The digital GP model has the potential to bring revolutionary changes in the healthcare sector in Bangladesh and scaling up the model across the country will help achieve SDG goals and realize the dream of digital Bangladesh and achieve universal health coverage. In future, CMED proposes to integrate the digital GP model with Bangladesh's existing healthcare infrastructures like Community Clinics and Upazila Health Complexes to accelerate the adaptation process of this model throughout the country. Also, the user app will have a data-driven symptom checker which will advise a person and give proper directions based on the symptoms of that person.

## References

1. Horder J (1983) The origin of the Royal College of general practitioners. *Soc Soc Hist Med Bull (Lond)* 33:13–14
2. Malone R (2021) The role of the physician associate: an overview. *Ir J Med Sci* 0123456789. <https://doi.org/10.1007/s11845-021-02661-9>
3. Gronseth IM, Malterud K, Nilsen S (2020) Why do doctors in Norway choose general practice and remain there? A qualitative study about motivational experiences. *Scand J Prim Health Care* 38(2):184–191. <https://doi.org/10.1080/02813432.2020.1753348>
4. Healthcare Direct (2018) The role of a GP | healthdirect [Online]. Available: <https://www.healthdirect.gov.au/the-role-of-a-gp>
5. Brant H et al (2021) Current provision of general practitioner services in or alongside emergency departments in England. *Emerg Med J* 38(10):780–783. <https://doi.org/10.1136/emmed-2020-210539>
6. Lo Presti L, Testa M, Marino V, Singer P (2019) Engagement in healthcare systems: adopting digital tools for a sustainable approach. *Sustainability* 11(1):1–15. <https://doi.org/10.3390/su11010220>
7. De Rosis S, Barsanti S (2016) Patient satisfaction, e-health and the evolution of the patient–general practitioner relationship: evidence from an Italian survey. *Health Policy (New York)* 120(11):1279–1292. <https://doi.org/10.1016/j.healthpol.2016.09.012>
8. Bangladesh: share of rural population 2020 | Statista. <https://www.statista.com/statistics/760934/bangladesh-share-of-rural-population/>
9. Khandakar MSA (2014) Rural health care system and patients' satisfaction towards medical care in Bangladesh: an empirical Study. *J Bus* 35(2)

10. Current health expenditure (% of GDP)—Bangladesh | Data. <https://data.worldbank.org/indicator/SH.XPD.CHEX.GD.ZS?locations=BD>
11. Sailunaz K, Alhussein M, Shahiduzzaman M, Anowar F, Al Mamun KA (2016) CMED: cloud based medical system framework for rural health monitoring in developing countries. *Comput Electr Eng* 53:469–481. <https://doi.org/10.1016/j.compeleceng.2016.02.005>
12. Choudhury KK, Hanifi MA, Rasheed S, Bhuiya A (2000) Gender inequality and severe malnutrition among children in a remote rural area of Bangladesh. *J Health Popul Nutr* 18(3):123–130. PMID: 11262764

# Chapter 12

## Design of $8 \times 8$ Microstrip Array Antenna for ISM and K-Band Applications



Prodip Kumar Saha Purnendu, Afrin Binte Anwar, Rima Islam, Debalina Mollik, Md. Anwar Hossain, and Mohiuddin Ahmad 

**Abstract** An  $8 \times 8$  microstrip array antenna is designed to maintain proper spacing in each antenna element where the frequency is under the ISM band and Ka-band. Designing and fabrication of many antennas can be expensive as well as complicated. This research work introduces three different operating along with two bands ISM and K-band which ensures that it can adapt to 5G communications technology and global applications. The proposed antenna can operate at (22.99/24.24/26.35) GHz with bandwidths of 0.287 GHz, 0.327 GHz, and 0.767 GHz. Previously, several microstrip antennas were designed, but here, it is shown that this antenna has a comparatively narrower beam width and also contains higher gain, better return loss, and VSWR values, respectively, for each bandwidth. For 22.4 GHz, 24 GHz, and 26 GHz, gain values are 16.81 dB, 18.41 dB, and 12.38 dB, respectively. S11 values are  $-22.65$  dB,  $-23.543$  dB, and  $-24.58$  dB for following operating frequencies 22.4 GHz, 24 GHz, and 26 GHz.

**Keywords** Bandwidth · Efficiency · Gain · Narrow beamwidth · Triple band · ISM and K-band

## 1 Introduction

In wireless communication, the introduction of the microstrip patch antenna is a turning point; since it continued to fulfill emerging needs that will apply to next-generation antenna technology [1]. Among other configurations of microstrip

---

P. K. S. Purnendu · A. B. Anwar (✉) · R. Islam · D. Mollik  
American International University-Bangladesh (AIUB), Dhaka, Bangladesh  
e-mail: [afrinanwar144@gmail.com](mailto:afrinanwar144@gmail.com)

Md. A. Hossain  
NEMEMW & TC, MoH & FW, Dhaka, Bangladesh  
e-mail: [anwar@bme.kuet.ac.bd](mailto:anwar@bme.kuet.ac.bd)

M. Ahmad  
Department of EEE, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [ahmad@eee.kuet.ac.bd](mailto:ahmad@eee.kuet.ac.bd)

patches, the microstrip array has the advantages of high gain and directivity. Over a century ago, the first antenna array was constructed. Two vertical antennas had been implemented which were separated by half a wavelength. To improve the directivity of a single monopole, antennas were fed out of phase and he discovered the improved directivity [2]. Since several kinds of research are going on development of array. With increased power in the antenna array, longer-range point-to-point targeted applications can perform [3]. Microwave-induced hyperthermia for medicinal applications is another application of focused antennas. It is preferable to enhance the power deposition in the antenna's near-field to heat malignant tissue while avoiding heating healthy tissues proximate to the tumor. It is required to lower the size of the microstrip patch antenna for increasing the bandwidth across a wide range for an advanced communication engineering system since it is the most important need in practical applications [4]. It is required to lower the size of the microstrip patch antenna for increasing the bandwidth across a wide range for an advanced communication engineering system since it is the most important need in practical applications. This sort of array antenna is necessary for low cost, simple array fabrication, and polarization diversity [5]. An array antenna is defined as a specific antenna formed up of several connections and configurations in a regular layout. The major benefit of an array antenna is that it may enhance total gain while also maximizing the signal to interference plus noise ratio (SINR). As it has pointed to point connection, this sort of antenna can also filter interference from a specific set of directions, making it most sensitive in a particular direction [6]. The range of the ISM band is 24–24.25 GHz, which has been licensed for global application, and the K-band (18–26 GHz), which will be used in future wireless communication networking systems and modern technologies [7, 8]. ISM band (24–24.25 GHz) is the worldwide band that can be used for armature satellites and earth exploration satellites [7].

This paper has discussed the triple band and simulation results of array antenna for 5G wireless communications. Operating in three individual frequencies with ISM and K-band can be easier than fabricating separate antennas for numerous technologies. It also compares results and data with recent research works and shows the outcomes of different parameters of the antenna in CST software.

## 2 Design Procedure of Array Antenna

The ground material is copper, Roger RT5880 (lossy) is chosen as substrate with permittivity 2.2, and the patch material is copper. The  $8 \times 8$  array antenna design was chosen for its high directivity and gain. For avoiding grating lobes the separate radiations, each antenna element is about  $1.3\lambda$  apart. Figure 1 depicts the planned antenna's geometric structure in free space, and Fig. 2 shows the space between two array elements.

The formulas required to calculate parameters are listed below [9]. The width and the effective length of the antenna's array are determined using the calculations of Eqs. (1) and (2).

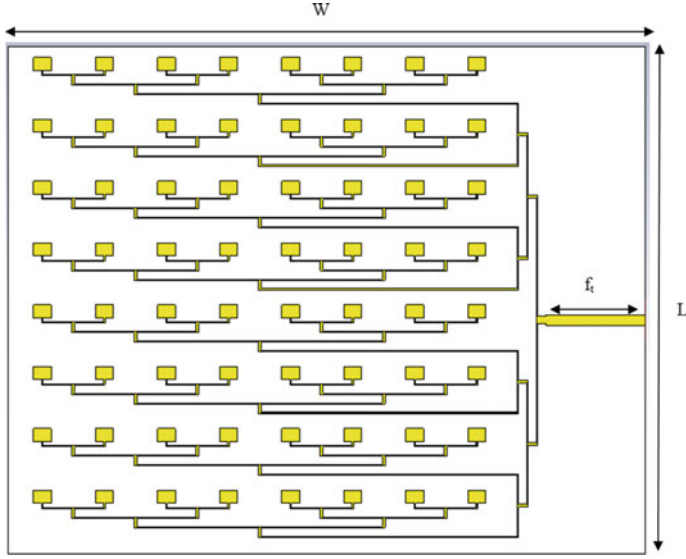


**Fig. 1** Prospective  $8 \times 8$  array antenna's structure in free space

**Fig. 2** Space between two array elements



$$W = \frac{c_0}{2f_r \sqrt{\frac{\epsilon_r + 1}{2}}} \quad (1)$$

$$L_{\text{eff}} = \frac{c_0}{2f_r \sqrt{\epsilon_{\text{reff}}}} \quad (2)$$

where  $L_{\text{eff}}$  = Effective length and the effective permittivity,  $\epsilon_{\text{reff}}$  is calculated by the following Eq. (3).

$$\epsilon_{\text{reff}} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left( 1 + 12 \frac{h}{w} \right)^{-0.5} \quad (3)$$

The extension length,  $\Delta L$ , and the length of the patch are calculated by Eqs. (4) and (5), respectively.

$$\Delta L = 0.412 \frac{\left( \frac{w}{h} + 0.264 \right) (\epsilon_{\text{reff}} + 0.3)}{(\epsilon_{\text{reff}} - 0.258) \left( \frac{w}{h} + 0.813 \right)} \quad (4)$$

$$L = L_{\text{eff}} - 2\Delta L \quad (5)$$



**Table 1** Parameters of proposed antenna's model

| Model's parameters and symbols                   | Dimension (mm) |
|--------------------------------------------------|----------------|
| Patch's width ( $W_p$ )                          | 4.236          |
| Patch's length ( $L_p$ )                         | 3.063          |
| Width of the ground and substrate plane ( $W$ )  | 150            |
| Length of the ground and substrate plane ( $L$ ) | 120            |
| Height of substrate ( $S_h$ )                    | 0.787          |
| Feedline width ( $f_i$ )                         | 2.37           |
| Ground thickness ( $h_i$ )                       | 0.0175         |

For feeding, the antenna element corporate is chosen. Firstly, the power divider is calculated theoretically; then, a power divider layout circuit has been created to supply power to the antenna element. At the load end of the power divider circuit, the impedance was  $144 \Omega$ . Because two antenna elements are in parallel, the corresponding resistance at the center is  $72 \Omega$ . Then, to convert  $72 \Omega$  impedance back to  $144 \Omega$  impedance, a quarter-wave transformer is selected. The procedure is then performed several times more to complete the power divider. The corresponding  $72 \Omega$  impedance is then reduced to  $50 \Omega$  using another quarter-wave transformer to convert this power divider into a  $50 \Omega$  transmission line. Because of its high directivity, this antenna consumes relatively minimal power. The proposed microstrip patch antennas-related parameters are listed in Table 1.

### 3 Results Analysis and Discussions

#### 3.1 Return Loss

The acceptable range of the  $S_{1,1}$  parameter is  $-10$  dB where about 90% of power is transmitted in the antenna. Here at 22.96, 24.24, and 26.35 GHz, the  $S_{1,1}$  value is  $-22.65$ ,  $-23.543$ , and  $-24.58$  dB which are appropriate for transmission (Fig. 3).

#### 3.2 Voltage Standing Wave Ratio (VSWR)

The ideal value of VSWR is 1. Here in the simulation result at 22.996 GHz, 24.244 GHz, and 26.35 GHz, VSWR is 1.159, 1.142, and 1.12 which are very close to standard values (Fig. 4).

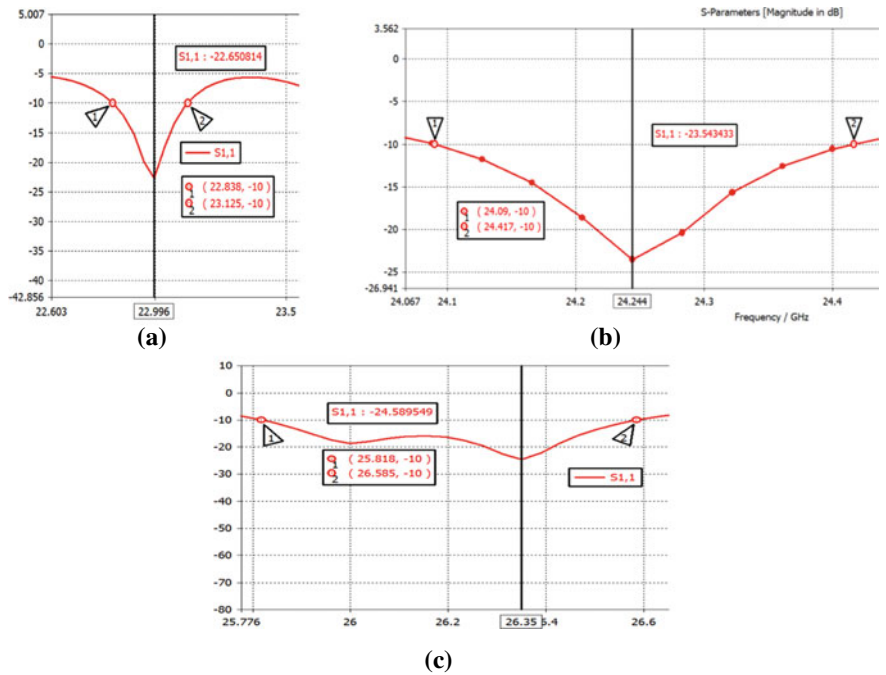


**Fig. 3** S<sub>1,1</sub> and bandwidth of respective frequency **a** at 22.966 GHz, **b** at 24.244 GHz, and **c** at 26.35 GHz

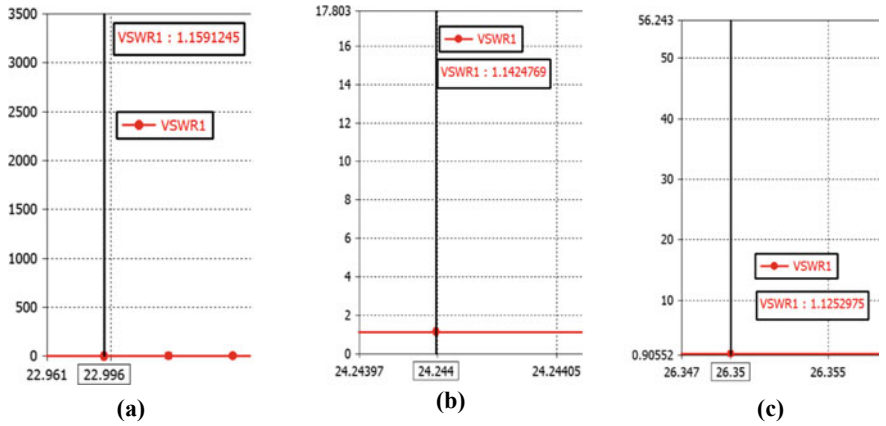


**Fig. 4** VSWR versus frequency **a** 22.996 GHz, **b** at 24.244 GHz, and **c** 26.35 GHz

3.3 Radiation Pattern 3D

Gain is the performance parameter of the antenna. It shows how much input radio power is radiating in a specific direction. In this proposed antenna at 22.996 GHz, 24.244 GHz, and 26.35 GHz, gain is 16.81 dB, 18.41 dB, and 12.38 dB (Figs. 5, 6 and 7; Table 2).

**Fig. 5** Radiation-pattern in three dimensions at frequency 22.996 GHz (directivity) (in free space)

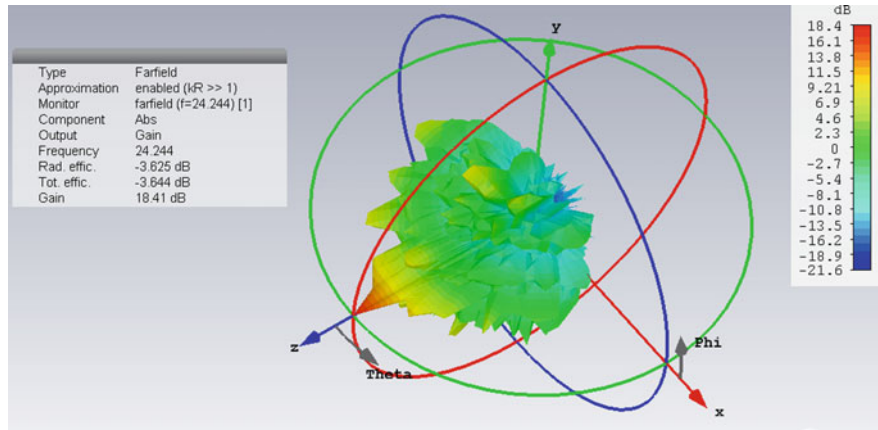
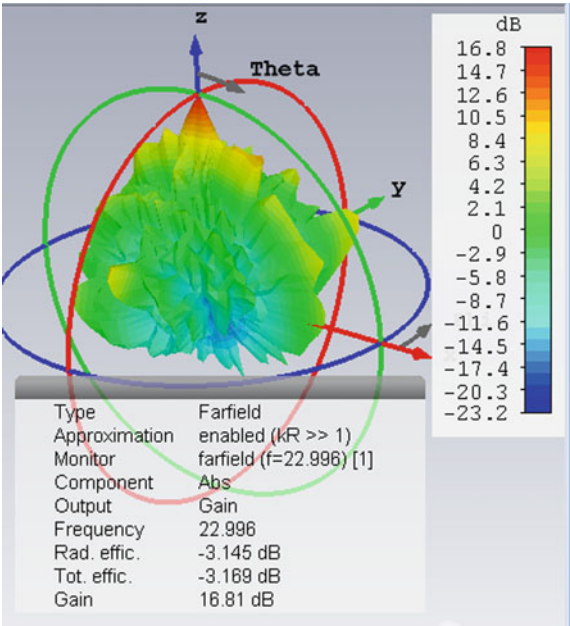


**Fig. 6** 3D view of gain of the designed antenna for frequency 24.244 GHz (in free space)

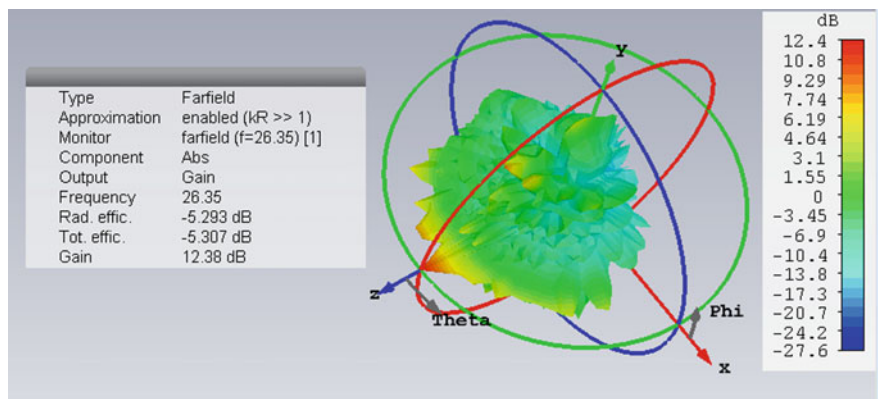


Fig. 7 3D view of gain of the designed antenna for frequency 26.35 GHz (in free space)

Table 2 Summary of simulation results

| Parameters                | Value  |         |                 |
|---------------------------|--------|---------|-----------------|
| Resonance frequency (GHz) | 22.99  | 24.24   | 26.35           |
| Return loss (S11) (dB)    | −22.65 | −23.543 | −24.58          |
| Bandwidth (GHz)           | 0.287  | 0.327   | 0.767           |
| VSWR                      | 1.159  | 1.142   | 1.12            |
| Gain (dB)                 | 16.81  | 18.41   | 12.38           |
| Directivity (dBi)         | 19.95  | 22.04   | 17.67           |
| Surface current (A/m)     | 58.7   | 67.07   | 47.65           |
| Antenna efficiency (%)    | 84.26  | 83.53   | 70.06           |
| Radiation efficiency (dB) | −3.145 | −3.625  | −5.29           |
| Total efficiency (dB)     | −3.169 | −3.644  | −5.307          |
| Band                      | K-band | ISM     | K-band (26 GHz) |

4 Comparative Study

All antenna parameter findings are presented in Table 3 and [10–12], as well as comparisons. This study has a higher gain of 18.41 dB, better bandwidth of 0.327 GHz, and S11 −23.5 dB than other research articles in Refs. [10–12], as seen in Table 3.

This paper, on the other hand, has a greater efficiency of 83.53% for this specific ISM band, whereas other research papers have a gain of 42 and 81.53% which are less than this paper. In Ref. [13], the dense dielectric patch array antenna’s result has been compared with this proposed antenna, which operates at 24.2444 GHz. [13] This paper has an efficiency of 71.8%, which is 11.73% less than the proposed work [14]. This paper presents a circularly polarized array that has fewer S11 parameter

**Table 3** Comparative analysis of antenna parameter of the existing and proposed antenna design at 24.24 GHz

| References | S1,1 (dB) | Resonant frequency (GHz) | Bandwidth | Band    | Gain     | Efficiency (%) |
|------------|-----------|--------------------------|-----------|---------|----------|----------------|
| [10]       | −19       | 24.5                     | 2 GHz     | –       | 20 dBi   | 42             |
| [11]       | −22.1     | 24.2                     | 1.2 GHz   | ISM     | 5.95 dBi | –              |
| [12]       | −14.7     | 24.4                     | 530 MHz   | ISM     | 6.65 dBi | 81.82          |
| [13]       | –         | 27.5                     | 5 GHz     | Ka-band | 16.3 dBi | 71.8           |
| [14]       | −23 dB    | 24.15 GHz                | 0.2 GHz   | ISM     | 18 dBi   | –              |
| This work  | −23.5     | 24.244                   | 0.327 GHz | ISM     | 18.41 dB | 83.53          |

and less gain compared to this work. The greater value of S11 will assist the designed model to absorb additional power into the antenna due to high return loss.

The comparison of different antenna parameters is also included in this Table 4. Antenna findings for 22.99 and 26.35 GHz are compared to 28 and 28.73 GHz. In Refs. [12, 15], both antennas are active in Ka-band, whereas this designed model will work in K-band. In comparison with Refs. [12, 15], this work has better efficiency which is 84.26% than the other research work. In Ref. [16], this paper illustrates the performance of different planner arrays; for example,  $1 \times 1$ ,  $2 \times 2$ ,  $4 \times 4$ , and  $8 \times 8$ . In Table 4, this proposed work has been compared with an  $8 \times 8$  phased array antenna. In Ref. [16], this paper has less S11 which is −17 dB, whereas this work has better S11 values such as −24.58 dB and −22.65 dB for operating frequencies 26.35 GHz and 22.99 GHz, respectively. References [13, 17] have low efficiency and less gain compared to this work. To sum up, this work has a better return loss, better gain, good efficiency, and bandwidth, according to the comparison.

**Table 4** Comparative analysis of antenna parameter of the existing and proposed antenna design at 22.996 and 26.35 GHz

| References | S1,1 (dB) | Resonant frequency (GHz) | Bandwidth | Band    | Gain     | Efficiency (%) |
|------------|-----------|--------------------------|-----------|---------|----------|----------------|
| [12]       | −19.3     | 28                       | 900 MHz   | Ka-band | 7.02 dBi | 85.5           |
| [15]       | −24.8     | 28.73                    | 7.48 GHz  | Ka-band | 13.2 dBi | 63.42          |
| [16]       | −17       | 22.25                    | 2.4 GHz   | K-band  | 19 dB    | –              |
| [17]       | −13       | 27                       | 6 GHz     | Ka-band | 13 dBi   | 75             |
| [13]       | –         | 27.5                     | 5 GHz     | Ka-band | 16.3 dBi | 71.8           |
| This work  | −24.58    | 26.35                    | 0.767 GHz | K-band  | 12.38 dB | 70.06          |
|            | −22.65    | 22.99                    | 0.287 GHz | K-band  | 16.81 dB | 84.26          |

## 5 Conclusion

The proposed array antenna is considered compatible to work with all three frequencies 22.99, 24.24, and 26.35 GHz range at high efficiency for forthcoming 5G communication. At 22.99, 24.24, and 26.35 GHz gains are 16.81, 18.41, and 12.38 dB which shows it can communicate in a particular direction very efficiently. Various types of antenna fabrication can be problematic, costly, and time-consuming. On other hand, this work can be proven cost-effective and has high efficiencies. As it has three resonant frequencies and two bands ISM and K-band, it can be used for different purposes. The power divider can distribute the power to array elements with high accuracy. A phase shifter can be added to this design by which this proposed antenna will be able to beam streaming for scan and point-to-point communication. After analysis, it can conclude the proposed work can be outsized in 5G communication which includes directive beam forming as well as will cover almost specific points due to point-to-point communication.

## References

1. Islam M, Islam M, Faruque M (2013) Dual-band operation of a microstrip patch antenna on a Duroid 5870 substrate for Ku- and K-bands. *Sci World J* 1–11:378420. <https://doi.org/10.1155/2013/378420>
2. Haupt R, Rahmat-Samii Y (2015) Antenna array developments: a perspective on the past, present and future. *IEEE Antennas Propag Mag* 57(1):86–96. Available: <https://doi.org/10.1109/map.2015.2397154>
3. Huang X, Guo Y, Bunton J (2010) A hybrid adaptive antenna array. *IEEE Trans Wirel Commun* 9(5):1770–1779. Available: <https://doi.org/10.1109/twc.2010.05.091020>
4. Karimkashi S, Kishk A (2009) Focused microstrip array antenna using a Dolph-Chebyshev near-field design. *IEEE Trans Antennas Propag* 57(12):3813–3820. Available: <https://doi.org/10.1109/tap.2009.2033435>
5. Dubey SK, Pathak SK, Modh KK (2011) High gain multiple resonance Ku-band microstrip patch antenna. In: 2011 IEEE applied electromagnetics conference (AEMC), pp 1–3. <https://doi.org/10.1109/AEMC.2011.6256845>
6. Midasala V, Siddaiah P (2016) Microstrip patch antenna array design to improve better gains. *Proc Comput Sci* 85:1–9, 401–409. <https://doi.org/10.1016/j.procs.2016.05.181>
7. Design and analysis of microstrip antenna in spiral structure for medical application. *Int Res J Eng Technol (IRJET)* 06(052019):1663–1666. Available: <https://www.irjet.net/archives/V6/i5/IRJET-V6I5332.pdf>. Accessed 14 July 2021
8. Motin MA, Hasan MI, Islam MS (2012) Design and simulation of a low cost three band microstrip patch antenna for the X-band, Ku-band and K-band applications. In: 2012 7th international conference on electrical and computer engineering, pp 397–400. <https://doi.org/10.1109/ICECE.2012.6471571>
9. Balanis A (1992) Antenna theory: a review. *Proc IEEE* 80(1):7–23
10. Ladan S, Guntupalli AB, Wu K (2014) A high-efficiency 24 GHz rectenna development towards millimeter-wave energy harvesting and wireless power transmission. *IEEE Trans Circuits Syst I Reg Papers* 61(12):3358–3366. <https://doi.org/10.1109/TCSI.2014.2338616>
11. Christina G, Rajeswari A, Mathivanan S (2017) Real time analysis of a 24 GHz planar microstrip antenna for vehicular communications. *Wireless Pers Commun* 97(1):1129–1139

12. Kamal MS, Islam MJ, Uddin MJ, Imran AZM (2018) Design of a tri-band microstrip patch antenna for 5G application. In: 2018 international conference on computer, communication, chemical, material and electronic engineering (IC4ME2), pp 1–3. <https://doi.org/10.1109/IC4ME2.2018.8465627>
13. Haraz OM, Elboushi A, Alshebeili SA, Sebak A (2014) Dense dielectric patch array antenna with improved radiation characteristics using EBG ground structure and dielectric superstrate for future 5G cellular networks. *IEEE Access* 2:909–913. <https://doi.org/10.1109/ACCESS.2014.2352679>
14. Radnovic I, Jokanovic B, Boryssenko A (2018) Circularly polarized patch antenna array at 24 GHz for radar applications. In: 2018 26th telecommunications forum (TELFOR), pp 1–4. <https://doi.org/10.1109/TELFOR.2018.861182>
15. Jose M, Sankararajan R, Sreeja BS, Kumar P (2019) Laser micro-machined 28 GHz broadband single-feed microstrip antenna for 5G mm-wave applications. *Circuit World* 46:6–12. <https://doi.org/10.1108/CW-06-2019-0053>
16. Ojaroudiparchin N, Shen M, Pedersen GF (2016) 8×8 planar phased array antenna with high efficiency and insensitivity properties for 5G mobile base stations. In: 2016 10th European conference on antennas and propagation (EuCAP), pp 1–5. <https://doi.org/10.1109/EuCAP.2016.7481323>
17. Stanley M, Huang Y, Loh T, Xu Q, Wang H, Zhou H (2017) A high gain steerable millimeter-wave antenna array for 5G smartphone applications. In: 2017 11th European conference on antennas and propagation (EUCAP), pp 1311–1314. <https://doi.org/10.23919/EuCAP.2017.7928542>

# Chapter 13

## Spatial-Spectral Kernel Convolutional Neural Network-Based Subspace Detection for the Task of Hyperspectral Image Classification



Md. Hafiz Ahamed and Md. Ali Hossain

**Abstract** Subspace uncovering is one of the most critical tasks for efficient hyperspectral image (HSI) classification. Recently, convolutional neural networks (CNNs) outperformed in HSI classification problems. However, the inherent complexity of complex HSI data cubes still affects the classification outcomes of many CNN-based models. Additionally, CNN sometimes degrades its performance for complex non-linear data because it provides only pointwise nonlinearity. To address the above-mentioned issues, the spatial-spectral kernel convolutional neural network (SSKCNN) has been proposed to classify the HSI data effectively. This model jointly extracts spatial and spectral features via patch-wise convolutional kernel functions. The experimental analysis of the University of Pavia dataset indicates that the proposed model provides a noteworthy improvement in terms of various classification metrics.

**Keywords** Subspace detection · Hyperspectral image classification · Kernel learning · CNN

## 1 Introduction

With the remarkable progress in remote sensing devices and their underlying technology, hyperspectral imaging have gained its position in earth surface monitoring and subspace detection. Hyperspectral image (HSI) contains numbers of the rich spectral band. In contrast, each spectral band having high spatial resolution provides detailed information about the subspaces, such as precise agricultural area monitoring, environment observation, and geological exploring [1]. So, the subspace exploration and the classification of HSI data is one of the fundamental prospects in HSI processing. A large number of spectral bands with few labels sample make it

---

Md. Hafiz Ahamed (✉) · Md. Ali Hossain  
Rajshahi University of Engineering and Technology, Rajshahi 6204, Bangladesh  
e-mail: [hafiz.mte13.ruet@gmail.com](mailto:hafiz.mte13.ruet@gmail.com)



difficult to classify the HSI data; therefore, an efficient learning algorithm is essential to classify the HSI in the most efficient way.

To address the HSI classification challenges, researchers have introduced several learning techniques. Initially, various handcrafted feature-based learning algorithms like simple KNN [2], extreme learning machine (ELM) [3], random forest [4], etc. These algorithms mainly focus on the original spectral signature of the rich spectral band and transfer it into the learned feature space. Additionally, SVM [5] with kernel trick plays a significant role in classifying the hyperspectral image. These pixels wise classifiers often provide satisfactory results having a significant amount of high-quality data. However, when the number of the training sample is limited, these methods do not perform well due to the curse of dimensionality. To incorporate this issue, various deep learning-based approaches have been proposed incorporating both spatial and spectral features. With the presence of an end-to-end structure, deep learning-based classifier boosts up the classifier's performance at the remarkable stage. Additionally, various deep learning-based pixel-wise classifier like stacked autoencoder (SAE) [6], deep belief network (DBN) [7], 1-D CNN [8], RNN [9] has gained attention in the field of HSI. To incorporate the spatial-spectral feature 2-D CNN [10], 3-D CNN [11] is also designed to improve the classifier accuracy. These methods process the raw feature data and directly transfer the raw feature space into the final classification feature map. While directly processing this feature map may hamper the classifier performance for some time while the data is more complex. Skip connection-based residual network (Resnet) [12, 13] can eliminate this type of problem for the HSI classification task. Additionally, various attention mechanisms [14, 15] based learning algorithm has also been introduced for HSI to capture the spatial-spectral features to accelerate the classification accuracy effectively.

Although these numerous deep learning-based algorithms most effectively classify the HSI data and perform better than other statistical techniques, it still suffers from nonlinearity errors. When the data are not linearly separable, the kernel function can handle this problem by mapping the original feature space into a high dimensional feature space to make the data linearly separable. Kernel-based classifiers can effectively handle the curse of dimensionality, noise exploration, limited labeled samples, etc. In this article, to address the deep learning-based data-driven methods of patch-wise nonlinearity in subspace identification for effectual HSI data classification, a spectral-spatial kernel convolution neural network is proposed for HSI classification. Here, the base convolutional operation is transferred into kernel convolution operation, and a composite spectral-spatial structure is proposed to classify the HSI data effectively.

The rest of the paper is organized as follows: Sect. 2 describes the detailed methodology and the structural architecture, the experimental evaluation is described in Sect. 3, and Sect. 4 concludes and summarizes our work.

## 2 Methodology

This section will describe the detailed working principle of spatial-spectral kernel CNN (SSKCNN) architecture. Figure 1 represents the structural block diagram of the proposed SSKCNN model, which consists of a spatial kernel convolution block and spectral kernel convolution block.

Let  $\mathbf{X} \in \mathcal{R}^{h \times w \times b}$  represents the HSI data cube where  $h, w$  represent the height and width of the data cube, respectively, and  $b$  represents the number of spectral bands. The CNN-based method explores spatial and spectral features from the adjacent HSIs pixels. To apply the CNN to the HSI data, let us consider the raw image data cube as a square box where the data square can be represented as  $\mathbf{X}_s = \{X_1, X_2, X_3, \dots, X_n\}$ , where,  $\mathbf{X}_i \in \mathcal{R}^{n \times n \times b}$ , and  $n \times n$  is the spatial dimension of the HSI cube. To apply the convolution operation to the square data cube, the dot product is performed with respect to  $w$  kernel filter.

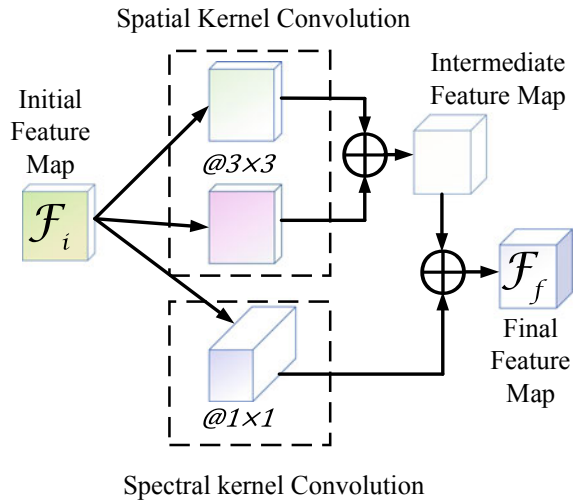
$$\mathbf{X}^C = \mathbf{W}^c * \mathbf{X}^{c_0} + b \quad (1)$$

In general, Eq. (1) can be written as  $\mathbf{X}^C = \mathbf{X} \oplus \mathbf{W}$ . Where  $\oplus$  represents the generalized convolution operator. To extract the complex non-linear features, each element of the inner product can be replaced by a kernel function [16].

After transforming the inner product from linear space to higher-dimensional space, the final convolution operation be like  $\mathbf{X}^{ck} = \mathbf{X} \otimes \mathbf{W}$ , where the  $\mathbf{X}^{ck}$  can be defined as:

$$\mathbf{X}^{ck} = \langle \phi(\mathbf{W}^c) * \phi(\mathbf{X}^{c_0}) \rangle \quad (2)$$

**Fig. 1** Structural block diagram of our proposed model



where,  $\phi(\circ): \mathcal{R}^n \rightarrow \mathcal{R}^d$  is a nonlinear mapping or kernel function. To map each of the inner components to the high-dimensional feature space will increase the computational complexity. Using the benefit of the kernel trick reduces this complexity; it maps the entire feature vector to the higher-dimensional feature space without explicitly mapping each inner product. In this work, to map the feature, we use the most commonly used two kernel functions, Gaussian, and sigmoid kernel functions.

Gaussian Kernel:

$$\phi^G(X, W) = \exp\left(-\frac{\|X^{c0} - W^c\|^2}{2\sigma^2}\right) \quad (3)$$

where  $\sigma$  represents the spectral width of the Gaussian kernel.

Sigmoid Kernel:

$$\phi^S(X, W) = \tanh\left((X^{c0})^T \cdot W^c\right) \quad (4)$$

The overall architectural block comprises spatial convolutional kernel operation (kernel size  $3 \times 3$ ), spectral convolutional operation (kernel size  $1 \times 1$ ), batch normalization for normalizing the feature map, and ReLu was utilized to add nonlinearity to the intermediate layer. Max pooling, global average pooling, and dropout were also utilized to build the end-to-end architecture, and finally, SoftMax provided the classification map of the architecture.

### 3 Experimental Evaluation

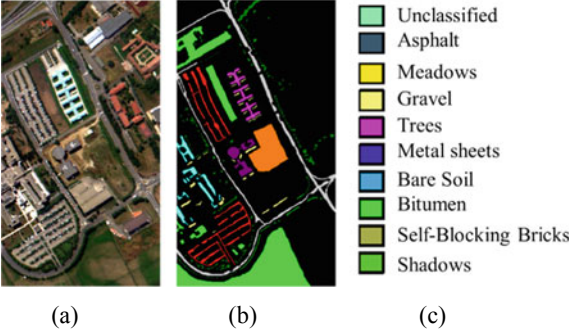
#### 3.1 Dataset Description

To evaluate our overall experiment, we chose the University of Pavia (UP) dataset, which was captured by the ROSIS sensor. The entire Pavia Centre's size is  $1096 * 1096$  and has 102 spectral bands, and the Pavia University holds  $610 * 610$  pixels and 103 spectral bands. Each image contains some samples containing no information that needed to be eliminated before the exploration. This data holds a total of 42,776 labeled pixels having 1.3 spatial resolution. The ground truth image has a total of 9 different classes. The RGB color space, ground truth, and the class label are shown in Fig. 2.

#### 3.2 Performance Evaluation

To justify our proposed method, some qualitative and quantitative analyzes with the method have been performed. To do this, the attained result is compared with

**Fig. 2** University of Pavia dataset, **a** false color map, **b** ground truth, **c** class labels



both handcrafted model-based and deep learning-based classifiers. Table 1 represents the comparative outcome of the UP-dataset classification report. The 1D and 2D CNN perform better than the traditional handcrafted methods; further, 3D CNN and ResNet boost the classification accuracy with the spectral and spatial feature handling capability. However, our method outperforms other methods with non-linear spectral and spatial feature handling capability.

Figure 3 visualize the classification map of various classification models. From the false-color map, it is clear that SVM, 1D, and 2D CNN experienced serious misclassification problems, whereas 3D CNN provides better output in terms of pixel quality but fails to capture all class labels correctly. The outcome from the proposed model eliminates these issues and provides superior results for both qualitative and quantitative analysis.

**Table 1** Comparative performance analysis on 10% training samples on up dataset having  $11 \times 11$  input size

| Class        | LR    | SVM        | MLP   | 1-D CNN | 2-D CNN    | 3-D CNN    | ResNet       | SSKCNN        |
|--------------|-------|------------|-------|---------|------------|------------|--------------|---------------|
| Asphalt      | 92.78 | 93.20      | 94.07 | 95.30   | 97.99      | 99.61      | 98.12        | <b>99.68</b>  |
| Meadows      | 96.28 | 97.44      | 97.89 | 98.10   | 98.78      | 99.99      | 98.81        | <b>100</b>    |
| Gravel       | 73.86 | 78.21      | 74.30 | 75.08   | 93.08      | 91.61      | 93.51        | <b>99.52</b>  |
| Trees        | 89.00 | 95.44      | 93.54 | 96.49   | 98.86      | 98.56      | <b>99.20</b> | 98.48         |
| Metal sheets | 99.21 | 99.41      | 99.85 | 99.86   | <b>100</b> | 99.87      | <b>100</b>   | 99.84         |
| Bare soil    | 77.34 | 91.00      | 93.31 | 91.30   | 94.49      | <b>100</b> | 95.36        | <b>100</b>    |
| Bitumen      | 57.17 | 85.05      | 81.10 | 91.25   | 93.11      | 97.11      | 90.09        | <b>100</b>    |
| Bricks       | 87.96 | 89.31      | 92.35 | 91.56   | 98.15      | 96.55      | 97.55        | <b>99.22</b>  |
| Shadows      | 99.76 | <b>100</b> | 99.61 | 99.91   | 99.89      | 98.48      | 99.30        | 97.07         |
| OA %         | 90.21 | 94.10      | 94.40 | 94.90   | 97.71      | 99.00      | 97.73        | <b>99.686</b> |
| AA %         | 85.95 | 92.15      | 91.80 | 93.16   | 97.15      | 97.97      | 96.88        | <b>99.271</b> |
| Kappa        | —     | 92.05      | 92.56 | 93.22   | 96.92      | 98.67      | 96.98        | <b>99.584</b> |

The best output of each row is bold

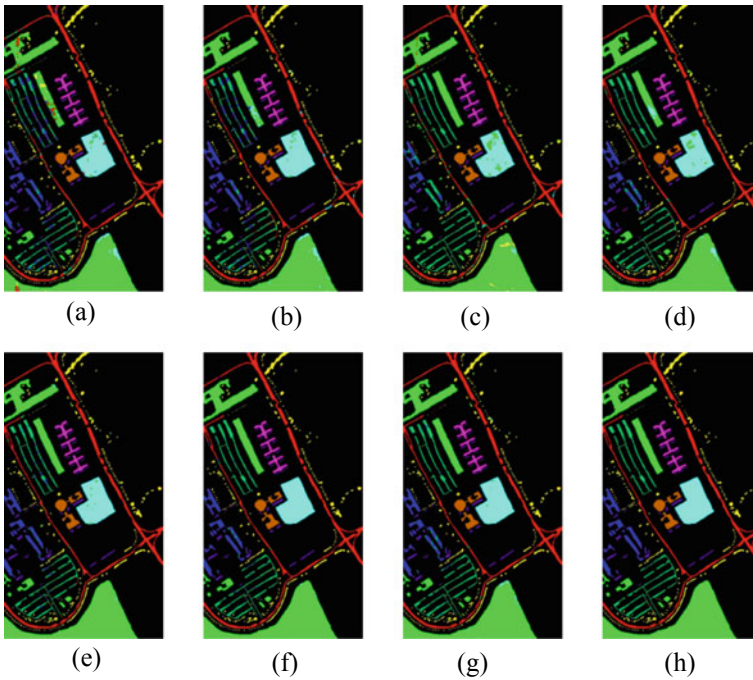


**Fig. 3** Classification map on UP dataset. **a** LR, **b** SVM, **c** MLP, **d** 1-D CNN, **e** 2-D CNN, **f** 3-D CNN, **g** ResNet, **h** SSKCNN

To validate the robustness of the proposed methods, some ablation studies on the model have been performed. Figure 4 indicates that by increasing the spatial dimension of the input space, the classification accuracy gradually increases, but the increasing spatial window will also lead to feature overlapping. We chose  $11 \times 11$  as our default input window size to get the optimal output without overlapping.

Figure 5 represents the variation in training sample size and the corresponding performance metrics. It indicates that the minimum number of training samples, i.e., 5% training samples, provides more than 98% accuracy in all three metrics, which indicates that the proposed model is robust against various training samples and can handle the training data incompetence problems.

## 4 Conclusion

This paper presents the proposed spatial-spectral kernel convolutional neural network (SSKCNN) for hyperspectral image classification. This proposed method extracts jointly spatial and spectral features in nonlinear space via kernel convolution. Here, Gaussian and sigmoid kernel functions provide patch-wise nonlinearity, and the

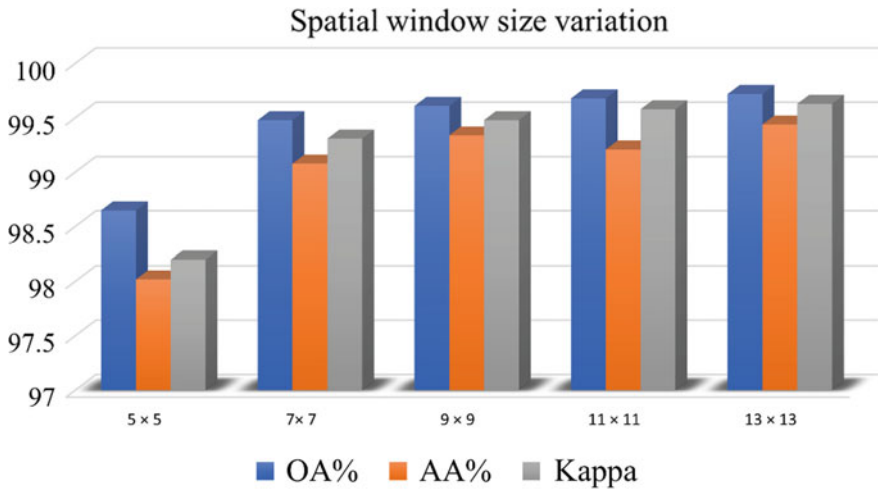


**Fig. 4** Variation of the spatial window on University of Pavia dataset having 10% training sample

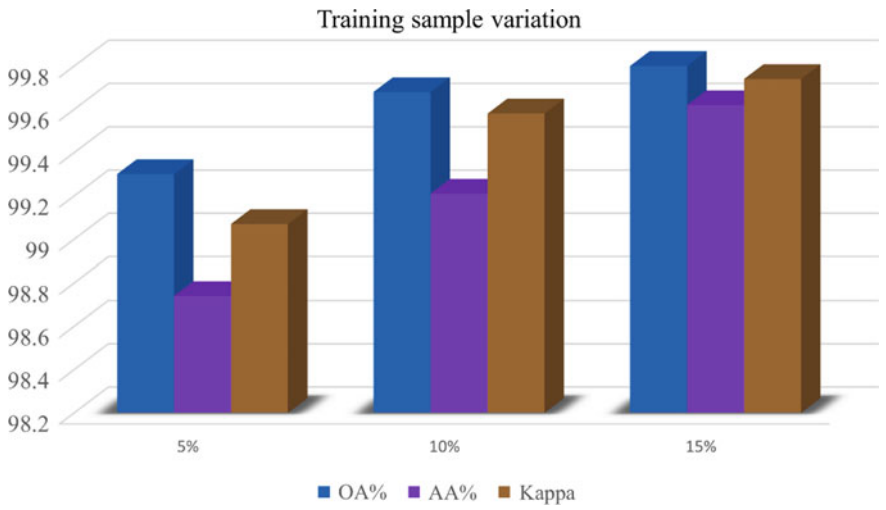


**Fig. 5** Variation of training sample on University of Pavia dataset and the corresponding performance metrics

kernel trick benefits from increasing additional computational complexity. Overall, the proposed method provides superior performance to the state-of-the-art methods, and it converges faster than base CNN architecture. We plan to incorporate composite and selective kernels for better generalization in the future.

## References

1. Yu S, Jia S, Xu C (2017) Convolutional neural networks for hyperspectral image classification. *Neurocomputing* 219:88–98. <https://doi.org/10.1016/J.NEUCOM.2016.09.010>
2. Blanzieri E, Melgani F (2008) Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Trans Geosci Remote Sens* 46(6):1804–1811. <https://doi.org/10.1109/TGRS.2008.916090>
3. Su H, Cai Y, Du Q (2017) Firefly-algorithm-inspired framework with band selection and extreme learning machine for hyperspectral image classification. *IEEE J Sel Top Appl Earth Observations Remote Sens* 10(1):309–320. <https://doi.org/10.1109/JSTARS.2016.2591004>
4. Ham JS, Chen Y, Crawford MM, Ghosh J (2005) Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans Geosci Remote Sens* 43(3):492–501. <https://doi.org/10.1109/TGRS.2004.842481>
5. Wang Y, Yu W, Fang Z (2020) Multiple kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information. *Remote Sens* 12:120. <http://doi.org/10.3390/RS12010120>
6. Xing C, Ma L, Yang X (2016) Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. *J Sens* 2016. <http://doi.org/10.1155/2016/3632943>
7. Chen Y, Zhao X, Jia X (2015) Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE J Sel Top Appl Earth Observations Remote Sens* 8(6):2381–2392. <https://doi.org/10.1109/JSTARS.2015.2388577>
8. Hu W, Huang Y, Wei L, Zhang F, Li H (2015) Deep convolutional neural networks for hyperspectral image classification. *J Sens* 2015. <http://doi.org/10.1155/2015/258619>
9. Mou L, Ghamisi P, Zhu XX (2017) Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 55(7):3639–3655. <https://doi.org/10.1109/TGRS.2016.2636241>
10. Paoletti ME, Haut JM, Fernandez-Beltran R, Plaza J, Plaza AJ, Pla F (2019) Deep pyramidal residual networks for spectral-spatial hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 57(2):740–754. <https://doi.org/10.1109/TGRS.2018.2860125>
11. Lin J, Mou L, Zhu XX, Ji X, Wang ZJ (2021) Attention-aware pseudo-3-D convolutional neural network for hyperspectral image classification. *IEEE Trans Geosci Remote Sens*. <https://doi.org/10.1109/TGRS.2020.3038212>
12. Zhong Z, Li J, Luo Z, Chapman M (2018) Spectral-spatial residual network for hyperspectral image classification: a 3-D deep learning framework. *IEEE Trans Geosci Remote Sens* 56(2):847–858. <https://doi.org/10.1109/TGRS.2017.2755542>
13. Sarker Y, Hafiz Ahamed M, Asif NA, Rahman Fahim S, Das SK (2021) Classification of hyperspectral imagery using spectral-spatial residual attention network. In: 2021 international conference on automation, control and mechatronics for industry 4.0, ACMI 2021. <http://doi.org/10.1109/ACMI53878.2021.9528299>
14. Haut JM, Paoletti ME, Plaza J, Plaza A, Li J (2019) Visual attention-driven hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 57(10):8065–8080. <https://doi.org/10.1109/TGRS.2019.2918080>
15. Hang R, Li Z, Liu Q, Ghamisi P, Bhattacharyya SS (2021) Hyperspectral image classification with attention-aided CNNs. *IEEE Trans Geosci Remote Sens* 59(3):2281–2293. <https://doi.org/10.1109/TGRS.2020.3007921>
16. Wang C, Yang J, Xie L, Yuan J (2019) Kervolutional neural networks. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, vol 2019, pp 31–40. <http://doi.org/10.1109/CVPR.2019.00012>

# Chapter 14

## Sentiment Analysis and Emotion Detection from Khasi Text Data—A Survey



Banteilang Mukhim and Sufal Das

**Abstract** The state of Meghalaya, which is one of India's seven north-eastern states and shares an international border with Bangladesh, boasts 1.6 million native speakers of the Khasi language (according to the 2001 census). The main objective of this paper is to provide an overview of the most recent techniques and models for sentiment analysis, with special attention to low-resource languages like the Khasi language. The survey summarises the popular data sets and models applied to them, the accuracy obtained from them, and also a comparison of these models. This paper also highlighted the scope and challenges of sentiment analysis and emotion detection on Khasi text data. Finally, a future directions are provided for research work with sentiment analysis with Khasi text data.

**Keywords** Sentiment analysis · Emotion detection · Khasi text analysis · Classification · Opinion mining

## 1 Introduction

A powerful sensation resulting from one's situation or attitude, or interactions with others is what the Oxford Dictionary defines as “emotion”. The American Psychological Association defines “emotion” as a multifaceted series of changes brought on by a perceived personally significant event, involving physiological responses, senses, cognitive functions, and behavioural responses. Generally speaking, “emotion” refers to a person's feeling or response to a particular situation. There are many different emotions one might experience, including joy, sorrow, anger, and fear.

Finding or detecting people's emotions [1], likes, dislikes, etc. from the content that they post on social media platforms is becoming an area of great interest in today's world. Research in this field is known as “emotion analysis” or “sentiment analysis”. The polarity of a text may be ascertained using the natural language processing (NLP)

---

B. Mukhim · S. Das (✉)

Department of Information Technology, North Eastern Hill University, Shillong, Meghalaya, India  
e-mail: [sufal.das@gmail.com](mailto:sufal.das@gmail.com)



approach of sentiment analysis, such as whether it is positive, negative, neutral, or ambiguous. In addition to polarity, sentiment analysis may also pinpoint certain feelings and moods like anger, happiness, sorrow, and so on.

Sentiment analysis [1] has many areas of application, namely, e-commerce (for finding likes, dislikes, consumer behaviour, etc. which is useful for predicting future trends), public opinion analysis (to obtain the state of public opinion on some sensitive issues; this will help the authorities to take timely action), big search (a search that understands user search intentions based on their geographic and temporal position, feeling, and previous preferences), information prediction (e.g. election and financial prognostication), personalised recommendation (channelising only useful information to the user based on the emotional information extracted from their comment), health care (e.g. evaluating depression), and online teaching, students may provide emotional data about their courses, which can be used by professors to identify issues with their curricula, knowledge systems, and instructional strategies [2].

Bengali was the first script used to write the Khasi language. But Thomas Jones, a Welsh-born Christian missionary, was the first to use the Roman script to establish the Khasi alphabet. His earliest works in the language are “Ka Kot Pule Nyngkong” (Khasi First Reader) and “Ka Kitab Nyngkong”. These two primers served as the earliest introductions to the 21 alphabets of the Roman script in the Khasi language. There were 16 consonant sounds (b, k, d, g, h, j, l, m, n, p, r, s, t, w, y) and five vowel sounds (a, e, i, o, u). In 1896, two more sounds-ĩ and ñ-were added, increasing the total number of alphabets to 23 [3].

The use of sentiment analysis or emotion analysis on the Khasi texts will be beneficial to all and have a good social impact. To my knowledge, there hasn’t been any related study in this field for the Khasi language as of yet. However, there are few works in other areas for Khasi language, namely, Low-Resource Neural Machine Translation From English to Khasi: A Transformer-Based Approach [4], Low-Resource and Domain-Specific English to Khasi SMT and NMT Systems [5], Khasi Shallow Parser [6], Bidirectional Gated Recurrent Unit for Shallow Parsing [7], Identification of POS Tag for Khasi Language based on Hidden Markov Model POS Tagger [8].

## 2 Related Works

Sentiment analysis was first applied on written paper documents in the 1950s, when it dates its beginnings. Nowadays, sentiment analysis is increasingly often used to extract subjective data from online content, such as social media postings, news stories, blogs, and comments. Some of the related works in this field are discussed below and the method and data set as well as the accuracy score of each method is highlighted in Table 2.

## ***2.1 Opinion Mining Based on Aspects***

In this research, the author uses text-only customer reviews with no labels for the purpose of aspect-based opinion polling. In an experiment on reviews of Chinese restaurants, accuracy in tasks using aspect-based opinion polling was 75.50% [9]. The author of this work proposes a multi-aspect segmentation (MAS) model that creates several single-aspect parts from a multi-aspect statement.

## ***2.2 Cross-Domain Sentiment Analysis***

It is a form of sentiment analysis where a sentiment classifier created for one specific domain is being applied to another domain. The author uses both labelled and unlabelled data on the source domain and unlabelled data on the target domain to develop a sentiment sensitive distributional thesaurus. The feature vectors are then constructed using the thesaurus [10].

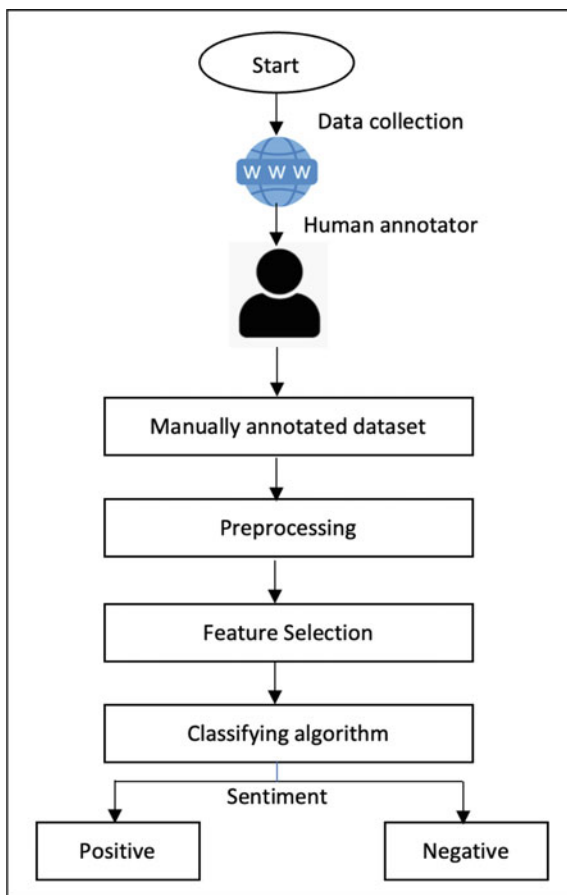
## ***2.3 Evaluating the Effectiveness of MOOCs on Students' Results***

In this research, the author examines student perceptions of the course content and instructor's clarity using natural language processing techniques and data mining algorithms. It also tries to determine if textual information communicated through MOOCs, such as quality vs number of student forum conversations, might act as a leading indication of student achievement in MOOCs [11].

## ***2.4 Sentiment Analysis on Limited-Resource Language***

In this work, the author seeks to present a benchmark investigation for sentiment analysis on Urdu text data [12]. This study compares the performance of deep learning classifiers like the one-dimensional convolutional neural network [13] and long short-term memory (LSTM) with machine learning classifiers like the random forest (RF), naive bayes (NB), support vector machine (SVM), AdaBoost, multilayer perceptron (MLP), and logistic regression (LR). Among these word, n-gram [14] features with LR have proven to be the most effective method, obtaining the highest  $F1$  score of 82.05%. Figure 1 [12] depicts the system's overall design. The lack of recognisable lexical resources is just one of several challenges that the author faces in this research [15, 16].

**Fig. 1** System design for sentiment analysis in Urdu [12]



The authors of this work compiled a standard Urdu language text corpus by gathering information about different genres from widely accessible online repositories. Data on politics, cinema, Urdu theatre, TV discussion shows, and sporting events were gathered.

The data set was then manually annotated by the appointed annotators. Figure 2 [12] provides samples of user reviews that fall into the negative and positive classes. The following are the annotations rules:

1. A statement is categorised as positive if it contains both favourable and unfavourable opinions, or if it also contains agreement and approbation [17].
2. Sentences including phrases like congrats and appreciation were also classified as positive [18].
3. If a statement primarily expresses unfavourable feelings or has more unfavourable words than favourable ones, it is said to be negative [19].
4. Any statement that demonstrates disagreement is categorised as negative [18].

| Negative Review                                                                                                | Positive Review                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>یہ سافٹ ویئر قابل عمل نہیں ہے<br/>(Ye Software Kabil e Amal nahi hai)<br/>This software is not workable</p> | <p>یہ خوش کن ہے<br/>(Ye Khush Koun Hai )<br/>This is Pleasing</p>                                                                                   |
| <p>مجھے ایسے لوگوں سے نفرت ہے<br/>(Mujhy Eisay Lougoo se nafrat hai)<br/>I hate such People</p>                | <p>میں نے کوشش کی ہے کہ یہ نسخہ بہت اچھا ہے<br/>(Main Ne koshish ki hai ye nuskha Bahot Acha hai)<br/>I have tried this recipe, It is very good</p> |

Fig. 2 Data set samples and their translations [12]

Table 1 Number of positive and negative reviews

| Review   | Number |
|----------|--------|
| Negative | 4758   |
| Positive | 4843   |
| Total    | 9601   |

- 5. A statement is classified as negative if it contains words like “ban” or “penalty” [18].
- 6. When a sentence has both a negative content and a positive adjective, the sentence is considered negative [20].

Table 1 [12] gives the number of positive and negative reviews in the data set.

Advantages

- 1. The data set used in this study is substantially larger when compared to the earlier research in this field [21].
- 2. This finding opens up a brand-new field for future scholars to investigate languages with little resources.

Disadvantages

- 1. There are only positive and negative classifications in this research.

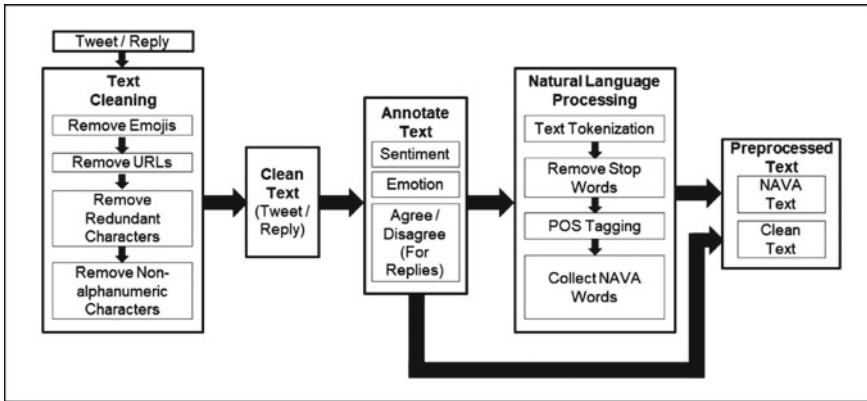


Fig. 3 Pre-processing workflow [22]

## 2.5 Emotion Detection From Textual Tweets

This research aims to identify and analyse the sentiment and feelings individuals convey via their text in their tweets in order to produce both general and specific user suggestions based on their tweet activity [22]. Figures 3 [22] and 4 [22] depict the pre-processing processes and the workflow for classifying the pre-processed text and calculating the influence score to produce recommendations, respectively. For classifications of mood and sentiment, the pre-processed text comprising tweets and replies was split into training sets and test sets. The “user influence score” was then calculated while taking emotion and sentiment into account. The recommender system used the score to produce a list of people who had similar feelings and sentiments about a certain subject. The pre-processed text (tweets and replies) is then subjected to the Naive Bayes algorithm.

Various tweets, comments, retweets, and user data were collected for this study Fig. 5 [22]. The few steps that were performed to prepare the data set are listed below:

1. Gathering random tweets on a subject.
2. Information gathering of users (user ID, location, gender, etc.)
3. Collecting feedback and responses for each tweet.
4. Gathering of commenters’ information.
5. Garnering retweets for the tweet.
6. Gathering each retweeter’s user information.

For each keyword, the aforementioned procedures were repeated. A total of 7246 tweets from 3607 users were collected.

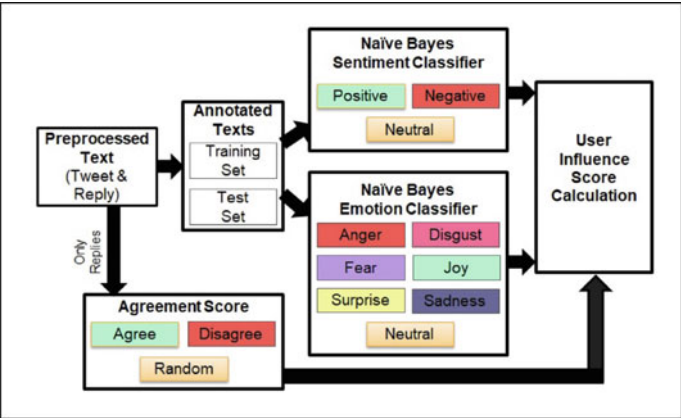


Fig. 4 Workflow for classification and calculating influence scores [22]

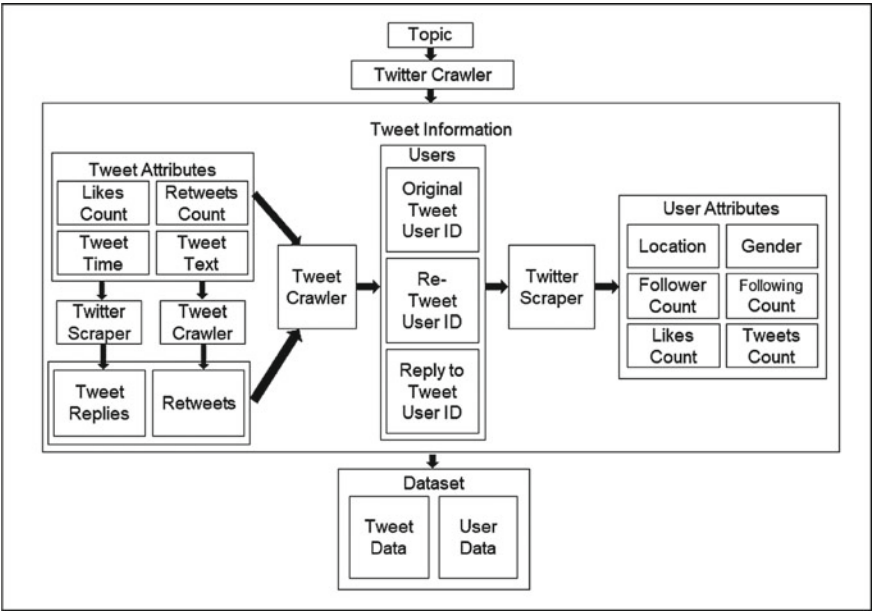


Fig. 5 Data set creation workflow [22]

2.6 Sentiment Analysis Based on the Attention Model

Multi-AFM [23] is an aspect-based sentiment analysis task that blends global and local attention together through gating unit control. In this study, the author creates a sentence representation based on the embedded aspects using a bidirectional LSTM [24] first, allowing the sentence to concentrate on a single distinct aspect at a time. The integration of global and local attention is then controlled using the gating approach

**Table 2** Comparison of existing sentiment analysis methods

| No | Name           | Method                                                     | Data set                                                                                                                                                                                                   | Accuracy                                                                                                                                                 |
|----|----------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Zhu [9]        | MAS [9]                                                    | Reviews of restaurant                                                                                                                                                                                      | Precision—0.6,<br>Recall—0.56, $F1$ —0.62                                                                                                                |
| 2  | Bollegala [10] |                                                            | Product reviews on Amazon online shopping                                                                                                                                                                  | Precision—79.95%,<br>Recall—84.26%,<br>$F1$ —82.05%                                                                                                      |
| 3  | Khan [12]      | SVM, Naïve Bayes, Random forest, MLP, LR, 1D-CNN, and LSTM | User reviews on political parties, gaming platforms, and various goods and services                                                                                                                        |                                                                                                                                                          |
| 4  | Sailunaz [22]  | Naïve Bayes                                                | Twitter                                                                                                                                                                                                    | 66.86                                                                                                                                                    |
| 5  | Zhai [23]      | Multi-AFM                                                  | Course data set ( <a href="https://www.kaggle.com/septa97/">https://www.kaggle.com/septa97/</a> ), Restaurant data set ( <a href="http://alt.qcri.org/semeval2014/">http://alt.qcri.org/semeval2014/</a> ) | Education:<br>Accuracy—94.6,<br>Precision—92.2,<br>Recall—93.6, $F1$ —90.7,<br>Restaurant:<br>Accuracy—79.6,<br>Precision—76.8,<br>Recall—69, $F1$ —70.3 |

to provide an appropriate attention distribution. After this, the global attention and local attention related to the embedded sentences are created. The Softmax classifier is then fed the outcome for final classification.

To gauge how well this model works, experiments were conducted not only on the education data sets but on other data sets as well. Three data sets, namely, the education data set, course data set, and restaurant data set, were used to evaluate the multi-AFM model [23].

1. The education data set includes information on more than 3000 college freshmen's course evaluations during the academic years of 2014–2017 across a variety of courses, grades, and instructors. The four areas that the data set focuses on are difficulty, substance, practicability, and instructor.
2. The education website Coursera provided the course data set.
3. The restaurant data set contains information about user reviews (Table 2).

### 3 Challenges of Sentiment Analysis

Emotion analysis faces a slew of unforeseen obstacles. Despite the fact that many academics have provided probable solutions for the majority of them, there are still a number of unsolved challenges that need to be investigated further. Among the difficulties of emotion analysis are the following:

1. The ability to extract emotional features from text may suffer from the lack of a common definition for emotion and a clear criteria for correctly and scientifically categorising emotions.

2. The majority of sentiment analysis models presuppose that a data set's positive and negative sample populations are balanced and evenly distributed. Positive and negative samples are, however, dispersed unevenly in practise. As a result, when balanced classification methods are used to handle unbalanced data, the expected effects of the analytical outputs typically do not materialise, negatively compromising the performance of emotion categorization.
3. The majority of the currently used emotion analysis techniques are designed to be used with English texts. Adapting emotion analysis techniques from English to other languages is challenging.
4. Words' domain relevance can occasionally influence emotion interpretation. Depending on the situation, adjectives and adverbs like "a long time" might imply a number of emotions. For example, long waiting times in food and beverage evaluations are frequently expressed as a negative feeling, whereas long battery standby hours in smartphone reviews are frequently expressed as a positive emotion [25].
5. Till date, there has been no sentiment analysis or emotion detection work carried out on Khasi text data. Therefore, constructing such a model will offer a lot of potential.

## 4 Challenges Associated with Sentiment Analysis on Khasi Text

Some of the challenges associated with sentiment analysis on Khasi texts may be elaborated as follows:

1. Lack of e-resources related with Khasi language data.
2. There is no complete Wordnet in Khasi language.
3. There is no standard recognized online POS in Khasi language.
4. No proper list of sentiment words in Khasi language is available.
5. Most of the comment text available on social media platforms is not purely Khasi language but a mixture of English and Khasi language.
6. Grammar errors are being common practice with Khasi text comments data.
7. In current days, several symbols, emoji, etc. are being used to provide reviews with Khasi text data.

## 5 Conclusion

In this paper, we have set out to evaluate the sentiment analysis research that has already been carried out. Sentiment analysis in other low-resource languages, such as the Urdu language, is highlighted since the Khasi language is one of them. This survey also highlighted some of the challenges associated with sentiment analysis on Khasi text data. We hope that this survey will provide data that may be used to help develop sentiment analysis and emotion identification algorithms.



## References

1. Liu B (2012) Sentiment analysis and opinion mining. In: *Synthesis lectures on human language technologies*, vol 5, no 1, pp 1–167
2. Peng S et al (2021) A survey on deep learning for textual emotion analysis in social networks. *Digital Commun Netw*
3. Hujon AV, Singh TD (2018) Existing English to Khasi translated documents for parallel corpora development: a survey. *Int J Nat Lang Comput (IJNLC)* 7(5):81–91
4. Thabab NDJ, Purkayastha BS (2021) Low resource neural machine translation from English to Khasi: a transformer based approach. In: *Proceedings of the international conference on computing and communication systems: I3CS 2020, NEHU, Shillong, India*, vol 170
5. Singh TD, Hujon AV (2020) Low resource and domain specific English to Khasi SMT and NMT systems. In: *2020 international conference on computational performance evaluation (ComPE)*. IEEE
6. Tham MJ (2018) Khasi shallow parser. In: *Proceedings of the 15th international conference on natural language processing (ICON 2018)*, Patiala, Punjab, India
7. Tham MJ (2020) Bidirectional gated recurrent unit for shallow parsing. *Indian J Comput Sci Eng (IJCSE)* 11(5):517–521
8. Warjri S et al (2019) Identification of POS tag for Khasi language based on hidden Markov model POS tagger. *Computación y Sistemas* 23(3):795–802
9. Zhu J et al (2011) Aspect-based opinion polling from customer reviews. *IEEE Trans Affect Comput* 2(1):37–49
10. Danushka B, David W, John C (2012) Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Trans Knowl Data Eng* 25(8):1719–1731
11. Tucker C, Pursel BK, Divinsky A (2014) Mining student-generated textual data in MOOCs and quantifying their effects on student performance and learning outcomes. In: *2014 ASEE annual conference and exposition*
12. Khan L et al (2021) Urdu sentiment analysis with deep learning methods. *IEEE Access* 9:97803–97812
13. Zhang Y et al (2019) Vibration-based structural state identification by a 1-dimensional convolutional neural network. *Comput Aided Civ Infrastruct Eng* 34(9):822–839
14. Yu HP, Xu HW (2012) Evaluation of machine translation with dependent Skip-Ngrams. In: *Proceedings of 2012 international conference on measurement, information and control*, vol 1. IEEE
15. Ijaz M, Hussain S (2007) Corpus based Urdu lexicon development. In: *The proceedings of conference on language technology (CLT07)*, University of Peshawar, Pakistan, vol 73
16. Daud A, Khan W, Che D (2017) Urdu language processing: a survey. *Artif Intell Rev* 47(3):279–311
17. Homburg T, Chiarcos C, Calzolari N (2016) *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*
18. Ganapathibhotla M, Liu B (2008) Mining opinions in comparative sentences. In: *Proceedings of the 22nd international conference on computational linguistics (Coling 2008)*
19. Rehman ZU, Bajwa IS (2016) Lexicon-based sentiment analysis for Urdu language. In: *2016 sixth international conference on innovative computing technology (INTECH)*. IEEE
20. Grave E et al (2018) Learning word vectors for 157 languages. *arXiv preprint [arXiv:1802.06893](https://arxiv.org/abs/1802.06893)*
21. Mukhtar N, Khan MA (2018) Urdu sentiment analysis using supervised machine learning approach. *Int J Pattern Recognit Artif Intell* 32(02):1851001
22. Kashfia S, Reda A (2019) Emotion and sentiment analysis from Twitter text. *J Comput Sci* 36:101003
23. Zhai G et al (2020) Multi-attention fusion modeling for sentiment analysis of educational big data. *Big Data Min Analytics* 3(4):311–319
24. Sepp H, Jürgen S (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
25. Cao Z et al (2021) Deep transfer learning mechanism for fine-grained cross-domain sentiment classification. *Connection Sci* 33(4):911–928

# Chapter 15

## Classification of Brain Hemorrhage Using Deep Learning from CT Scan Images



Nipa Anjum, Abu Noman Md. Sakib, and Sk. Md. Masudul Ahsan

**Abstract** Brain hemorrhage is a life-threatening problem that happens by bleeding inside human head. In this study, computed tomography (CT) scan images have been used to classify whether the case is hemorrhage or non-hemorrhage. Different convolutional neural network (CNN) models have been observed along with some pre-trained deep learning models such as VGG16, VGG19, ResNet150, ResNet152 and InceptionV3. Pre-trained models have performed well on the dataset but all of them are heavyweight architectures in terms of number of total parameters. But the proposed model is a lightweight architecture as well as a well performing one. After evaluating the model performance, it has been observed that the proposed model gave 96.67% accuracy, 97.08% sensitivity and 96.25% specificity which is the best among other custom CNN models.

**Keywords** Brain hemorrhage classification · Deep learning · Convolutional neural network · InceptionV3 · VGG16 · VGG19 · ResNet50 · ResNet152

## 1 Introduction

Brain hemorrhage is a potentially fatal condition that can be caused by physical trauma or a variety of medical problems such as high blood pressure [1]. When the artery in human brain bursts and starts bleeding inside the brain tissue, it causes the damage in human brain which leads to hemorrhage. Due to high blood pressure, aneurysm, bleeding disorder, brain tumor or any kind of trauma, brain hemorrhage can occur. If any person suddenly feels severe headache, weakness in arm or leg that makes him/her numb, starts vomiting and losing consciousness, these can be

---

N. Anjum (✉) · A. N. Md. Sakib · Sk. Md. Masudul Ahsan  
Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [nipaanjum1998@gmail.com](mailto:nipaanjum1998@gmail.com)

Sk. Md. Masudul Ahsan  
e-mail: [smahsan@cse.kuet.ac.bd](mailto:smahsan@cse.kuet.ac.bd)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_15](https://doi.org/10.1007/978-981-19-7528-8_15)

181

an augury of brain hemorrhage [2]. For diagnosing a person who is surmised of brain hemorrhage, physical symptoms are measured then brain images are observed. These images are obtained by computed tomography (CT) scan, magnetic resonance imaging (MRI) and magnetic resonance angiogram (MRA) techniques.

Among the world population, the incidence rate of brain hemorrhage in black, white, Asian, Hispanic people was 22.9, 24.2, 51.8 and 19.6, respectively. This survey was per 100,000 per-years. They also showed that the rate of hemorrhage increases in people who are more than 45 years old [3]. People who have had hemorrhage, has a chance to survive if proper treatment is provided instantly. For this, faster classification of brain hemorrhage is important. After surviving intracerebral hemorrhage, among 72,432 people, almost 18% died due to infection, cardiac disease, respiratory failure, ischemic stroke, recurrent intracranial hemorrhage within 4 years [4]. The percentage of fatality for a person diagnosed with brain hemorrhage might be as high as 57% depending on the type of hemorrhage [5]. Automated brain hemorrhage classification systems help to diagnose hemorrhage in human brain that helps to increase survival rate in human life. Brain hemorrhage can occur in older and younger people. Sometimes due to birth injury or force, infants can also have brain hemorrhage.

People are doing research works for finding better solutions to classify brain hemorrhage for doctors to give better and reliable treatments to the patients. Researchers [6–10] used classical image processing, machine learning algorithms or combines both algorithms, while some of them worked with deep learning approaches. Working with medical dataset is very challenging task as the resource is very scarce. Image processing techniques require huge time to classify images, and if the size of the dataset increases, it becomes very tedious. In medical image classification, identification of the problem requires high domain knowledge but the performance of such methods may not be appeasable. Machine learning techniques at first extract the features with the help of experts in specific domain. In this feature extraction procedure, different image processing algorithms can help to better understand the data by making patterns more evident and reduce the complexity of data. On the contrary, deep learning works well when the dataset size increases and deep learning algorithms extract the high-level features by learning the dataset on its own. This eliminates the knowledge of a specific domain which helps to solve any problem with limited domain knowledge. This has motivated to implement a deep learning-based approach to build a convolutional neural network that can classify brain hemorrhage. The focus in this research is to work on a small dataset and build a solution that can diagnose brain hemorrhage with an efficient result which can run on minimal system requirements. Brain hemorrhage can be of different types, the proposed method classifies whether there is a hemorrhage or not, overlooking the type of hemorrhage. The contribution of this research work can be stated as:

- Designed a lightweight automated system that would classify hemorrhage and non-hemorrhage accurately.
- Introduced a novel approach for classifying brain hemorrhage.
- Conducted a comparative study on brain hemorrhage classification.
- delivered an accurate prediction within a short time.

In Sect. 2, some previous researches relative to the work have been reviewed, Sect. 3 describes work methods that have been followed and information about the dataset, Sect. 4 contains the evaluation matrix, experimental result and analysis of the proposed method. At last in Sect. 5, the research work has been concluded.

## 2 Literature Review

Some remarkable works previously done on brain hemorrhage classification have been discussed in this section.

Napier et al. [6] proposed a CAD system that used different image processing techniques using different filters such as the Gaussian filter, the median filter, the bilateral filter and the Wiener Filter and morphological operations have been used to detect brain hemorrhage from CT scan. 36 head CT scan images were used to execute the method in this study. The proposal achieved an accuracy of 88.89%, a precision of 91.259%, a specificity of 94.4% and sensitivity of 94.4%. Srivastava et al. [7] proposed a way to classify hematomas in brain CT images by using support vector machine using 150 brain CT scan images. An average accuracy of 88% is gained from O-V-A SVM while O-V-O SVM gained 97% accuracy. Their method SVM (O-V-O) outperforms other classical machine learning approaches by some metrics.

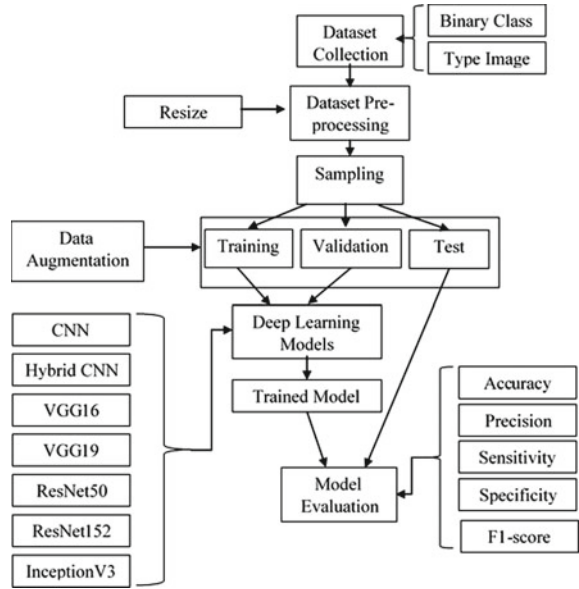
Urbancic et al. [8] they proposed a concept that was to tune transfer learning model based on grey wolf optimizer algorithm. The dataset had a total of 200 images from head CT—hemorrhage [11] dataset. They have used a CNN-based model, VGG-16 as their model architecture. After applying their GWOTLT method, they were able to secure 91% accuracy. Mushtaq et al. [9] proposed BHCNet to classify the existence of hemorrhage in human brain. During preprocessing, the images were resized to  $128 \times 128$  pixels and different image augmentations were applied to increase the size of the dataset. The images were split into train and test set by maintaining a 90% to 10% ratio. The performance of the proposed architecture was evaluated using metrics such as accuracy, precision, sensitivity, specificity and *F1*-score. The best model was found at 24 epochs with an accuracy of 95%.

Patel et al. [10] proposed an architecture with combination of VGG-like CNN with bidirectional LSTM. The CNN was initially trained using the dataset. The pre-trained CNN model was later used to initialize full model and two bidirectional LSTM layers were added. They have achieved 98% sensitivity and 78% specificity. End to end training without using pre-trained CNN model gives inferior result. Training the CNN and then training again after adding layers is a costly process.

## 3 Research Methodology

Image classification refers to the task of identifying the actual class of an image. Figure 1 shows the workflow of the classification task. Images in the head CT—

**Fig. 1** Flowchart of hemorrhage classification process



hemorrhage [11] dataset have been resized and split into training set, test set and validation set. Multiple augmentation techniques have been applied for the classification of brain hemorrhage. Different deep learning models have been used to perform the classification task. The trained models are evaluated with the help of test dataset. The predicted results have been compared with the actual outcomes using multiple evaluation metrics to test the performance.

**3.1 Data Preprocessing and Augmentation**

Images have been resized to  $224 \times 224$  for the model requirement. The size of the dataset is artificially increased by using augmentation. After augmenting the dataset, variations in the dataset can be found. Rotation of 0 to  $10^\circ$  clockwise, vertical and horizontal with a range of 0–0.2, zoom range of 0.2 and horizontal flip are used for the augmentation procedure. After augmentation, a total of 2400 images have been acquired, where training, validation and testing each set have 1536 (64%), 384 (16%) and 480 (20%) images, respectively. As model learns in training period, most of the features are used in training which is the reason for using large portion of dataset in training set.

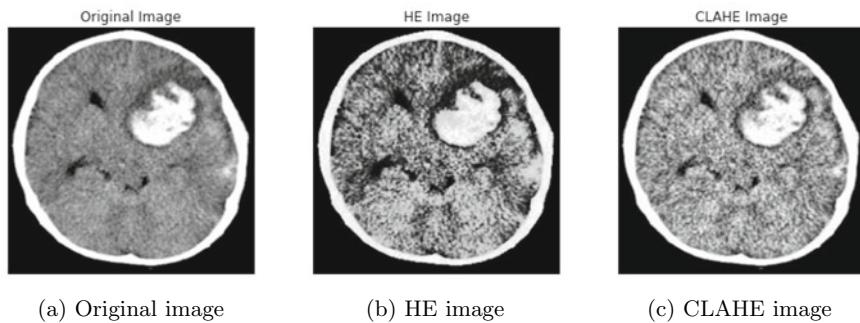


**Fig. 2** Hemorrhagic image and their output after histogram equalization and contrast limited adaptive histogram equalization

### 3.2 *Histogram Equalization (HE)*

Frequency of pixel intensity levels in an image can be represented by histogram. The contrast of an image can be tuned by using histogram equalization. It improves the image's contrast by stretching out the intensity range or spreading out pixel intensity that are most frequent which results in more contrast in low contrast sections of the image [12]. In Fig. 2b, equalized hemorrhagic image is shown.

### 3.3 *Contrast Limited Adaptive Histogram Equalization (CLAHE)*

The image is split into equal sized rectangular blocks, with each block undergoing histogram correction. Histogram formation, clipping and redistribution are all parts of histogram adjustment. The difference in CLAHE is that it uses a clip point to chop off the peak value in each block's histogram by limiting the contrast which are redistributed to each gray level [13]. Figure 2c shows how the original image is enhanced after applying contrast limited adaptive histogram equalization on it.

### 3.4 *Proposed CNN Model*

To better anticipate the hemorrhage, different CNN architectures have been designed with variation in convolution layers, max pooling layers, flatten layers and fully connected layers in all approaches. Proposed CNN model has five convolution layers with  $3 \times 3$  sized filters followed by ReLU activation function. After each convolution layer, the output is downsampled using max pooling layer having a  $2 \times 2$  pool. The output of last pooling layer has been flattened and then passed through the fully

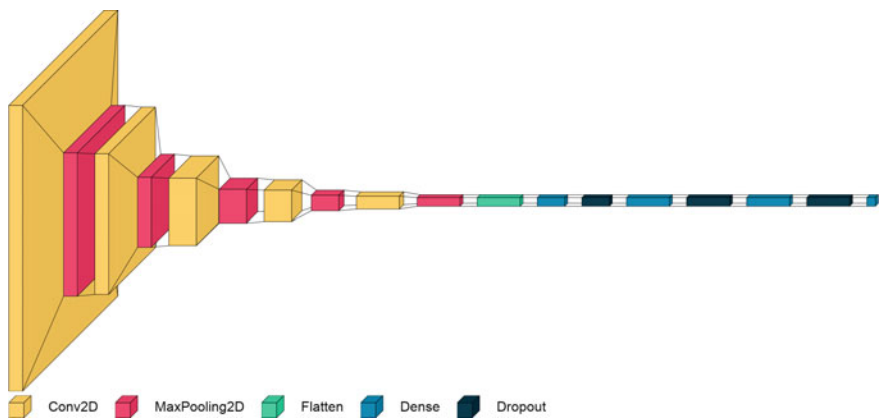


**Fig. 3** Proposed CNN architecture

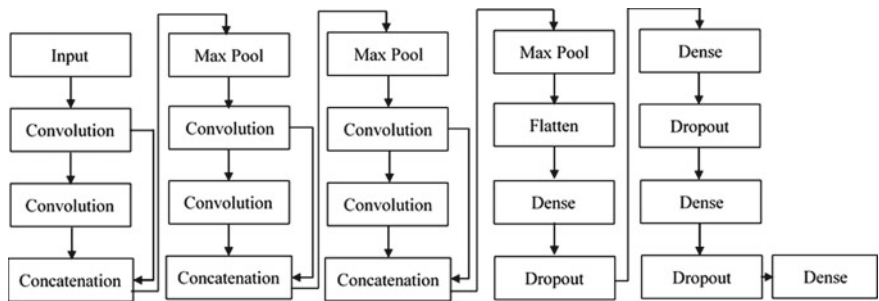


**Fig. 4** Hybrid CNN model

connected layers, followed by ReLU activation function. 40% of the neurons have been dropped out after each dense layer. In the output layer, sigmoid activation function has been used for the binary classification problem (Fig. 3).

**3.5 Hybrid CNN Model**

A hybrid CNN architecture has been built to experiment the models learning capability. The architecture has 6 convolution layers with same padding followed by ReLU activation function, 3 concatenation layers, 3 max pooling layers, 3 fully connected layers followed by ReLU activation function and 3 dropout layers. The concatenation layer concatenates the immediate output of the convolution layer with the previous convolution layer output. A max pooling layer is added after each concatenation. Sigmoid activation function is used in the output layer. Figure 4 illustrates the model architecture of hybrid CNN.

### **3.6 Pre-trained Models**

Transfer learning is the process of using feature representations from a previously trained model to avoid training a new model from the ground up. This previously trained models, known as pre-trained models, are typically trained on large datasets [14] and these are used as benchmark models as they are used in various applications of computer vision with great performance. These models reduce the training time and less erroneous. When there is a small training dataset, transfer learning works well by initializing the weights from pre-trained models. Previously acquired knowledge, which is weights, is used in training new dataset. VGG16 [15], VGG19 [15], ResNet50 [16], ResNet152 [16], InceptionV3 [17] pre-trained models from keras have been used to classify brain hemorrhage. These models are trained on ImageNet dataset and works well on image classification as it contains 1000 classes in the dataset.

## **4 Results and Discussion**

Building this brain hemorrhage classification system and training models, a computer having 2.5 GHz dual-core Intel Core i5-7200U processor with 8 GB RAM and NVIDIA GeForce 940MX has been used. Python and tensorflow have been used. Google Colaboratory with 12.7 GB RAM and 12 GB NVIDIA TESLA K80 GPU and Kaggle with 16 GB RAM and 13 GB NVIDIA TESLA P100 GPU are primary platform for model training and evaluation of the system.

### **4.1 Dataset**

Dataset plays an important role in any research work. It is tough to conduct medical research work as there is scarcity in finding appropriate data. In this study, the dataset, head CT—hemorrhage [11] is used that contains 200 images in which 100 images are of hemorrhagic brain and 100 images are of non-hemorrhagic brain. Deep learning model requires a large number of data with all possible variations to train the model. A small sized dataset is sometimes beneficial as it requires less time to train and the likelihood of overfitting reduces. But another risk in that the accuracy is not appeasable. The problem in the size of the dataset has been slightly solved by using data augmentation techniques.

### **4.2 Brain Hemorrhage Classification**

The results determine the performance of those models. In brain hemorrhage classification problem, confusion matrix, accuracy, precision, recall,  $F$ -1 score and speci-



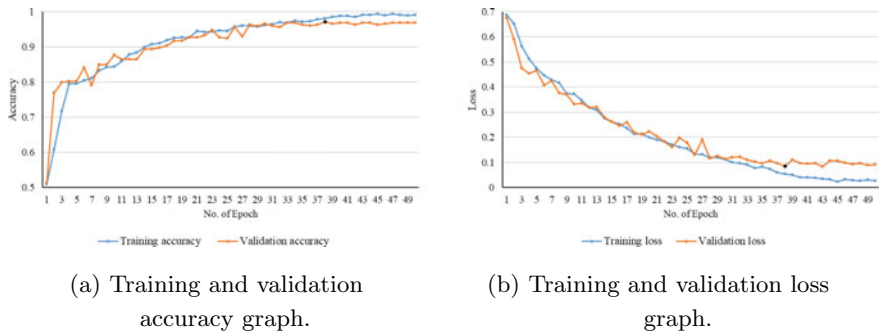


**Fig. 5** Training and validation performance of proposed CNN

**Table 1** Model descriptions of different CNN models

| Model        | No. of convolution layer | No. of dense layer | Dropout | No. of parameters | Filter size | Pool size | Activation function |
|--------------|--------------------------|--------------------|---------|-------------------|-------------|-----------|---------------------|
| CNN-1        | 3                        | 3                  | 0.40    | 697,057           | 3 × 3       | 2 × 2     | Relu and sigmoid    |
| CNN-2        | 5                        | 3                  | 0.40    | 336,865           |             |           |                     |
| Proposed CNN | 4                        | 3                  | 0.40    | 385,889           |             |           |                     |

ficity have been used as evaluation metrics. Figure 5a shows the model accuracy where the orange line is for validation and blue line is for training. Figure 5b shows the binary cross entropy loss graph where the blue line indicates training and orange line indicates the validation loss. The black dot has been added to show the point where the best model has been saved which is at 38th epoch by observing the validation accuracy criteria where the accuracy is maximum. In CNN approach, multiple CNN configurations have been observed to find out the best combination for the outcome. Table 1 gives different configurations of CNN models that have been tried out to find the best one. Number of hidden layer has been changed in each model where the number of filters also varied and in fully connected layers, and number of neurons have also been changed.

Figure 6 shows the accuracy, precision, sensitivity, *F1*-score and specificity of all models. Comparing between the proposed CNN and InceptionV3, the pre-trained InceptionV3 model has achieved the higher accuracy. Observing the other metrics of all nine models, proposed CNN predicts that someone has brain hemorrhage, which is correct 96.28% of the time. It is the precision of the model which is higher than other custom models and sensitivity of the proposed model which means actual hemorrhagic cases that are predicted as hemorrhage is 97.08%. The proposed CNN achieved 96.68% *F1*-score. Specificity of the proposed model is 96.25% which determines

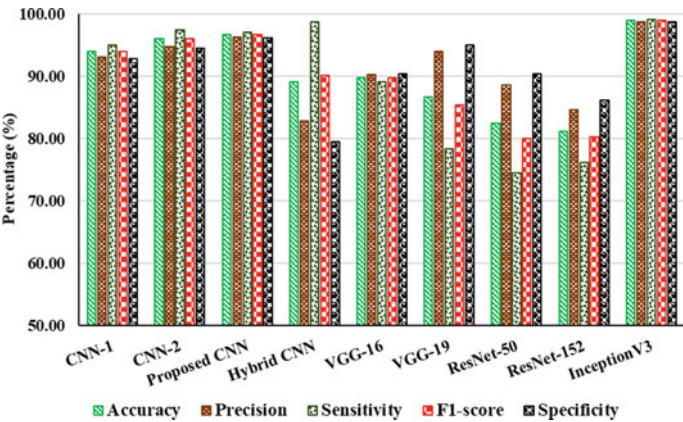


**Fig. 6** Comparison with respect to different evaluation metrics of CNN models on hemorrhage data

**Table 2** Performance comparison with respect to parameters

| Models              | Parameters     |
|---------------------|----------------|
| VGG16               | 17,926,209     |
| VGG19               | 23,235,905     |
| ResNet50            | 126,350,209    |
| ResNet152           | 161,133,441    |
| InceptionV3         | 21,853,985     |
| Hybrid CNN          | 13,265,889     |
| <b>Proposed CNN</b> | <b>385,889</b> |

Bold significance indicates performance of the methods in this paper

the non-hemorrhagic cases are predicted to be non-hemorrhage 96.25% of the time. Figure 7 visualizes the ROC curve analysis of the proposed model along with the other models. It can be seen that InceptionV3 model architecture functions admirably, exceeding the performances than custom trained model. But the pre-trained models are all heavyweight models in terms of total parameters. The number of parameters of the InceptionV3 model makes it computationally very expensive. On the other hand, the architecture proposed in this study is both a lightweight and a high-performing one for practical use.

As, it is seen from Table 2 that the number of parameters of the custom CNN is reasonably small in comparison to other models. Pre-trained models are trained with a large number of parameters and Inception v3 model has the lowest parameters among the other pre-trained models. To see whether the model performs better after increasing the number of parameters, the hybrid CNN model has been evaluated and the performance has not fulfilled the expected result. The models with a smaller number of parameters work better on the dataset. Inception V3 has less parameters than VGG16, VGG19, ResNet50 and ResNet152 and the model is trained with a

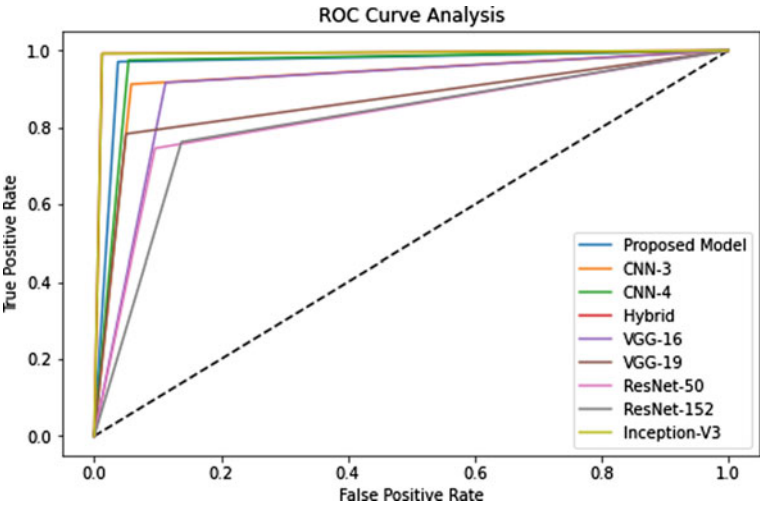


Fig. 7 ROC curve analysis

large dataset, and it performed better but in terms of parameters, the proposed model has got less parameters than this pre-trained one. Two variations on the dataset have been made after applying HE and CLAHE and fed to the proposed CNN. The result of both datasets could not outperform the previous result as there is a large area of low-intensity. Figure 8 shows the performance of proposed model on these datasets.

In Fig. 9, among 480 test images, four images are shown with the actual and predicted labels. The model predicted 233 hemorrhagic and 231 non-hemorrhagic images correctly. Total 16 images have been misclassified, 9 hemorrhagic images are predicted as non-hemorrhagic and 7 non-hemorrhagic images are predicted as hemorrhagic. Due to the complexities in CT scan images where head slices are taken from different direction, there are some images where white portions increases that lead the model toward misclassification. Also, in some images, the hemorrhagic part is diminutive to detect by the model.

Table 3 gives the validation accuracy of each fold. The average accuracy after five-fold cross validation is 96.46% which doesn't differ much from the models accuracy proving that the model has not been overfitted using this dataset. The comparison with other state-of-the-art works with the same dataset is shown in Table 4. It shows the similarities and differences among the previously proposed methods and the proposed method. Fivefold cross validation has been executed to check the performance.

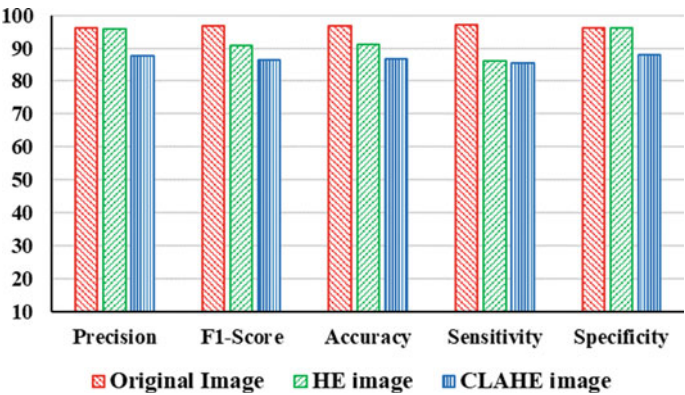


**Fig. 8** Performance comparison with original, histogram equalized and contrast limited adaptive histogram equalized dataset

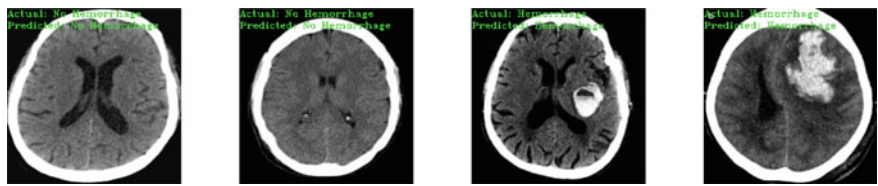


**Fig. 9** Sample of some predictions

**Table 3** Cross validation results of proposed model

| Iteration    | 1     | 2     | 3     | 4     | 5     | Average |
|--------------|-------|-------|-------|-------|-------|---------|
| Accuracy (%) | 96.88 | 96.25 | 95.83 | 95.63 | 97.71 | 96.46   |

**Table 4** Comparison with previous works with same dataset

| Literature        | Year | Technique                             | Accuracy (%) | F1-score (%) | No. of parameters |
|-------------------|------|---------------------------------------|--------------|--------------|-------------------|
| Greywolf [8]      | 2019 | VGG16 + Gray wolf optimizer algorithm | 91.00        | 91.45        | –                 |
| BHCNet [9]        | 2021 | CNN                                   | 95.00        | 95.23        | –                 |
| Transfer learning | 2022 | InceptionV3                           | <b>98.96</b> | <b>98.96</b> | 21,853,985        |
| Proposed model    | 2022 | CNN                                   | 96.67        | 96.68        | <b>385,889</b>    |

Bold significance indicates performance of the methods in this paper

## 5 Conclusion

In this study, different CNN models with variations in dataset and a comparison among the performances of pre-trained deep learning models and previous works with the proposed CNN model have been done. The most difficult part of this was working with very little dataset, whereas a deep learning model requires a large number of images. It is not easy to get medical image dataset easily and requires knowledge to understand the data which sometimes can be expensive. Data augmentation has added the solution to this problem by increasing the amount and variation of the dataset. The proposed method is capable of solving the brain hemorrhage classification task faster than other existing works. In the future, the dataset can be enhanced with more images. Multi-class classification can be done by using different types of hemorrhage. Also the localization of brain hemorrhage can be another inclusion. For improving the model performance, ensemble learning, long short-term memory (LSTM), gated recurrent units (GRU) with CNN can be tried out.

## References

1. Heit JJ, Iv M, Wintermark M (2017) Imaging of intracranial hemorrhage. *J Stroke* 19(1):11
2. Mohammed A et al (2022) Deep learning models for intracranial hemorrhage recognition: a comparative study. *Procedia Comput Sci* 196:418–425
3. Van Asch CJJ et al (2010) Incidence, case fatality, and functional outcome of intracerebral haemorrhage over time, according to age, sex, and ethnic origin: a systematic review and meta-analysis. *Lancet Neurol* 9(2):167–176
4. Kuohn LR et al (2020) Cause of death in spontaneous intracerebral hemorrhage survivors: multistate longitudinal study. *Neurology* 95(20):e2736–e2745
5. Haselsberger K, Pucher R, Auer LM (1988) Prognosis after acute subdural or epidural haemorrhage. *Acta Neurochir* 90(3):111–116
6. Napier J et al (2019) A CAD system for brain haemorrhage detection in head CT scans. In: *IEEE EUROCON 2019-18th international conference on smart technologies*. IEEE, pp 1–6
7. Srivastava DK, Sharma B, Singh A (2018) Classification of hematomas in brain CT images using support vector machine. In: *Information and communication technology for sustainable development*. Springer, Berlin, pp 375–385
8. Vrbancic G, Zorman M, Podgorelec V (2019) Transfer learning tuning utilizing grey wolf optimizer for identification of brain hemorrhage from head CT images. In: *StuCoSReC: proceedings of the 2019 6th student computer science research conference*, pp 61–66
9. Mushtaq MF et al (2021) BHCNet: neural network-based brain hemorrhage classification using head CT scan. *IEEE Access* 9:113901–113916
10. Ajay P et al (2019) Image level training and prediction: intracranial hemorrhage identification in 3D non-contrast CT. *IEEE Access* 7:92355–92364
11. Kitamura F (2022) Head CT—hemorrhage. <https://www.kaggle.com/felipekitamura/head-ct-hemorrhage>. [Online; Accessed 21 Mar 2022]
12. Das S, Riaz Rahman Aranya OFM, Labiba NN (2019) Brain tumor classification using convolutional neural network. In: *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*. IEEE, pp 1–5
13. Yakun C et al (2018) Automatic contrast-limited adaptive histogram equalization with dual gamma correction. *IEEE Access* 6:11782–11792

14. Krishna ST, Kalluri HK (2019) Deep learning and transfer learning approaches for image classification. *Int J Recent Technol Eng (IJRTE)* 7(5S4):427–432
15. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
16. He K et al (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
17. Szegedy C et al (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2818–2826

# Chapter 16

## An Automated Remote Health Monitoring System in IoT Facilities



Pallab Kumar Nandi and Mohiuddin Ahmad

**Abstract** Patient monitoring is one of the key features of the medical system, and an automated remote patient monitoring system is an innovative way to optimize the medical technology. The success of the proposed research is to build an advanced technology to provide Internet-based medical support to the patient in remote areas with no medical center. The doctor can view the patient's real-time data from a distance using the Internet and their mobile device. The device collects health status through a microcontroller sensor placed on the patient's body. Arduino UNO is used as a microcontroller. The system will enable the patient's doctors and their family members to be informed about the patient's instant health status through the Internet and provide complete medical enlightenment whether any critical medical situation occurs. The data collected can be used to analyze chronic illnesses such as heart attacks in the early stage itself which will also provide a convenient way to make decisions.

**Keywords** Patient monitoring · Microcontroller · Arduino UNO · Pulse rate sensor · Temperature sensor · Internet of things

### 1 Introduction

Health is the major element for every human. In Bangladesh, even around the world, the COVID-19 pandemic has taken the lives of many people, and many people could not go to the doctor for fear of contagion. If the doctor can observe the state of health

---

P. K. Nandi (✉) · M. Ahmad

Institute of Information and Communication Technology (IICT), Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

e-mail: [pallabnwu@gmail.com](mailto:pallabnwu@gmail.com)

M. Ahmad

e-mail: [ahmad@eee.kuet.ac.bd](mailto:ahmad@eee.kuet.ac.bd)

M. Ahmad

Department of Electrical and Electronic Engineering (EEE), Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

of the patient from a long distance via the Internet, then the patient is much more possibility to recover. IoT is the process of linking the instrument to the Internet utilizing sensors and networks [1, 2]. Remotely, patient monitoring with real-time health condition detection is vital for telemedicine.

In this paper, an automated remote health monitoring system (ARHMS) can be able to detect the patient body temperature and heart pulse rate. A physical display has been added to the system, displaying a unit of body temperature in Fahrenheit and a pulse rate in bpm [3]. The number of heartbeats per minute is the heart pulse rate. The average comfortable heart rate is about 70 bpm for adult males and 75 bpm for adult females. Among women 12 years of age or older, the rate of heart incompatibility with men is usually higher [4]. A person's body temperature is scientifically determined only by the sum of body heat and heat radiated by the body. In normal healthy adults human, it can be between 97.8 °F (36.5 °C) and 99 °F (37.2 °C) [5]. Symptoms of the physical illness include a decrease or increase in body temperature [6]. The main objective of the project is to design and implement a remotely patient health monitoring system that uses sensors to monitor the patient's health and uses the Internet to inform about the health conditions of the patient's doctor and family members [7, 8].

This paper originates into seven sections. Section 1 represents the introduction. Section 2 discusses related work. The hardware components are focused on in Sect. 3. System design and implementation are discussed in Sect. 4. Section 5 is presented the security of the device. Section 6 is represented the experimental results. Section 7 concludes the paper. In Fig. 1, the whole process is shown through a block diagram.

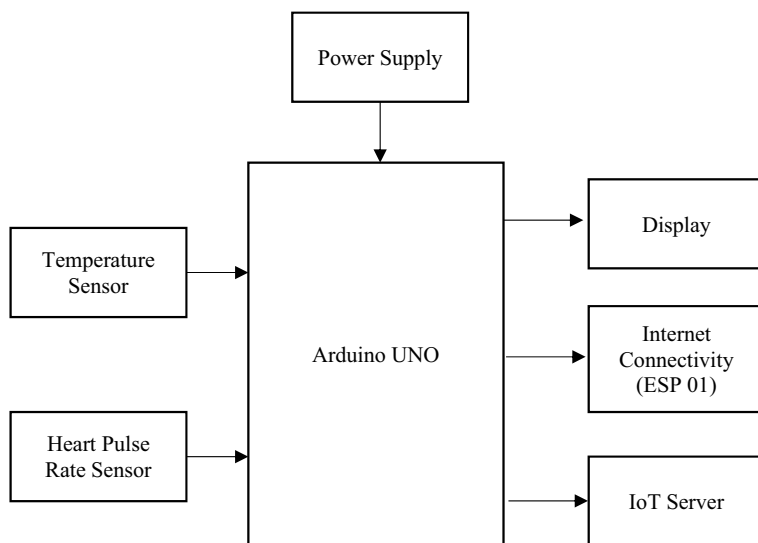


**Fig. 1** Block diagram of the ARHMS system



## 2 Related Work

There have been several significant developments in medical science using IoT to observe patients' health. The research related to these areas is described as follows.

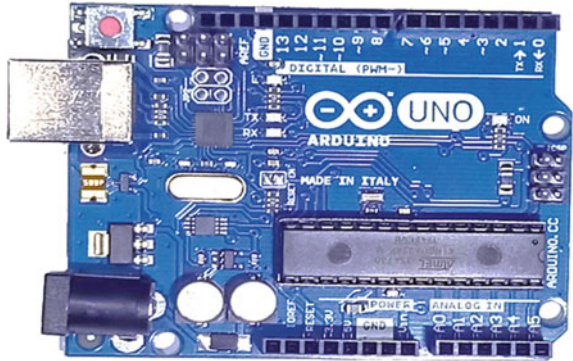
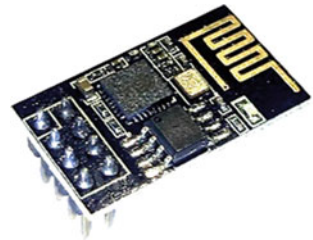
Milon Islam et al. [9] created a healthcare monitoring system in an IoT environment that can monitor a patient's primary parameters such as heartbeat, body temperature, room temperature, and CO<sub>2</sub> sensor in an IoT network. For this ground, the system used ESP32 as a junction with the processing device. The device did not use any direct display for offline views. Valsalan et al. [10] proposed a healthcare monitoring kit that includes a wearable sensor node to track patient health circumstances and uses the Internet to communicate the health situation with doctors. It was a complex system including a base station with a wireless communication system. Banka et al. [11] implemented a model which collects data and records the data using the Raspberry Pi element. They use GSM components to communicate with the Internet. The model is a little costly but a doctor can access the patient information. Hameed et al. [12] present an approach including a fuzzy neural network (FNN)-based system that collects data and processes via the fuzzification rule. It has a cloud-based data storage process that the researcher can get access to the previous data. AlMotiri et al. [13] proposed a mobile health system that consolidates real-time data from victims using a mobile device and stores it on a network server connected to the Internet that enables access to only an appointed person. This data can be used to diagnose a patient's health status, and it is obtained using a network of several wearable devices and body sensors.

## 3 The Hardware Components

Some electronics modules and microcontrollers are being used to develop the system. Here is a brief description of the elements.

### 3.1 *Arduino UNO*

An Arduino UNO is the most useful and popular microcontroller based on ATmega328p which is shown in Fig. 2. The Arduino UNO is used as a microcontroller to control the other sensors and modules for collecting medical data. As a microcontroller, it controls the electrical voltage to the sensors and display modules and also continuously collects and processes the sensor's value and view to the display.

**Fig. 2** Arduino UNO R3**Fig. 3** ESP8266-01

### 3.2 *ESP8266-01*

The ESP-01 is a module for connecting any IoT device to the Internet via TCP/IP protocol which is shown in Fig. 3. The module can be used as both an Internet access point and a signal station, so it can easily exchange data through the Internet, and it also makes the Internet of things more accessible and convenient. ESP-01 is integrated with the ESP8266 chip which can communicate with other Wi-Fi devices.

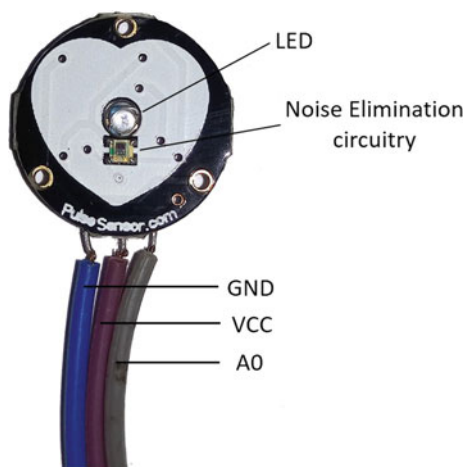
### 3.3 *LCD Display*

An LCD is an electronics module that uses liquid crystals to create visible images which is shown in Fig. 4. It has 16 characters columns in 2 rows, and each character consists of a  $5 \times 7$  pixel matrix. This display is very suitable for Arduino and other microcontrollers. It has 8 data pins with I2C display communication which consumes low power.



**Fig. 4** 16 × 2 LCD display

**Fig. 5** Heart pulse rate sensor



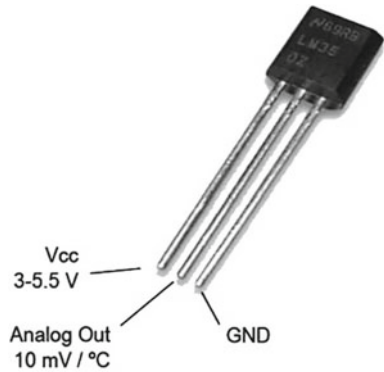
### 3.4 Heart Pulse Rate Sensor

When the heart circulates blood, the pulse rate sensor measures the volume of blood vessels through light waves. The emergency conditions are measured by the pulse rate sensor. Cardiac arrest, vasovagal syncope, and pulmonary embolism are detected by abnormality of heart pulse rate. The sensor in Fig. 5 works when the object is placed in front of the optical panel. Arduino processes the heart rate reading in the bpm unit.

### 3.5 Temperature Sensor

The function of the temperature sensor is to detect body temperature, heatstroke, and fever. In Fig. 6, the temperature sensor LM-35 has 3 pins the  $V_{cc}$  and GND pin

**Fig. 6** LM-35, analog temperature sensor



pass the voltage for the sensor, and the other pin outputs an analog signal which is a process by the microcontroller.

### 3.6 Power Source

For this entire device active, it requires a 9 V DC power supplier. A 9 V rechargeable battery keeps the whole system running, and a power adapter is used to charge the battery. Lithium-ion batteries have been used to make the device more portable.

## 4 System Design and Implementation

All the sensors are wired connections to the microcontroller that are used to collect data by collecting physiological symptoms from the patient's body. The collected data are then processed via the microcontroller Arduino UNO, and the ESP32-01 module sends it to the gateway of the Webserver. ThingSpeak is capable of displaying graphical explanations and collected reports and is used for illustrated displays by Web users. ThingSpeak views the present report and proceedings. Data can be viewed and collected from the wave server by any device capable of connecting to the Internet. The wave server is a channel-based scheme, which requires a password when accessing data and there the data are displayed graphically.

In the flowchart of Fig. 7, the microcontroller apprises the reading of the sensor. Both the heart rate sensor and body temperature sensor put forward the collected value to the microcontroller. The heart rate sensor provides the value in the bpm unit, and the temperature sensor provides the value in the Fahrenheit unit. The microcontroller processes the data according to the program and sends it to the Webserver. In this way, the health of the patient can be known remotely through the Webserver. If the heart pulse rate is not normal range, then the system can smartly send the notification to the

selected person like a doctor or family member. Similarly, if the body temperature is not normal range, then the system can send the notification wisely.

The user prototype has been built based on the circuit diagram, and it shows the entire connection method as shown in Fig. 8. In the user prototype in Fig. 9, here, the heart pulse sensor and body temperature sensor are attached to the patient’s body, and the sensor output is connected with Arduino UNO. The Internet communication modules ESP-01 and LCD are connected with Arduino. The microcontroller processes the sensor value and shows the result on the LCD, and the patient’s real-time report will be sent to the Webserver via the ESP-01 module. The comparison of the existing research and the proposed research is shown in Table 1.

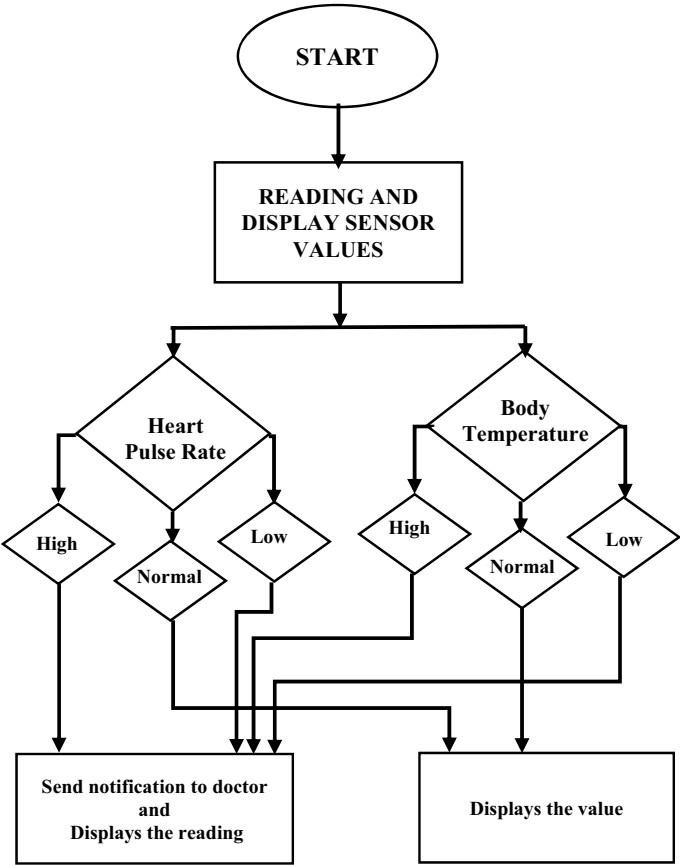


Fig. 7 Working strategy of the developed ARHMS scheme

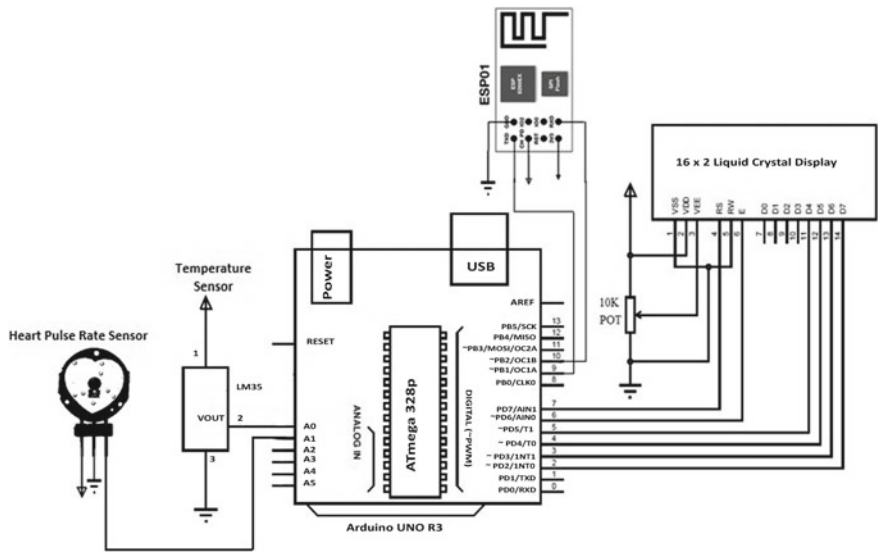


Fig. 8 Complete connection diagram

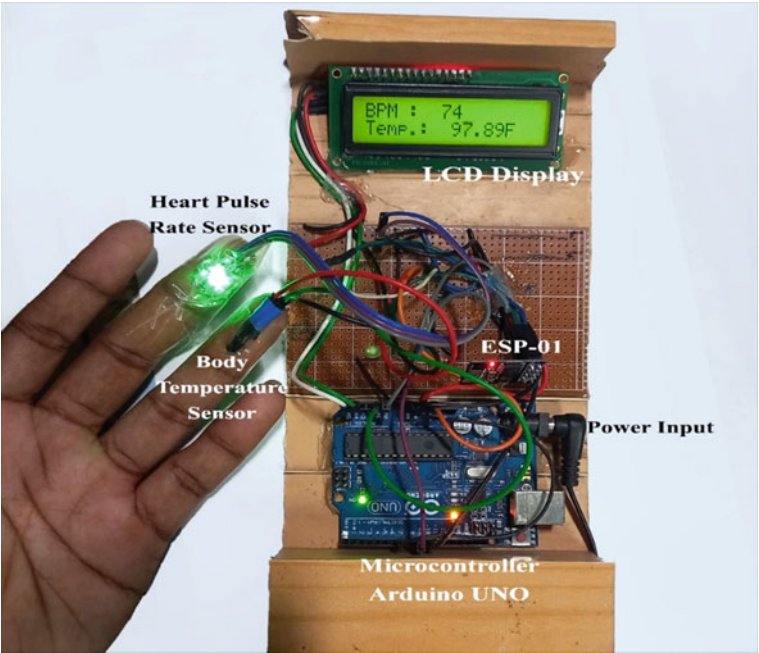


Fig. 9 User prototype of the proposed system

**Table 1** Comparison between existing researches versus proposed research

| S. No. | Author's name          | Existing researches                                                                                                                                                                                                    | Proposed research                                                                                                                                                                                                                                                                         |
|--------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Milon Islam et al. [9] | The healthcare monitoring system built-in ESP32 microcontroller for viewing patient health conditions online. But there is no offline display for local view                                                           | In this work, the patient monitoring system works on not only online facilities but also offline mode. The patient's medical data can be viewed on the prototype built-in display, and it can be shown online for remote access                                                           |
| 2      | Valsalan et al. [10]   | They developed a wearable sensor node for monitoring patients. In their complex system, the patient's basic parameters are collected via sensor node and sent into the base station for monitoring the patient         | In this developed system, the use and installation are easy. The collected patient data are transferred via Wi-Fi technology no need to install a base station. The proper authority can view the real-time data without any hassle                                                       |
| 3      | Banka et al. [11]      | They implemented a smart model which collects data from sensor nodes and records the data using the Raspberry Pi element. It communicates the Internet via GSM components                                              | In this proposed system, the data are collected from sensor terminals that are attached to the patient's body. Webserver cloud space is used to record the medical data. An Arduino UNO and ESP-01 module are used to communicate with the Webserver                                      |
| 4      | Hameed et al. [12]     | They proposed a smart healthcare architecture based on a fuzzy neural network that collects patients' data and views it via an online-based android application                                                        | In this smart technology, the patient's medical data are viewed in a prototype display and also view real-time data in a Webserver. It has also capable to store the previous data in cloud storage                                                                                       |
| 5      | AlMotiri et al. [13]   | They described the mobile health technology to monitor patients' body symptoms. They also focus on medical data processing and architect it on three-layer that are data collection, data processing, and data storage | In this technology, the medical data are collected and processed inside the microcontroller unit, and the data are sent into the cloud-based IoT server using an Internet connection. The medical data are accessed by the proper authority, and the data can be used for future research |

## 5 Security of ARHMS Device

The patient has the right to control access to personal health data. For this reason, the system has the administration id and password login system in Web-based access. So that only the authorized medical crew can access and view the data. However, system security needs to be adjusted with availability. Unauthorized access is detrimental to the system, which can cause problems for patients and users. But if the patient's data are not accessed at the right time in case of an emergency, it can be life-threatening for the patient.

## 6 Experimental Results

The proposed approach was tested with different people of various ages in different environments. In the case of tests, for heart rate and body temperature, we compared the observed values produced by the system and actual values observed from the traditional method. The approximate success rate between the observed data and the actual data is over 96% in all the cases. Patients can check their health status at home at any time and go to the hospital if they need to. This can be done using this system, the results of which can be seen from any part of the world. As it is a prototype model, the system displays real-time values of several health parameters and mimics how it can be successfully applied in real life. Physicians can use the data chart of the patient's body condition to study and visualize the effects of the drug or other things.

Table 2 and Fig. 10 have illustrated here to compare the difference between the report obtained by the developed system from the actual report. Six different people were tested separately for heart rate, and their pulse rate reports were collected. The error percentage is calculated between the observed data taken by this developed system and actual data taken by an analog machine.

**Table 2** Heart rate data are collected by the developed system (observed) versus traditional analog machine (actual)

| Subject | Observed data (bpm) | Actual data (bpm) | Error (%) |
|---------|---------------------|-------------------|-----------|
| S1      | 69                  | 70                | 1.43      |
| S2      | 72                  | 74                | 2.70      |
| S3      | 75                  | 76                | 1.31      |
| S4      | 74                  | 73                | 1.37      |
| S5      | 76                  | 78                | 2.56      |
| S6      | 82                  | 80                | 2.50      |



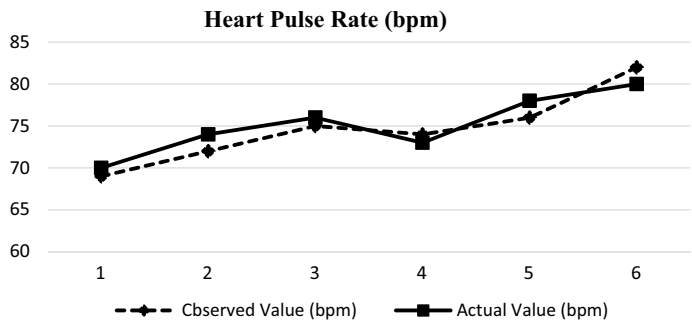


**Fig. 10** Accuracy test result for heart pulse rate

The body temperature collected by six various humans of different ages is shown in Table 3 and Fig. 11. The error percentage of body temperature is calculated between this prototype sensor and the traditional mercury thermometer.

A snapshot of the collected continuously real-time data on the Webserver (ThingSpeak) is pictorial in Fig. 12. Here, field 1 in the ThingSpeak Webserver shows the temperature data graph with temperature value on the Fahrenheit scale, and in field 2, the Webserver shows the heart pulse rate graph with bpm value.

**Table 3** Body temperature data are collected by the developed system (observed) versus traditional analog machine (actual)

| Subject | Observed data (°F) | Actual data (°F) | Error (%) |
|---------|--------------------|------------------|-----------|
| S1      | 97.9               | 97.5             | 0.41      |
| S2      | 98.2               | 97.8             | 0.41      |
| S3      | 97.6               | 97.7             | 0.10      |
| S4      | 98.8               | 98.0             | 0.82      |
| S5      | 97.4               | 97.6             | 0.20      |
| S6      | 97.0               | 97.4             | 0.41      |

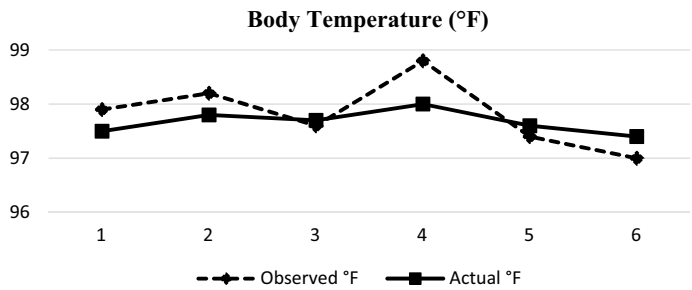


**Fig. 11** Accuracy test result for body temperature

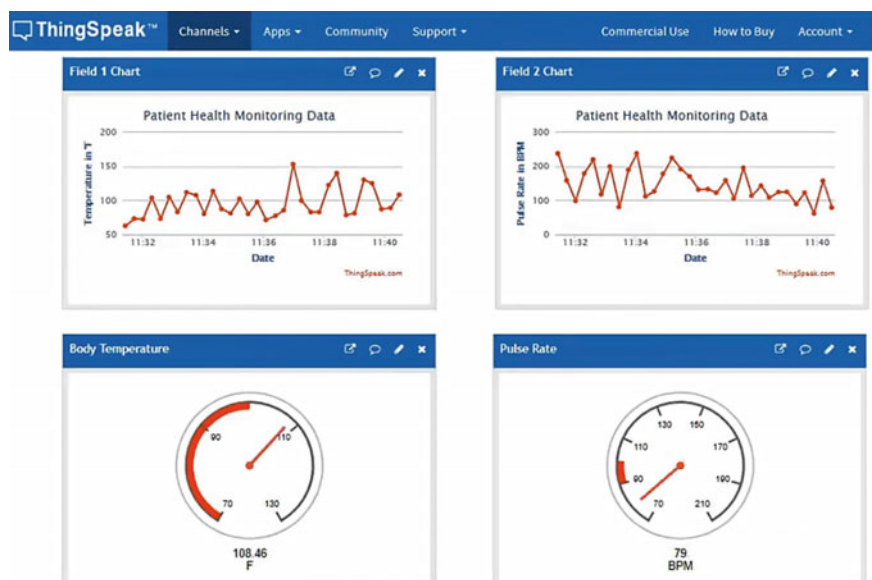


Fig. 12 Snapshot of ThingSpeak webserver

## 7 Conclusion

The system has launched initially smart healthcare facilities to monitor patients' early vital signs, such as heart rate and body temperature. The computerized low-cost modern remotely patient monitoring system can also be able to transmit patient body symptoms smartly. The success rate between the observed data and the actual data in the developed system is more than 96%, which is very optimistic. Under the system, authorized medical crews will be able to monitor medical data in real time when patients are performing medical examinations outside the medical center or at home. The system can assist nurses and doctors in the event of a COVID-19 epidemic because it is possible to collect and analyze the symptoms of many patients in a short time. The design and installation process of the prototype are very simple and also user-friendly. The approach is very favorable in treating contagious diseases like novel coronavirus (COVID-19). The proposed system will make better the remaining healthcare arrangement which could save many lives from death. Due to the large amount of medical data collected and analyzed, this system can provide a warning of any major infectious disease.

## References

1. Sobhan Babu B, Srikanth K, Ramanjaneyulu T, Lakshmi Narayana I (2016) IoT for healthcare. *Int J Sci Res (IJSR)* J 2(5):99–110
2. Vishnu S, Jino Ramson SR (2021) An internet of things paradigm: pandemic management (incl. COVID-19). In: 2021 international conference on artificial intelligence and smart systems (ICAIS)
3. Pradhan B, Bhattacharyya S, Pal K (2021) IoT-based applications in healthcare devices. *J Healthc Eng* 2021(Article ID 6632599)
4. Joseph S, Shahila DFD, Patnaik S (2019) IoT based remote heartbeat monitoring. In: 2019 international conference on advances in computing, communication and control (ICAC3), pp 1–5. <http://doi.org/10.1109/ICAC347590.2019.9036735>
5. Kodali RK, Swamy G, Lakshmi B (2015) An implementation of IoT for healthcare. In: 2015 IEEE recent advances in intelligent computational systems (RAICS), 10–12 Dec 2015, Trivandrum
6. Tyagi S, Agarwal A, Maheshwari P (2016) A conceptual framework for IoT-based healthcare system using cloud computing. In: 2016 6th international conference—cloud system and big data engineering (confluence)
7. Wu T, Wu F, Redouté J, Yuce MR (2017) An autonomous wireless body area network implementation towards IoT connected healthcare applications. *IEEE Access* 5:11413–11422. <http://doi.org/10.1109/ACCESS.2017.2716344>
8. Selvaraj S, Sundaravaradhan S (2020) Challenges and opportunities in IoT healthcare systems: a systematic review. *SN Appl Sci* 2:139
9. Milon Islam Md, Rahaman A, Islam R (2020) Development of smart healthcare monitoring system in iot environment. *SN Comput Sci* 1:185. <http://doi.org/10.1007/s42979-020-00195-y>
10. Valsalan P, Baomar TAB, Baabood AHO (2020) IoT based health monitoring system. *Multi Rev J* 7(4). ISSN 2394-5125
11. Banka S, Madan I, Saranya SS (2018) Smart healthcare monitoring using IoT. *Int J Appl Eng Res* 13(15):11984–11989. ISSN 0973-4562
12. Hameed K, Bajwa IS, Ramzan S, Anwar W, Khan A (2020) An intelligent IoT based healthcare system using fuzzy neural networks. *Hindawi Sci Program* 2020:Article ID 8836927
13. AlMotiri SH, Khan MA, AlGhamdi MA (2016) Mobile health (m-health) system in the context of IoT. In: 2016 4th international conference on future internet of things and cloud workshops, pp 39–42. <http://doi.org/10.1109/W-FiCloud.2016.24>

# Chapter 17

## Surface Reflectance: A Metric for Untextured Surgical Scene Segmentation



Shahnewaz Ali, Yaqub Jonmohamadi, Yu Takeda, Jonathan Roberts, Ross Crawford, Cameron Brown, and Ajay K. Pandey

**Abstract** Segmentation is a process to understand scene context captured by a camera, and it is commonly used to solve many robotic problems such as localization and tracking. The utility of semantic segmentation in robot-assisted minimally invasive surgery (MIS) is increasing as it can provide situational awareness. In this work, we present a new approach for image segmentation for knee arthroscopy, by capturing the evolution of tissue-dependent spectral reflectance within the cavity. Our approach relies on using spectral sensitivity of the image sensor that provides a cue to estimate surface reflectance from color tristimulus values. We demonstrate that light reflectance conveys pivotal information about the tissue surfaces that could be used to segment surgical frames which are difficult to visualize as limited scene context, and lack of texture and features. The methodology conducted in cadaveric experiments confirms the robustness of reconstructed spectral reflectance in binary segmentations of surgical sites. To the best of our knowledge, this is the first use of surface reflectance as a segmentation tool for identifying tissue types in endoscopic images.

**Keywords** Spectral reflectance · Sensor spectral sensitivity · Segmentation · Decision tree classifier · Support vector machine · Knee arthroscopy · MIS · Robotic-assisted surgery

---

S. Ali (✉) · Y. Jonmohamadi · Y. Takeda · J. Roberts · R. Crawford · A. K. Pandey  
School of Electrical Engineering and Robotics, Queensland University of Technology, Gardens Point, Brisbane, QLD 4001, Australia  
e-mail: [shahnewaz.ali@hdr.qut.edu.au](mailto:shahnewaz.ali@hdr.qut.edu.au)

C. Brown  
Medical Engineering Research Facility, Photonics and Mechanics of Biomedical Materials, Queensland University of Technology, Gardens Point, Brisbane, QLD 4001, Australia

## 1 Introduction

Scene segmentation is a well-studied area in computer vision, where each pixel is grouped into a specific class. Segmentation has become fundamental to solve many robotics problems. The advent of deep neural networks, these approaches to segmentations have very effective. However, the success is still limited in the context of robot-assisted minimally invasive surgery (MIS). This is primarily due to the unstructured environments that characterize the complexity associated with visualizing anatomical details in MIS [1]. For instance, stereo vision, segmentation, and localization of surgical tools inside a knee cavity remain a formidable technological challenge [1–4]. Recently, deep learning techniques have been successfully applied in laparoscopic video sequences [5–7]. However, only, a limited literature describes segmentation in the context of knee arthroscopy, a well-established MIS procedure used for bone joint ailments [1–3, 8, 9]. Arthroscopic video frames basically record different tissue types, which exhibit smooth texture-less regions. Moreover, the surgical planning has to be unique to patients' conditions. Frames require further enhancement techniques [10, 11].

In this article, we present a new approach toward auto-labeling of tissue types by using the surface reflectance information present in 2D images based on the phenomena of light-matter interactions. It is unique to the material interacting surface and is referred to as reflectance that has many diagnostic and therapeutic applications [12].

Depending on the intrinsic construction of tissue, optical properties are different for each tissue type. Considering this hypothesis, we have evaluated spectral reflectance as an alternative to pixel-level information and examine its potential for surgical scene segmentation. Therefore, in this study, instead of standard optical tools such as spectrometer, we reconstruct reflectance spectra from the sensor responses. The reconstructed reflectances were directly compared with ground truth spectra obtained from the spectrometer. In the subsequent phases, we conducted a standard knee arthroscopic procedure that was performed on a human cadaver, and we assess the feasibility of the surface reflectance in that constrained and confined space. Furthermore, we include underwater environment in our experimental setup that matched with clinical flow. After the feasibility experiments of capturing surface reflectance from RGB pixels, we developed a model for segmenting surgical scenes and assess pros and cons of this essential metric to achieve long-standing demand for contextual awareness in the context of knee arthroscopy. The methods based on this study have been published in MCCA 2021 and IJCACI 2021 [2, 13].

## 2 Background Study

Spectral reflectance is a quantitative measure that expresses how a material responds at different wavelengths of light photons. Reconstruction of reflectance from red,



linear estimation method in such a way that the estimated reflectance gets minimum error defined by Eq. (4);

$$\mathbf{e} = |\mathbf{r} - \mathbf{r\_est}| \quad (4)$$

In Eq. (4),  $\mathbf{r}$  is the actual reflectance, and  $\mathbf{r\_est}$  is the estimated reflectance. The main objective is to estimate camera sensitivity from known responses. Estimation of the translation matrix is an iterative process. The system requires an intensive training process with  $n$  number of measured spectrum and associated RGB values.

This study follows the data-driven-based spectral reconstruction method that is originally proposed by Otsu et al. [19]. Their spectral reconstruction is achieved by defining the relation between the colors and the weights of basis functions. Hence, the measure spectra  $s$  is represented as [19, 20];

$$s = \sum_{j=1}^n w_j b_j \quad (5)$$

Otsu et al. in their research used tristimulus colors (in CIE XYZ and RGB space), and it is confirmed that their proposed method can reconstruct spectra using three basis functions which makes the above equation as follows [19],

$$s = \sum_{j=1}^3 w_j b_j \quad (6)$$

Therefore, Eq. (6) confirms the ability to reconstruct reflectance from RGB color space which can be expressed in three basis function. Data-driven approach calculates weight of the basis function from given color tristimulus values and corresponding reflectances at minimum reconstruction error.

Apart from the applications of reconstruction reflectance in bio-photonics, to our best knowledge, we are the first where we conducted scientific experiments, study, and assess the reconstructed reflectance for segmentation surgical scenes in an uncontrolled lighting environment inside the intra-articular cavity.

### 3 Cadaver Experiment and Data Collection

Ocean Optics Flame Spectrometer equipped with a tungsten halogen lamp as a light source was used during the cadaver experiments. The experiments were carried out with human cadaver at QUT's Medical Engineering Research Facility (MERF) to confirm our opening hypothesis. The surface reflectance that we obtained from calibrated spectrometer is termed as ground truth.

In Fig. 2, the left image represents surface reflectance of bone cartilage at three distinct light intensities. The red spectra express the ground truth spectral reflectance and corresponding reconstructed spectral reflectance (in green and blue). It confirms that the adopted method is able to accurately reconstruct surface reflectance which is independent to light intensity. Hence, it is concluded that at different light intensity or near and far tissue distance from the lighting conditions that is the imaging system, the surface reflectances are confined to a same space although their color information varies. Similarly, the right image of Fig. 2. represents the ground truth (GT) spectral reflectance of tissue-type ACL (in red) and its corresponding reconstructed reflectance from RGB color value obtained from video frame (in blue) under CIE illuminant series A. This evidence confirms that the endoscopic video frames preserve surface information among the hard and soft-tissue types.

The knee anatomy was suffering from osteoarthritis; bone cartilages were slightly degenerated in some parts. To build the confidence of the opening hypothesis, only, healthy cartilage is considered. Figure 3a represents the spectral reflectance of ACL and bone cartilage and their corresponding reconstructed spectrum. It is apparently demonstrating that, within the spectral band of 450–600 nm, the reflectance represents a clear difference. ACL reflects a significant amount of photon energy with respect to cartilage tissue. In Fig. 3b, the knee anatomy and observed reconstructed spectral reflectance are depicted. The reflectance spectra were reconstructed from RGB pixel value obtained from arthroscope which was introduced to the knee cavity through a small incision. Noticeable differences within the cartilage surface appeared due to some part of the femur contained degenerated cartilage tissue.

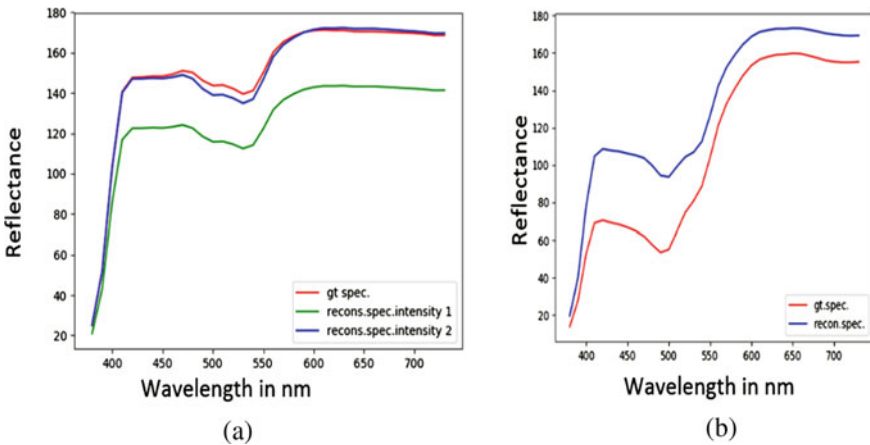
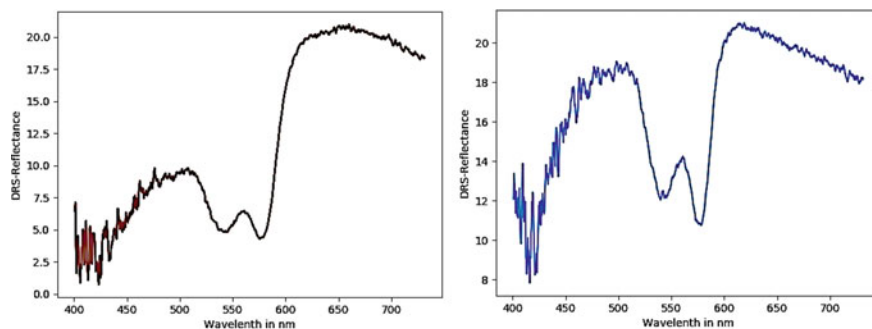
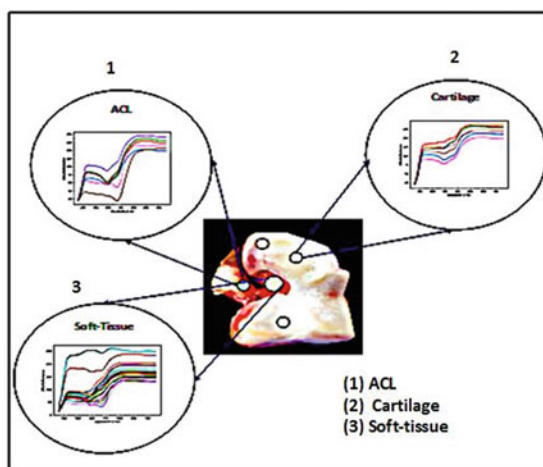


**Fig. 2** **a** Ground truth spectral reflectance (gt in red) and corresponding reconstructed spectral reflectance from RGB color at two distinct light intensities (in green and blue), **b** the ground truth (gt) spectral reflectance of tissue-type ACL (in red) and its corresponding reconstructed reflectance from RGB color value obtained from video frame (in blue)





(a)



(b)

**Fig. 3** Reflectance of the anatomical structure of knee joint. **a** Represent the ACL (left) and cartilage (right). **b** Upper left corner represents reflectance's of the ACL, upper right corner represents tissue-type cartilage, and bottom left corner represents different unstructured soft-tissue types such as fat, synovium and dense collagen

Our experiments include images captured by prototype stereo arthroscope that contains two miniaturized camera sensors for stereo vision, model-muC103A, with two micro-white light-emitting diode (LED) model-T0402W as lighting source.

Figure 4 shows the effect of different illuminant in creating reflectance data of the same tissue type. In that context, the spectrum generated, after performing a white balance procedure, by our camera sensor using white micro-LEDs comes very close to that obtained from a commercial arthroscopic video frames captured using tungsten halogen lamp. In order to confined into same color space that is CIE illuminant series A which closely supports tungsten halogen lamp, an automatic white balancing is performed using the method proposed by Afifi et al. [21]. The white balancing

process also adjusts the temperature of the color. The comparative result shows in Fig. 4. The spectra of the two illuminants are shown in Fig. 5. Moreover, arthroscopy is performed in underwater environment. Water acts as a diffuser; light scattering causes color attenuation; therefore, haze images are observed. An evaluation has been performed following the state-of-the-art color correction methods [21–23]. From our experiment, it is confirmed that color correction process does not have sufficient effect on reflectance provided that the camera and surface always reside at close proximity (approx. 10 mm). It is also validated using color checker box placed in a water tank at different distances. The measured reflectance varies only at intensity amplitude level (Fig. 2), but the resultant spectral distribution is preserved.

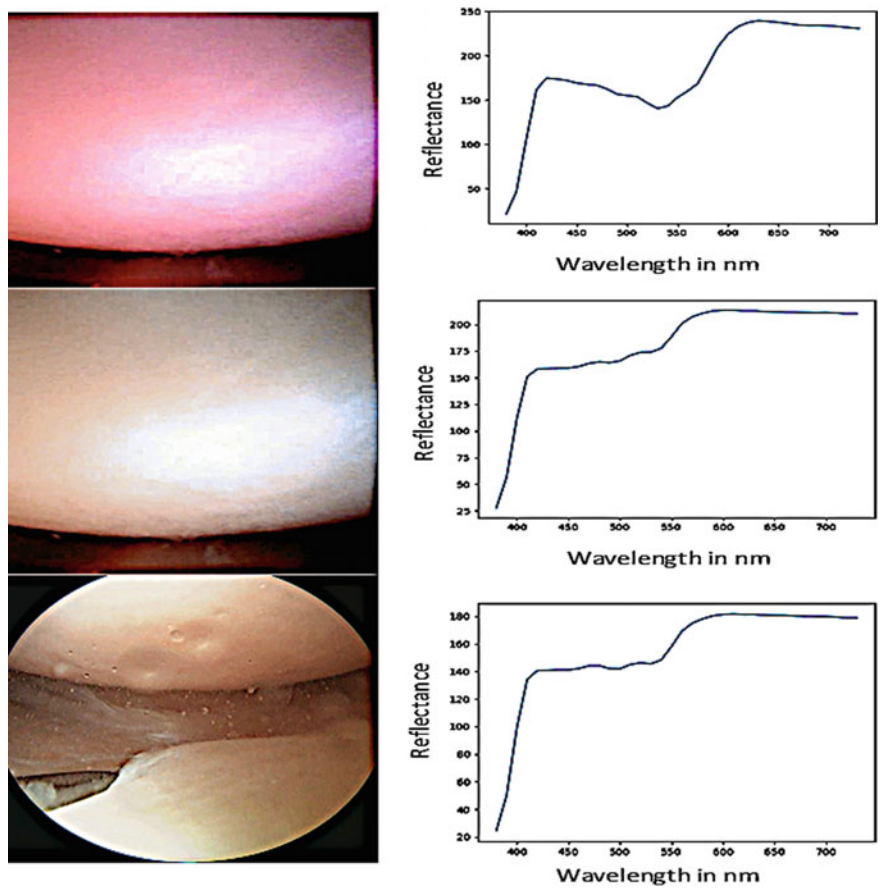


**Fig. 4** Image frames and corresponding reconstructed spectral reflectance from RGB color under two different illuminants. Top image represents the raw RGB; middle row represents white balanced image and its corresponding reflectance. Bottom image taken from a commercial arthroscope (Stryker). Reflectance data presented in the right panel show that after correcting white balance, the resultant spectrum is confined to same color space

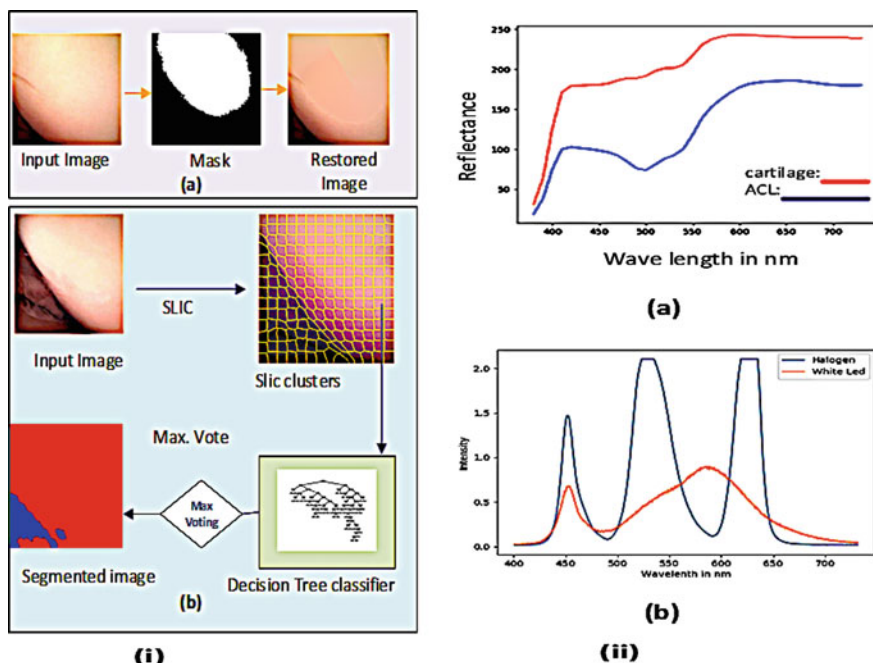


**Fig. 5** Figure (i) represents the workflow for MIS-binary segmentation methodology. After correcting the saturated regions, the restored image is clustered into superpixels level. Reflectances are reconstructed from a set of pixels in superpixel clusters. Each reflectance is then classified by decision tree classifier. Single pixels can be affected by random noise; hence, classification is done over a set of pixels obtained from each superpixels cluster. Final label is selected by evaluating the maximum class frequency in a superpixel set. Figure (ii)**a** represents the spectral response of tissue-type ACL (blue) and cartilage (red) estimated under specular reflection. Substantially, the spectral characteristics of these tissue types are identical. Figure (ii)**b** represents the light spectrum for white led (in orange) and halogen lamp (in blue) used during the caver knee arthroscopy. Under both illumination, the resultant reflectance represents identical spectra

## 4 Segmentation Methodology

To quantify the importance of surface reflectance for recognizing untextured tissue types, a binary segmentation pipeline has been proposed in this work follows basic three stages: (i) image preprocessing, (ii) cluster image into superpixel level, and (iii) segmentation process. Image preprocessing stage restores degraded images as described in Fig. 5. Scene lighting condition can degrade pixel quality [10]. All the three color channels can get saturated under excessive lighting conditions. In this work, we used image inpaint technique to correct saturated pixels proposed by Telea et al. as an example of the importance of such a method [24]. Learning-based method although outperforms over such classic methods, however, due to lack of sufficient amount of ground truth clinical image, it was not possible to adopt such techniques, and it is beyond the context of this study [25]. Telea inpaint method requires a mask

image to identify degraded pixels. To identify saturated pixels, we used the threshold value from our previous study [10]; hence, pixel is marked as a saturated pixel if the intensity is above the threshold value. After calculating the mask, the inpaint method is invoked to restore degraded image pixels. Segmentation method proposed in this section is based on the reflectance computed for each pixel from its RGB value. Pixel-wise operation is a well-used approach; however, it increases computation time. Moreover, neighbor pixels may share almost identical characteristics. Superpixel is an obvious choice which represents a group of pixels that has similar quantitative features such as color and perception [26]. Superpixels are computed using simple linear iterative clustering (SLIC) algorithm [26]. According to this method, image color is transformed into CIELAB color space that preserves perception quality. Along with CIELAB values, pixels location in terms of coordinates are used to define the image in a 5-dimensional space. Local clusters of an image are achieved using the K-means algorithm. Figure 5 shows an example of SLIC clusters of an endoscopic image.

As described in the earlier section, when reflected light directly reaches to a camera sensor without any other surface interaction, then it preserves the optical property of that surface. If the sensor receives contributions from other rays reflected from different surfaces, then its optical property differs from the actual one. Moreover, the structural arrangement of an interacting surface can cause light scattering, and therefore, it splits into different directions.

Depends on the camera position and the structure of the surgical site, projected light can experience both specular and diffuse reflections specifically on the object edges. Therefore, computed reflectance of each pixel is not directly evaluated to its ideal one.

Figure 3 represents the reconstructed reflectances of cartilage and ACL in almost ideal specular reflection. The overall similarity between ACL and cartilage is approximately 89%. Similarity computed using Pearson correlation coefficient. Figure 6 represents reflectances of bone at various camera positions, and for each reflectance, the corresponding correlation coefficients are calculated with respect to ACL and cartilage reflectance under specular reflection. Unstructured soft tissue for instance, fat, dense collagen, and synovium appears randomly in surgical scenes. Moreover, the rounded (both concave and convex) shape of bone produces diffuse reflectance. As a result, the bone (cartilage) reflectance differs from its standard one.

If the reflection is not specular, then reflectance deviates from its ideal one. To incorporate the affected reflectance, a machine learning approach named decision tree method is used. The key benefit of the decision tree is to identify a boundary condition of reflectance and subsequently it decides whether a deviated reflectance is originally generated from tissue type or cartilage. Decision tree is a non-parametric supervised learning method that supports tree structure, and it learns decision rules to identify decision boundary to classify data. It is important to identify decision boundary that allows maximum deviation without losing accuracy. Hence, splitting a dataset based on features is an important step. Decision tree follows the greedy algorithm thus it examines every feature from the dataset over all data to find best splitting criteria for root node (top node of the tree). Hence, it receives the best

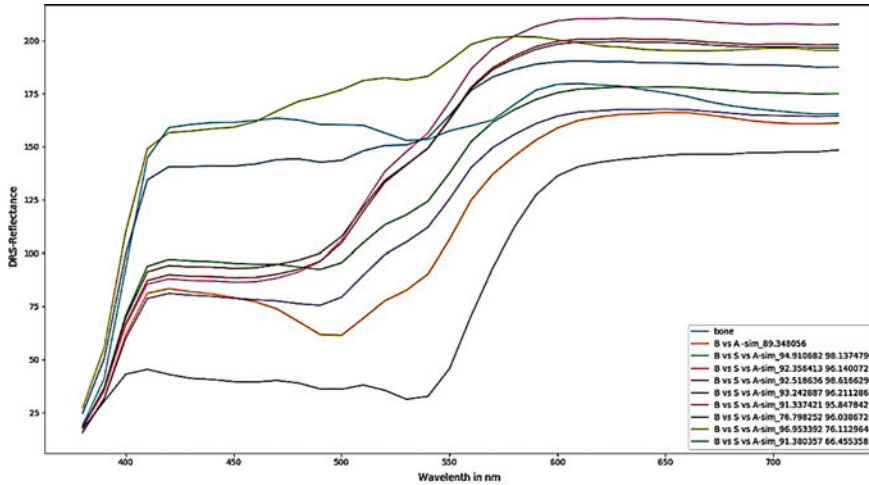


**Fig. 6** Figure represents the example spectra of different tissue types in the arthroscopic scenes. In ideal situation (first surface reflection), Pearson correlation coefficient of the cartilage reflectance differs from soft tissue within a range of 87–90% approximately. However, through the trajectory of the camera and light source inside a very confined space like the knee joint, the surgical scene meets various complex situations where the first surface reflection merely occurred. Often the reflected ray contains contributions from other rays. Thus, the resultant reflectance deviated from the ideal one. Figure’s legend represents Pearson correlation coefficient. Reflectance of soft-tissue types (S) is compared with bone (B) and ACL (A) in ideal situations. Last two rows of the legend represent that soft-tissue type (meniscus) is deviated toward bone tissue (96% and 91% similarity)

classification criterion for root node. In the successive phases, the other best feature is used to split the dataset further, and it follows until it finishes all the features.

## 5 Result and Discussion Conclusion

Figure 7 shows the visual and quantitative result obtained from our methodology. In order to measure segmentation accuracy, IoU is used. Images are taken under two different light sources. First two rows represent the images that are taken under the lighting condition provided by white micro-led and for other two images are taken under halogen light source. Test method confirms that when camera view along with light sources is directional toward the scene contents the proposed method receives high accuracy (row 1 and row 4) because of the reflected light directly reaches to the image sensor that preserves surface property. With angular view, method still receives high accuracy until the deviations of the received reflectances are small and distinguishable by the classifier. It also confirms that in shadow area or inside the cavity between two surfaces, reflectance drops most of its identical optical property which limits this approach.

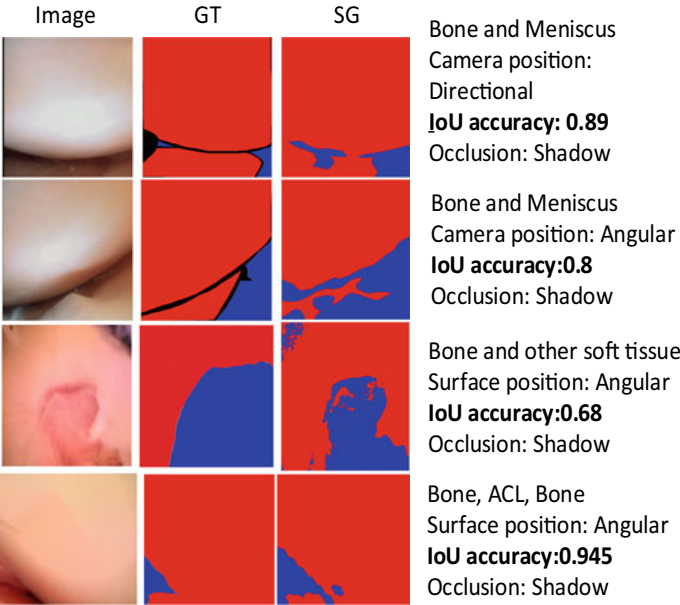


**Fig. 7** Represent the test result obtained from our proposed methodology. Column one represents input images; column two represents ground truth; column three represents segmented images obtained from our methodology. Last column represents quantitative information of each test case. Overall, the binary segmentation received high accuracy when specular reflections are observed. The method can address a small amount of light diffraction

This work realizes the feasibility and fitness of surface reflectance to address the segmentation process. However, it has some limitations as it is described in Fig. 7. In order to illustrate its limitations, a representative example of a ray tracing model for the knee joint is provided in Fig. 8a. At position P, the light and camera are projecting toward ACL. However, light scatters at the edge of ACL tissue (point 1). Some light part hits the second surface that is the edge of meniscus tissue (point 2). The light then again reflected toward the bone surface (point 3) and finally reflected back to the image sensor. Hence, the contributions of all the surfaces change the optical property of the ACL.

Similarly, Fig. 8b represents an example of a surgical scene. In the right image (Fig. 8b), light gets scattered at point 2 that is the edge of the rounded-convex curved bone (femur). This structural property also creates a shadow region and a small local cavity. Due to light rays bounced back between these two surfaces, the reconstructed spectra contain very small bone similarity (<70%). Point 3 describes almost similar phenomena. Additionally, this area is in close contact with bone (rounded shaped) and soft tissue (meniscus), so their contributions to each other are very significant. It is presented by the white area on the confidence map (left image) in Fig. 8b. Reflectance comes from the blue area has a strong contribution toward soft tissue which is expected, and the green area has little deviation from ideal soft-tissue

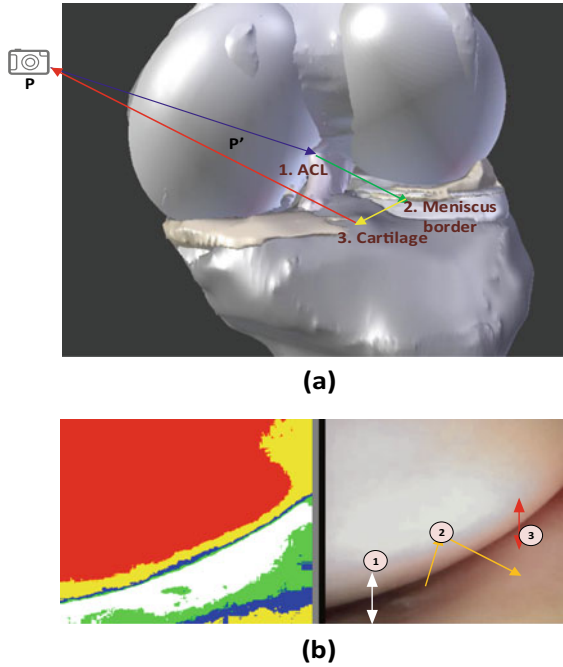


**Fig. 8** Figure **a** represents the example of a ray tracing model inside the knee cavity. Position of the camera and light source, geometrical and the surface roughness causes reflection, refraction, and scattering that perturb optical property from ideal one which may limit the proposed method. Figure **b** represents a similar example drawn from an endoscopic image. Left image shows a confidence map with different colors. Area marked with red color is strongly correlated to bone. The light reflected back from this area does not interact with other surfaces. However, at the edges of the curved bone, the reflectance spectra lose some property. Small deviation from bone spectra represented by yellow color. Similarly, blue and green color represents large and small deviation from ideal reflectance of soft tissue (ACL and meniscus). Shadow area marked with white color receives a larger deviation from bone or soft tissue

reflectance. Similarly, the yellow area has little deviation from bone reflectance under specular reflection. Red area represents bone with more than 97% similarity to its specular reflectance.

Concisely, this research evidences that surface reflectance can solve the surgical scene segmentation problem, and it does not require to learn all pixels with possible color variations. The limitations are observed only when a deviated reflectance is very close to reflectance of another surface. Shadow regions, inside the gap between the curved surfaces and intense specular reflection produced by high intensity of incident light and smooth surface, are fundamental grounds for such deviations. However, segmentation is performed on static image from the dynamic MIS video sequences. Dynamic nature of MIS provides an opportunity of observing scene surface from multiple view angle. Surface responses at different view angle will be different which can be an alternative approach to overcome its limitation.



## 6 Conclusion

MIS procedure is performed through the indirect vision of surgical space that reinforces the demand of augmented vision to improve its accuracy. However, surgical images of MIS are texture less, and subsequently, it is showing very limited image features. These are restricting the applicability of vision-based solutions such as surgical scene segmentation that has an obvious demand to improve quality of life. In this work, surface reflectance is evaluated as an alternate to pixel information such as texture or feature to perform scene segmentation. This research confirms that, surface reflectance can be great alternative to attain surgical scene segmentation.

In this work, we are proposing arthroscopic scene segmentation based on multi-spectral response reconstructed from RGB triplet; thus, it does not require additional imaging system or optical tool. Moreover, this work confirms that the segmentation accuracy of the proposed method solely depends on the spatial features of the anatomical structure at wavelength from 380 to 730 nm irrespective to texture and features details.

**Acknowledgements** This work is supported by Australian Indian Strategic Research Fund (AISRF), The Medical Engineering Research Facility and QUT Center for Robotics.

## References

1. Jonmohamadi Y, Ali S, Liu F, Roberts J, Crawford R, Carneiro G, Pandey AK (2021) 3D semantic mapping from arthroscopy using out-of-distribution pose and depth and in-distribution segmentation training. In: International conference on medical image computing and computer-assisted intervention (MICCAI). Springer, Cham, pp 383–393
2. Ali S, Jonmohamadi, Takeda Y, Roberts J, Crawford R, Brown C, Pandey AK (2021) Arthroscopic multi-spectral scene segmentation using deep learning. arXiv preprint [arXiv:2103.02465](https://arxiv.org/abs/2103.02465)
3. Shahnewaz A, Pandey AK (2020) Color and depth sensing sensor technologies for robotics and machine vision. In: Machine vision and navigation. Springer, Cham, pp 59–86
4. Jansen-van Vuuren RD, Shahnewaz A, Pandey AK (2020) Image and signal sensors for computing and machine vision: developments to meet future needs. In: Machine vision and navigation. Springer, Cham, pp 3–32
5. Madad Zadeh S, Francois T, Calvet L, Chauvet P, Canis M, Bartoli A, Bourdel N (2020) SurgAI: deep learning for computerized laparoscopic image understanding in gynaecology. *Surg Endosc* 34(12):1–7
6. Khan ZA, Beghdadi A, Kaaniche M, Cheikh FA (2020) Residual networks based distortion classification and ranking for laparoscopic image quality assessment. In: 2020 IEEE international conference on image processing (ICIP). IEEE, pp 176–180
7. Maqbool S, Riaz A, Sajid H, Hasan O (2020) m2caiSeg: semantic segmentation of laparoscopic images using convolutional neural networks. arXiv preprint [arXiv:2008.10134](https://arxiv.org/abs/2008.10134)
8. Ali S, Crawford R, Maire, Pandey AK (2021) Towards robotic knee arthroscopy: multi-scale network for tissue-tool segmentation. arXiv preprint [arXiv:2110.02657](https://arxiv.org/abs/2110.02657)
9. Ali S, Pandey AK (2022) ArthroNet: monocular depth estimation technique toward 3D segmented maps for knee arthroscopic. *Intell Med*. <https://doi.org/10.1016/j.imed.2022.05.001>



10. Ali S, Jonmohamadi Y, Takeda Y, Roberts J, Crawford R, Pandey AK (2020) Supervised scene illumination control in stereo arthroscopes for robot assisted minimally invasive surgery. *IEEE Sens J* 21(10):11577–11587
11. Ali S, Jonmohamadi Y, Crawford R, Fontanarosa D, Pandey AK (2021) Surgery scene restoration for robot assisted minimally invasive surgery. *arXiv preprint [arXiv:2109.02253](https://arxiv.org/abs/2109.02253)*
12. de Boer LL, Molenkamp BG, Bydlon TM et al (2015) Fat/water ratios measured with diffuse reflectance spectroscopy to detect breast tumor boundaries. *Breast Cancer Res Treat* 152:509
13. Ali S, Pandey AK (2022) Towards robotic knee arthroscopy: spatial and spectral learning model for surgical scene segmentation. In: *Proceedings of international joint conference on advances in computational intelligence*. Springer, Singapore, pp 269–281
14. Stigell P, Miyata K, Hauta-Kasari M (2007) Wiener estimation method in estimating of spectral reflectance from RGB images. *Pattern Recogn Image Anal* 17(2):233–242
15. Nguyen RMH, Prasad DK, Brown MS (2014) Training-based spectral reconstruction from a single RGB image. In: *Computer vision. ECCV 2014. Lecture notes in computer science*, vol 8695. Springer, Cham, pp 186–201
16. Xiao K, Zhu Y, Li C, Connah D, Yates JM, Wuerger S (2016) Improved method for skin reflectance reconstruction from camera images. *Opt Express* 24:14934–14950
17. Liang J, Wan X (2017) Optimized method for spectral reflectance reconstruction from camera responses. *Opt Express* 25:28273–28287
18. Zhang X, Huang W, Wang Qi, Li X (2020) SSR-NET: spatial–spectral reconstruction network for hyperspectral and multispectral image fusion. *IEEE Trans Geosci Remote Sens* 59(7):5953–5965
19. Otsu H, Yamamoto M, Hachisuka T (2018) Reproducing spectral reflectances from tristimulus colours. *Comput Graphics Forum* 37(6):370–381
20. Chern JR, Wang CM (2005) A novel progressive refinement algorithm for full spectral rendering. *Real-Time Imaging* 11(2):117–127
21. Afifi M, Price B, Cohen S, Brown MS (2019) When color constancy goes wrong: correcting improperly white-balanced images. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 1535–1544
22. Singh D, Kumar V (2018) Single image haze removal using integrated dark and bright channel prior. *Mod Phys Lett B* 32(04):1850051
23. Peng Y-T, Cosman PC (2017) Underwater image restoration based on image blurriness and light absorption. *IEEE Trans Image Process* 26(4):1579–1594
24. Telea A (2004) An image inpainting technique based on the fast marching method. *J Graphics Tools* 9(1):25–36
25. Zeng Y, Fu J, Chao H, Guo B (2019) Learning pyramid-context encoder network for high-quality image inpainting. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 1486–1494
26. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 34(11):2274–2282

## Chapter 18

# Classification and Segmentation on Multi-regional Brain Tumors Using Volumetric Images of MRI with Customized 3D U-Net Framework



Md. Faysal Ahamed, Md. Robiul Islam, Tahmim Hossain, Khalid Syfullah, and Ovi Sarkar

**Abstract** Automated segmentation is a computerized technique that helps to find tumor location, size, and shape. Human segmentation is error prone, time consuming, and needs an expert radiologist. In our study, we developed a customized 3D U-Net model that processes 3D volumetric images for multiclass tumor segmentation. This framework is modified in such a way that the gradient flow is better for finding accurate output. The BraTS 2020 dataset is used to train this network with end-to-end learning strategy followed by defining the proper skip connection from encoder to decoder. In model evaluation, binary cross-entropy with Dice loss functions is utilized. Testing samples are predicted and classified into three regions: whole tumor (WT), tumor core (TC), and enhancing tumor (ET). Model performance is evaluated through Dice and Jaccard coefficient metrics for each class. Furthermore, in the testing dataset, the Dice scores are found at 0.753, 0.799, and 0.887 for the ET, TC, and WT classes.

**Keywords** Deep learning · Medical imaging · MRI · Semantic segmentation · 3D U-Net · BraTS 2020

## 1 Introduction

A brain tumor is described as the proliferation of abnormal cells in the human brain. These abnormally increased cells create swelling and pressure in the brain. Malignant type increases the cancer possibility [1]. In recent years, around 3 million people have been diagnosed with various forms of malignancies in central nervous system (CNS). Many people have died because of not receiving treatment, and it can be lethal for the affected person [2]. Different types of glioma are low-graded glioma (LGG), glioblastoma multiform (GBM), or high-graded glioma (HGG) [3]. Segmentation technique

---

Md. Faysal Ahamed (✉) · Md. Robiul Islam · T. Hossain · K. Syfullah · O. Sarkar  
Department of Electrical and Computer Engineering, Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh  
e-mail: [faysalahamedjishan@gmail.com](mailto:faysalahamedjishan@gmail.com)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_18](https://doi.org/10.1007/978-981-19-7528-8_18)

223

analyzes medical imaging to extract clinical information regarding tumor presence, location, and size. Manual segmentation by a radiologist is time-consuming work and needs to be done with expertise for accurate annotation. Automated segmentation is more efficient in terms of speeding up diagnostic, interventional radiology, and surveillance. There are various technologies for capturing brain images, including positron emission tomography (PET) using nuclear technology, computed tomography (CT), single-photon emission tomography (SPECT), magnetic resonance spectroscopy (MRS), and magnetic resonance imaging (MRI) [4]. MRI system extracts accurate information for detection, classification, and segmentation [5]. BraTS provides 3D volumetric HGG and LGG instances with accurate ground truth labeling information. A customized 3D U-Net model was built inside this study to segment multiclass designated necrotic/non-enhancing tumor (NCR/NET) as “whole tumor (WT)”, peritumoral edema (ED) as “tumor core (TC)”, and “enhancing tumor (ET)”. This paper is structured as an analysis of past work in Sect. 2, Sect. 3 is methodology, Sect. 4 is dataset, Sects. 5 and 6 are model with settings, and Sects. 7 and 8 cover the analysis of the performance and prediction of tumor classes with a comparative study. We have contributed the following significant contributions to our work:

- For better detection and classification, a customized 3D U-Net framework was developed.
- Higher performance was achieved for volumetric brain imaging.
- To segment tumors, we utilized three different classes: enhancing tumor, tumor core, and whole tumor regions.
- 3D volume images were processed and extracted the image features from each layer, then connected end to end with the decoder part.
- Calculated the Jaccard index to validate the Dice scores to determine class evaluation in both quantitative and qualitative measuring approaches.

## 2 Related Works

There are numerous machine learning algorithms used in segmentation to find tumor regions. Soltaninejad et al. [6] worked on brain tumor classification. They used the SVM classifier methodology to classify brain tumor grades. BraTS 2012 dataset was used, and FLAIR images were trained using bimodal MRI data. Dong et al. [7] segmented the tumors following 2D U-Net network from BraTS 2015 dataset. Myronenko [8] implemented autoencoder regularization algorithms to detect tumor regions on BraTS 2018. Valverde et al. [9] coupled U-Net and the VGG16 model on that MRI dataset. In this work, they also emphasized pre-trained CNN and completed a survey on the past works. To recognize model evaluation, a CCE loss function was applied. To improve detection outcomes, Kamnitsas et al. [10] utilized an ensemble task on various frameworks. Using EMMA, researchers explored a wide range of training approaches on various frameworks to generalize outcomes from BraTS 2018. Using a hyperdense inception 3D U-Net model, Qamar et al. [11] segmented

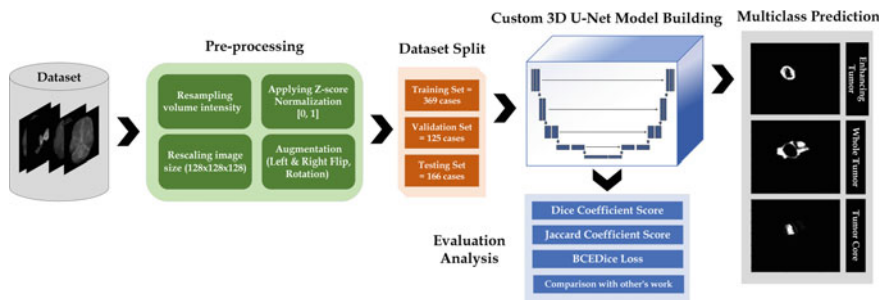


**Fig. 1** Overall working flow representing our proposed system

tumors into three classes: total malignancy, tumor core, and enhanced tumor. They applied multiclass segmentation on BraTS 2020 dataset. Ali et al. [12] worked on multiclass segmentation with 3D CNN and U-Net with a transfer learning approach. They worked on the BraTS 2019 dataset.

### 3 Methodology

Firstly, the dataset is separated into three sections: training, validation, and testing sets, with four categories of data for each patient sample. A few preprocessing tasks are performed, and a custom model is developed focusing on the 3D U-Net framework. After building, the hyperparameters are tuned to improve the model's performance. The multiclass segmentation is predicted into three categories. To create the assessment metrics, Dice and Jaccard indexes are computed. Finally, some patient samples are taken for prediction. The overall working flow in our paper is presented in Fig. 1.

## 4 Materials

### 4.1 Dataset

Brain Tumor Segmentation Challenge (BraTS) 2020 dataset includes a training and validation set for model training and a testing set for prediction purposes. It contains three-dimensional multimodal images from the magnetic resonance imaging (MRI) technique. The training set has 369 samples, whereas the validation set includes 125 samples with appropriate ground truth labeling and incorporates 293 and 76 HGG and LGG instances, respectively. The input image size is  $240 \times 240 \times 190$  with four channels [5, 13–15]. For the testing set, 166 cases are available. The

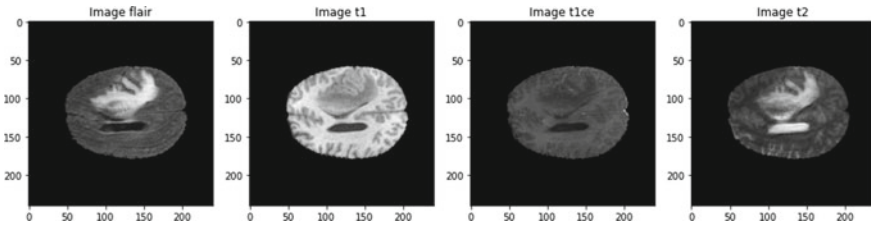


**Fig. 2** BraTS 2020 dataset contains FLAIR, t1, t1ce, t2 categories images and the volume of each one is T1 (two layers) 1–6 mm, T1ce 1 mm, T2 weighted 2–6 mm, FLAIR 1–6 mm

volumetric images are a combination of four categories of data, which are presented in Fig. 2. The data is annotated by multiple neuro-radiologists, and those annotations are denoted as “ground truth”. These annotations are divided into three categories: necrotic/non-enhancing tumor, peritumoral edema, and GD-enhancing tumor.

## 4.2 Preprocessing

BraTS 2020 contains skull-stripped ( $1 \text{ mm}^3$ ) MR volume data. A mean calculation is applied to resample the images, which increases the volume intensity. Z-score normalization (0, 1) is also used. The image size is rescaled to  $128 \times 128 \times 128$ . Phase augmentation is performed to increase the input data. Data is processed as rotation and scaling at each iteration, and the volume of images is flipped randomly to the left or right.

## 5 Custom 3D U-Net Model

We developed a custom 3D U-Net framework, and the working strategy of this model is comparable to that of the classic 2D U-Net model [16]. In Fig. 3, the complete working model diagram is presented. In this architecture, each block is colored to identify easily. Model input is volumetric images with four types of data. The significance of this work is traditional U-Net model which takes 2D input, but this customized model takes volumetric input for processing. After the training process at the output section, a multiclass result is generated where the image resolution is same as the input. At first, Çiçek et al. [17] developed this model for volumetric segmentation work, and we use that model for brain tumor multiclass segmentation by modifying some layers. Our designed model is explained below in detail.

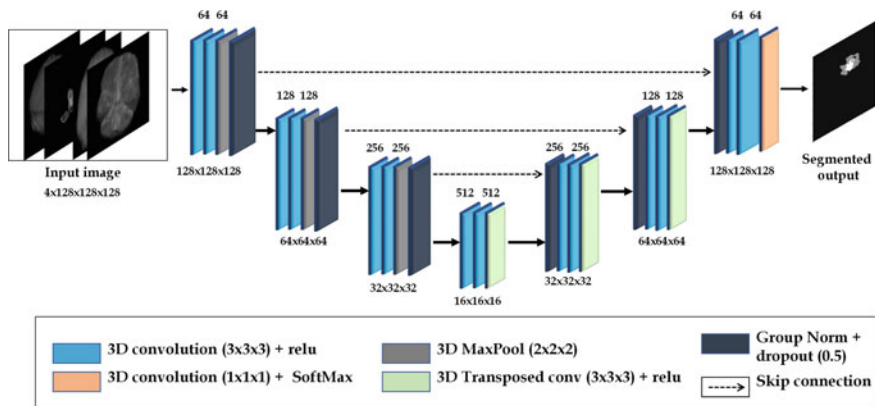


**Fig. 3** Schematic model arrangement of custom 3D U-Net framework

## 5.1 Encoder Part

The model's working strategy is divided into two modes, known as encoder and decoder parts, as shown in Fig. 3. Four resolution steps are taken for both the encoder and decoder paths. The first two consecutive 3D convolution blocks are taken to the size of  $3 \times 3 \times 3$  with  $m$  filters in the encoding section.  $m$  defines the initial number of filters. This 3D convolution is also known as a separable 3D block layer, and it adjusts the filtering at  $k * 2^n$  for each encoder path (resolution steps are defined as  $n$ ). After two consecutive conv3D layers, we use a  $2 \times 2 \times 2$  maxpool. Maxpool provides better gradient flow for each successive section. If we use average pooling, gradient performance degrades. Then we use group normalization techniques, ignoring batch normalization because of memory reliability. Overfitting cases are solved by setting the dropout frequency value to 0.5.

Our customized model adjusts the maximum variation and feature channel at each layer from the usual U-Net design, which speeds up 3D volume processing. The input image size in the encoder section is  $4 \times 128 \times 128 \times 128$ , with four channels. The encoder and decoder have three steps in total. Figure 3 depicts the specifics of each phase of convolution using ReLu and max pooling procedures, where the feature map depth is reduced as the feature channel is increased (doubled). In the transition phases, convolution  $3 \times 3 \times 3$  is utilized having stride 2, accompanied by group normalization (GN) with ReLu function. The feature map's resolution is reduced to  $16 \times 16 \times 16$  at the final stage, and the feature channel is 512.

## 5.2 Decoder Part

This part is similar to the encoder part. At the encoding step, the entire process is downsampled, and here a reverse path means the upsampling method is used. In this

case,  $3 \times 3 \times 3$  size 3D transposed convolution with  $k * 2^n$  filtering layer is used. The encoder part concatenates each layered transposed convolution output into the same steps. The relevance of employing transposed convolution is to standardize upsampling tasks by using an optimal interpolation strategy. Another approach is a pre-defined interpolation, but it is overexposed if data redundancy is high. Group normalization with the relu function is utilized to compensate for nonlinearity.

From Fig. 3, the decoder recovers the image resolution by the upsampling process via transposed convolution, where the feature map is enlarged and the feature channel is decreased. Likewise, group normalization and the ReLu function are used after transposed convolution  $3 \times 3 \times 3$  having stride 2. Skip connection is employed to reduce the gradient loss, and feature maps are combined with the encoder and decoder. The final segmentation is done using convolution  $1 \times 1 \times 1$  with the softmax function to define individual tumor classes. In the final layer, segmented all classes are found.

### 5.3 Loss Function

BCE-Dice loss was the error term we employed in our studies. The significance is that this combination form stabilizes the binary cross-entropy (BCE) loss. The formula of this function [18] is as follows:

$$\text{BCE Loss} = -(\beta * p \cdot \log(\hat{p}) + (1 - p) \cdot \log(1 - \hat{p})) \quad (1)$$

where  $\beta$  defines the false negative or false positive. If the resultant value carries more false negative values than true, that value is set to  $\beta > 1$ , or set to  $\beta < 1$  to reduce the false positives. Now, Dice loss is widely used, and this finds the similarity result between two points. The formula is as follows [18]:

$$\text{DSC Loss} = \frac{2|K \cap L|}{|K| + |L|} \quad (2)$$

where the upper portion of Eq. 2 represents the common area of  $K$  and  $L$ . The number of objects in group  $K$  is denoted by  $|K|$ , while the number of attributes in set  $L$  is denoted by  $|L|$ .

**Table 1** Hyperparameter values for model training time

| Parameter name               | Defined value/type                   |
|------------------------------|--------------------------------------|
| Input image size (Processed) | $4 \times 128 \times 128 \times 128$ |
| Training batch size          | 1                                    |
| Learning rate (lr)           | $5e-4$                               |
| Optimizer                    | Adam                                 |
| Epochs                       | 50                                   |
| Loss function                | BCE-Dice loss                        |
| Normalization                | Group normalization                  |

## 6 Experimental Analysis

### 6.1 Experimental Settings

This model is built using PyTorch, which was developed by Facebook AI. Pytorch works on the torch library, which helps to improve the computer vision application. Table 1 summarizes the hyperparameters that are utilized to train the proposed model. *Hardware:* Our working environment was Intel Core i5 7200u, GTX 1050Ti GeForce GPU, and RAM support of 32 GB.

### 6.2 Evaluation Metrics

Multiple criteria are used to determine the performance requirements and segmentation outcomes. The most popular way is to use the Dice coefficient metrics. The anticipated tumor area and the ground truth tumor area are represented by  $V$  and  $W$  in this formula, respectively. These two measurements overlap by a factor of 0 to 1. A value close to one indicates higher output [19].

$$\text{Dice}(V, W) = 2 \frac{|V \cap W|}{|V| + |W|} \quad (3)$$

Jaccard coefficient indicates similarity, and Jaccard distance determines dissimilarity between two binary (and non-binary) variables [19]. By eliminating the  $P$  from the straightforward matching coefficient, Jaccard coefficient obtains the corresponding pattern:

$$P_{ij} = \frac{L}{L + M + N} \quad (4)$$

$$J_{ij} = 1 - P_{ij} = 1 - \frac{L}{L + M + N} = \frac{M + N}{L + M + N} \quad (5)$$



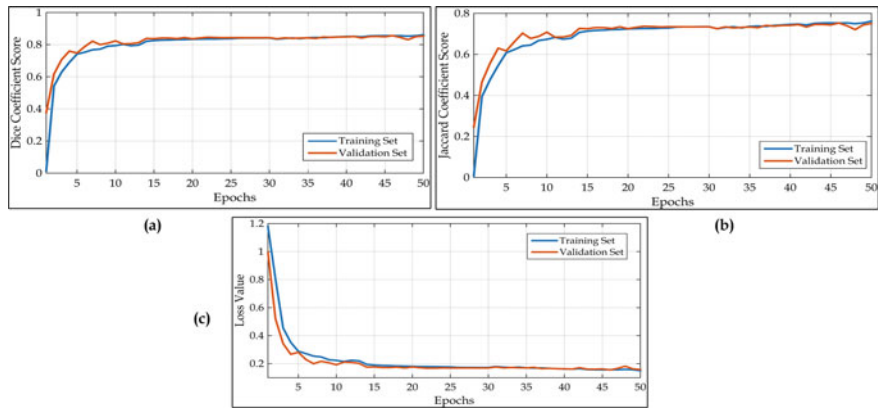


**Fig. 4** Model training time performance for 50 epochs, **a** Dice coefficient score, **b** Jaccard coefficient score, and **c** binary cross-entropy with Dice loss

**Table 2** Performance of a multiclass model evaluation for brain tumor segmentation is presented here

| Datasets           | Criteria | Dice coefficients |            |             | Jaccard coefficients |            |             |
|--------------------|----------|-------------------|------------|-------------|----------------------|------------|-------------|
|                    |          | Enhancing tumor   | Tumor core | Whole tumor | Enhancing tumor      | Tumor core | Whole tumor |
| Training dataset   | Mean     | 0.793             | 0.841      | 0.926       | 0.725                | 0.736      | 0.848       |
|                    | Median   | 0.832             | 0.879      | 0.948       | 0.743                | 0.758      | 0.874       |
|                    | Std dev  | 0.102             | 0.127      | 0.052       | 0.165                | 0.169      | 0.125       |
| Validation dataset | Mean     | 0.778             | 0.847      | 0.912       | 0.705                | 0.713      | 0.823       |
|                    | Median   | 0.801             | 0.857      | 0.931       | 0.745                | 0.750      | 0.865       |
|                    | Std dev  | 0.120             | 0.141      | 0.040       | 0.173                | 0.177      | 0.144       |
| Testing dataset    | Mean     | 0.753             | 0.799      | 0.887       | 0.637                | 0.677      | 0.830       |
|                    | Median   | 0.815             | 0.845      | 0.902       | 0.689                | 0.723      | 0.814       |
|                    | Std dev  | 0.201             | 0.152      | 0.071       | 0.210                | 0.173      | 0.101       |

where  $L$  = both objects have a positive number of variables,  $M = L$  measure of positive variables for every  $i$ th object and a measure of negative variables for every  $j$ th object,  $N = L$  measure of negative variables for every  $i$ th object and a measure of positive variables for every  $j$ th object,  $P$  = negative values for both objects variable, and  $J$  = Jaccard score.

## 7 Result Analysis

### 7.1 Quantitative Results

Our model performs well in multiclass segmentation. Figure 4 contains the curves of training time model loss, Dice coefficient score, and Jaccard coefficient score with the average of three classes. In total, 50 epochs are used for model training and show consistent performance on Dice and Jaccard scores. The BCE-Dice loss was found to be 0.156 for the training set and 0.158 for the validation set. In addition, we evaluated three-class performances for three types of datasets as well. “Enhancing tumor (ET)”, “tumor core (TC)”, and “whole tumor (WT)” are classified, and different measurement techniques are followed to explain the performance called the mean, median, and standard deviation (SD). The standard deviation (sigma) is the deviation score between the mean value and middle-order data, where less deviation indicates higher model performance. From the testing dataset, we obtained Dice scores of 0.753, 0.799, and 0.887 for the ET, TC, and WT classes, respectively. Table 2 presents all of the scores for the split datasets, and both evaluation indexes are shown for multiclass segmentation. Based on this research, we can see that our performance is better at the whole tumor class segmentation and also at the tumor core class segmentation. However, the results for the enhancing class are lower. Enhancing tumor class results is affected by the t1ce layer, and this layer’s intensity value is not as high compared to others. This class result is generated based on the other

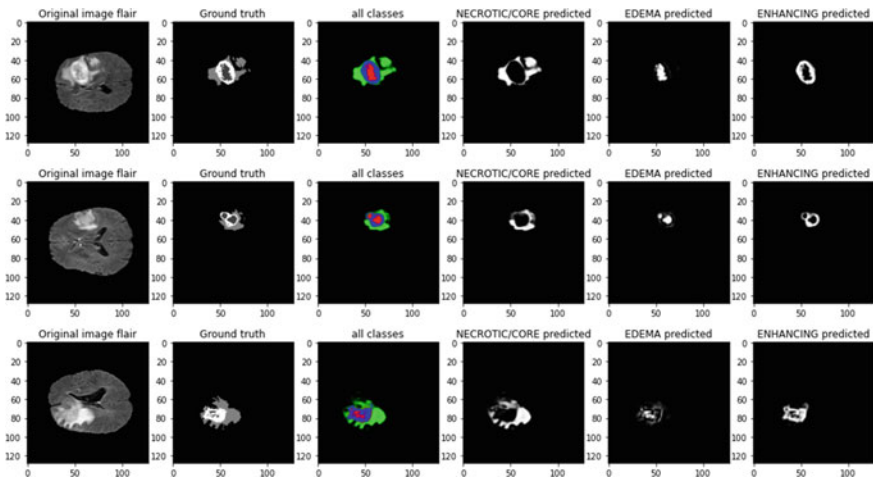


**Fig. 5** This image denotes three rows (input samples): first row contains the HGG input image (middle sliced of volume), second row contains the LGG input image, and third row contains the testing sample randomly. Three classes are predicted where green defines whole tumor “WT” presented as necrotic/core, pink defines tumor core “TC” presented as edema, and blue defines enhancing tumor “ET”

three categories of data (FLAIR, t1, and t2), which is why this class performance decreased.

7.2 Qualitative Results

Figure 5 shows some projections based on the BraTS 2020 testing dataset. Prediction is conducted after successfully training a customized 3D U-Net model. It is now possible to compare how accurately our model can segment from the ground truth label. An expert neuroradiologist has marked this ground truth result. BraTS delivered this ground truth together with a dataset of annotations classified as necrotic/core, edema, and enhancing classes. For easier understanding, we labeled those three classifications as whole tumor (WT), tumor core (TC), and enhancing tumor (ET). For prediction and in-class recognition, HGG and LGG patient samples are taken. The pixel intensity at HGG is higher, and classes are predicted more accurately. Flair images are used because it has adjustable intensity compared to the other t1, t1ce, and t2 categories. Furthermore, we performed a class separation, which is why each class is more accurately comprehensible.

**Table 3** Mean (average) value is derived from WT, TC, and ET classes

| Model name                         | Dice coefficients |              |              |              | Jaccard coefficients |       |       |       |
|------------------------------------|-------------------|--------------|--------------|--------------|----------------------|-------|-------|-------|
|                                    | WT                | TC           | ET           | Mean         | WT                   | TC    | ET    | Mean  |
| nnU-Net ensemble with SoftMax [20] | 0.910             | 0.844        | 0.776        | 0.843        | –                    | –     | –     | –     |
| nnU-Net + DRO [21]                 | 0.906             | 0.841        | 0.753        | 0.833        | –                    | –     | –     | –     |
| Single SA-Net model [22]           | 0.904             | 0.842        | 0.785        | 0.843        | –                    | –     | –     | –     |
| Context 3D U-Net [23]              | 0.891             | 0.842        | 0.756        | 0.829        | –                    | –     | –     | –     |
| 3D GAN [24]                        | 0.893             | 0.792        | 0.750        | 0.812        | –                    | –     | –     | –     |
| General 3D U-Net [25]              | 0.886             | 0.843        | <b>0.786</b> | 0.838        | –                    | –     | –     | –     |
| Customize 3D U-Net (Proposed)      | <b>0.912</b>      | <b>0.847</b> | 0.778        | <b>0.848</b> | 0.823                | 0.713 | 0.705 | 0.747 |

This comparison is based on validation dataset from the BraTS 2020. The entire result is presented here, along with a comparison to the performance of other existing models

## 8 Result Comparison and Discussion

The most essential terminology is whether our model has successfully segmented the tumor regions or not. This is contrasted with several comparable works. However, the outcomes may differ depending on the dataset's annotation. Table 3 presented some comparisons where the BraTS 2020 dataset is used on other models. Based on this comparison, it is straightforward to infer that our suggested framework significantly outperformed others. Comparative class segmentation score is higher for WT and TC classes, and ET shows a little lower score than the general 3D U-Net model. The average score also denoted single class performance. Other researchers just evaluated their model based on Dice score results; however, we developed multiclass Jaccard performance. Table 2 shows that our model maintained consistent performance in both Dice and Jaccard performance. In a real-life application, this deep learning method can have a significant effect if further improvement is made.

## 9 Conclusion

Throughout this paper, we developed a customized 3D U-Net framework for volumetric image segmentation and classification by configuring encoder and decoder convolution layers. BraTS 2020 is a challenge powered by Intel AI, intending to improve segmentation. Our goal is to use volumetric dataset to get an enhanced performance on prediction and then compare it to the same type of existing study. Finally, our solution is superior to others, yet certain limitations remain. We cannot employ any more post-processing in the final output and concentrate on intensity resolution and run-time model biasing correction. In the future, we will continue to develop this technique, which will integrate the model multiclass with other frameworks to increase classification efficiency. This research can help segment tumors and treatment planning processes for patients clinically.

## References

1. Bauer S, Wiest R, Nolte LP, Reyes M (2013) A survey of MRI-based medical image analysis for brain tumor studies. *Phys Med Biol* 58(13):R97
2. Brain tumor—statistics. Accessed on 03/04/2022, April 2022
3. Villa C, Miquel C, Mosses D, Bernier M, Di Stefano AL (2018) The 2016 world health organization classification of tumours of the central nervous system. *La Presse Médicale* 47(11–12):e187–e200
4. Wadhwa A, Bhardwaj A, Verma VS (2019) A review on brain tumor segmentation of MRI images. *Magn Reson Imaging* 61:247–259
5. Bakas S, Reyes M, Jakab A, Bauer S, Rempfler M, Crimi A, Shinohara RT, Berger C, Ha SM, Rozycki M et al (2018) Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint [arXiv:1811.02629](https://arxiv.org/abs/1811.02629)*

6. Soltaninejad M, Ye X, Yang G, Allinson N, Lambrou T et al (2014) Brain tumour grading in different MRI protocols using SVM on statistical features
7. Dong H, Yang G, Liu F, Mo Y, Guo Y (2017) Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks. In: Annual conference on medical image understanding and analysis. Springer, Berlin, pp 506–517
8. Myronenko A (2018) 3D MRI brain tumor segmentation using autoencoder regularization. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 311–320
9. Valverde JM, Imani V, Abdollahzadeh A, De Feo R, Prakash M, Ciszek R, Tohka J (2021) Transfer learning in magnetic resonance brain imaging: a systematic review. *J Imaging* 7(4):66
10. Kamnitsas K, Bai W, Ferrante E, McDonagh S, Sinclair M, Pawlowski N, Rajchl M, Lee M, Kainz B, Rueckert D et al (2017) Ensembles of multiple models and architectures for robust brain tumour segmentation. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 450–462
11. Qamar S, Ahmad P, Shen L (2020) Hi-net: hyperdense inception 3D UNet for brain tumor segmentation. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 50–57
12. Ali M, Gilani SO, Waris A, Zafar K, Jamil M (2020) Brain tumour image segmentation using deep networks. *IEEE Access* 8:153589–153598
13. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS, Freymann JB, Farahani K, Davatzikos C (2017) Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Sci Data* 4(1):1–13
14. Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R et al (2014) The multimodal brain tumor image segmentation benchmark (brats). *IEEE Trans Med Imaging* 34(10):1993–2024
15. Simpson AL, Antonelli M, Bakas S, Bilello M, Farahani K, Van Ginneken B, Kopp-Schneider A, Landman BA, Litjens G, Menze B et al (2019) A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint [arXiv:1902.09063](https://arxiv.org/abs/1902.09063)*
16. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 234–241
17. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O (2016) 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 424–432
18. Jadon S (2020) A survey of loss functions for semantic segmentation. In: 2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB). IEEE, pp 1–7
19. Bertels J, Eelbode T, Berman M, Vandermeulen D, Maes F, Bisschops R, Blaschko MB (2019) Optimizing the dice score and jaccard index for medical image segmentation: theory and practice. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 92–100
20. Fidon L, Ourselin S, Vercauteren T (2020) Generalized Wasserstein Dice score, distributionally robust deep learning, and ranger for brain tumor segmentation: brats 2020 challenge. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 200–214
21. Fidon L, Ourselin S, Vercauteren T (2020) Distributionally robust deep learning using hardness weighted sampling. *arXiv preprint [arXiv:2001.02658](https://arxiv.org/abs/2001.02658)*
22. Yuan Y (2020) Automatic brain tumor segmentation with scale attention network. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 285–294
23. Ahmad P, Qamar S, Shen L, Saeed A (2020) Context aware 3D UNet for brain tumor segmentation. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 207–218
24. Cirillo MD, Abramian D, Eklund A (2020) Vox2vox: 3D-GAN for brain tumour segmentation. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 274–284
25. Henry T, Carré A, Lerousseau M, Estienne T, Robert C, Paragios N, Deutsch E (2020) Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-Net neural networks: a brats 2020 challenge solution. In: International MICCAI Brainlesion workshop. Springer, Berlin, pp 327–339

# Chapter 19

## Explainable AI and Ensemble Learning for Water Quality Prediction



Nakayiza Hellen, Hasibul Hasan Sabuj, and Md. Ashraful Alam

**Abstract** Accessing safe water is considered a vital element of effective policy for protecting health. Numerous innovative technologies are slowly replacing human labor and other state-of-the-art methods in water quality evaluation. One of the biggest challenges faced by policy-makers and other responsible Public Health Authorities is the lack of a relatively generalizable machine learning model for water quality prediction for human consumption with provision of explanations for understanding the most influential water quality parameters. Therefore, in this paper, we proposed and applied Shapley Additive Explanations (SHAP) to study and model a robust generalizable ensemble machine learning model for water quality prediction based on water potability and other water quality metrics from various water quality samples around the world. We also calibrated our final ensemble model to achieve an accuracy score of over 90% hence making it more generalizable for water quality prediction.

**Keywords** Water quality · Explainable artificial intelligence (XAI) · Ensemble learning · Public health

## 1 Introduction

The World Health Organization (WHO) reported that half of the world population is likely to lack safe water by 2025. Furthermore, United Nations (UN) in its 2018 report mentioned that without taking necessary action, challenges will only increase

---

N. Hellen (✉) · H. H. Sabuj · Md. Ashraful Alam  
BRAC University, Dhaka 1212, Bangladesh  
e-mail: [nakayiza.hellen@g.bracu.ac.bd](mailto:nakayiza.hellen@g.bracu.ac.bd)

H. H. Sabuj  
e-mail: [sabuj.hh@gmail.com](mailto:sabuj.hh@gmail.com)

Md. Ashraful Alam  
e-mail: [ashraful.alam@bracu.ac.bd](mailto:ashraful.alam@bracu.ac.bd)

by 2050 yet the global demand for fresh water is expected to be a third greater than it is now [1].

Water quality deterioration adversely impacts health, environment, and infrastructure or development at national, regional, and local levels. According to United Nations (UN), waterborne diseases cause death of more than 1.5 million people each year, which is way greater than a combination of deaths caused by accidents, crimes, and terrorism [2]. Access to safe water is essential to health and is considered a component of effective policy for health protection. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions. Therefore, it is very vital to devise novel methodologies for assessing water quality and to predict its future trends. It is necessary to obtain a global water quality dataset with various water quality metrics to perform water quality modeling. Several studies have been done on the use of artificial intelligence in predicting water quality. However, there is still lack of a relatively generalizable machine learning model for water quality prediction for public consumption with provision of explanations for understanding the most influential water quality parameters. And besides, many other studies used either few or too many parameters which is not efficient enough in predicting the water quality.

In this work, we demonstrate the water quality features in machine learning modeling using various exploratory data analytic techniques and deploy Shapley Additive Explanations (SHAP) to interpret the water quality predictions of the models. We then ensemble the best machine learning models and calibrate the final ensemble model that is robust and generalizable enough to precisely predict water quality for human consumption.

The remaining part of this paper has been organized as follows: Sect. 2 briefly reviews how other researchers used machine learning and artificial intelligence to predict water quality. Section 3 discusses the components of our proposed model, its design and implementation. Section 4 explores results obtained. Section 5 summarizes the whole paper together and suggests future potential derivative work for further research.

## 2 Literature Review

### 2.1 Water Quality

Water pollution makes water unfit for human, industrial, and agricultural uses [3]. Getting water quality to a level needed for public usage requires that water supplies must be managed properly [4]. Water quality is usually calculated using water quality parameters attained through time-consuming lab analysis [5]. Multivariate statistical techniques like principal component analysis (PCA) and geo-statistical techniques

like kriging, transitional probability, multivariate interpolation, and regression analysis have been used to discover the relationship among the various water quality parameters.

## ***2.2 Existing Works on Water Quality Prediction***

Researchers gathered water samples from different areas and performed manual lab analysis by testing the samples against different parameters [6]. Feedforward neural networks (FNNs) were used to predict dissolved oxygen (DO) using 10 parameters [7]. Support vector machines (SVMs), neural networks (NNs), deep neural networks (Deep NNs), and k-nearest neighbors (KNNs) were used to estimate the water quality, and the deep neural network attained the highest accuracy of 93% based on three parameters: turbidity, temperature, and pH [8].

LSTM and NARNET models were used in predicting the water quality index, and support vector machine, Naive Bayes, and K-nearest neighbor were used to classify the water quality index data. The algorithms were evaluated on some statistical parameters. NARNET model had a slightly better performance compared to LSTM model based on the obtained value, and support vector machine attained the best accuracy [9]. Auto-deep learning was compared with the conventional deep learning model to predict water quality based on water quality index. The conventional deep learning approach gave a slightly better performance, but adoption of auto-deep learning made finding an appropriate deep learning model easier and gave better performance minus manual intervention [10]. Effectiveness of eight artificial intelligence methods in predicting the water quality parameter in a dry desert environment was examined based on the 114 samples collected from six aquifers at different time periods. The ability of these methods was tested based on two scenarios and two different input combinations that is replacement of the classical computational method with modeling approach and lack or unavailability of data in critical cases [11]. The performance of artificial intelligence techniques like artificial neural network (ANN), group method of data handling (GMDH), and support vector machine (SVM) for predicting water quality components of Tیره River was investigated. The performance of the GMDH model was slightly less than that of the ANN and SVM [12].

However, the different techniques and models used have certain limitations. Collection of water samples from different regions for manual lab analysis is quite expensive; parameter sensors suggested in some works are also costly, and the existing models still remain in the black box nature. Many other studies used either few or too many parameters which is not efficient enough in predicting the water quality.



### 2.3 Ensemble Learning

Ensemble learning involves training multiple machine learning models (merging several weak base learners or models) to obtain better performance with improved feature analysis. This is done by combining model outputs and treating them as a committee of decision makers instead of using a single model. Every trained ensemble represents a single hypothesis and uses the multiple classifier systems to allow for hybridization of hypotheses differently induced by weak learners, which leads to better predictions for significant diversity among the single model. Tree-based models such as the random forest are the mainly used base learners in ensemble learning models [13], while a variety of boosting and bagging approaches have been also proposed. The boosting approach is more suitable for high-bias predictions, while the bootstrapping method is more efficient for high-variance predictions. Ensemble learning significantly minimizes errors like the misleading positive and negative predictions [14].

### 2.4 Explainable Artificial Intelligence

Most times, machine learning models act as a black box making it really hard to understand how the models come up with the predictions because developers are oftentimes unaware of what really goes on under the hood after the model has been given an input [15, 16]. Explainable artificial intelligence (XAI) includes the interpretability of prediction accuracy with the ability to trace data to the underlying logic and data, and it includes the reasons and justifications to support actionable recommendations for faster and more accurate decisions and provides an assessment of future business outcomes and allows scenario simulations to determine the best actions. In this study, we used the Shapley Additive Explanations (SHAP) [17, 18] to interpret the model performance in prediction of water quality. SHAP is a game theoretical approach published by Lundberg and Lee to explain ML model outputs by connecting optimal credit portions with local descriptions using the Shapley values. This average marginal contribution of a feature value over all possible coalitions describes the Shapley values which are unified feature importance measures derived by;

$$\varphi_i(q) = \sum_{S \subseteq N/i} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (q(S \cup i) - q(S))$$

where marginal contribution of the feature  $[q(S \cup i) - q(S)]$  is computed out of all the subsets  $S$  to obtain the Shapley value for a feature  $i$ ; thus, the model estimate is computed for all the subsets with and without the feature and summed to get the Shapley value for that to make the additive explanations feature [19, 20]. Shapley

values of the segments are approximated, which reflect their contribution to the prediction [21].

### 3 Methodology

#### 3.1 Dataset Description

We used the water potability dataset [22] consisting of various metrics of water quality for 3276 different water bodies. The dataset comprises nine independent features with the output column (dependent) feature being potability. The dataset features are described in Table 1 with their respective recommended ranges as per the World Health Organization (WHO) guidelines.

**Table 1** Dataset description

| Feature                      | Description                                                       | WHO limits                                                                                          |
|------------------------------|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| pH value                     | Acidic or alkaline condition of water status                      | 6.5–8.5                                                                                             |
| Hardness                     | Capacity of water to precipitate soap                             |                                                                                                     |
| Total dissolved solids (TDS) | How mineralized water is                                          | 500 mg/l (desirable) and 1000 mg/l (maximum)                                                        |
| Chloramines                  | Major disinfectants used in public water systems                  | Up to 4 mg/l or 4 ppm                                                                               |
| Sulfate                      | Naturally occurring substances found in minerals, soil, and rocks | 2700 mg/l (sea water); 3–30 mg/l (fresh water supplies); 1000 mg/l (in some geographical locations) |
| Organic_Carbon (TOC)         | Total quantity of carbon in organic compounds in clean water      | 2 mg/L (in treated/drinking water); < 4 mg/L (in source water which is use for treatment)           |
| Trihalomethanes              | Chemicals found in water treated with chlorine                    | Up to 80 ppm                                                                                        |
| Turbidity                    | Light emitting properties of water                                | 5.00 NTU                                                                                            |
| Conductivity                 | Iron concentration for electrical conductivity of water           | 400 $\mu$ S/cm                                                                                      |
| Potability                   | How safe water is for human consumption                           | 0 (for not potable) or 1 (for potable)                                                              |

### 3.2 Proposed Approach

An overview of the steps we have taken in training our models is summarized in a flow diagram in Fig. 1 using an appropriate open global water quality dataset obtained from Kaggle.

**Feature Engineering.** pH, trihalomethanes, and sulfate features had missing values, and anomalies were also detected during the data cleaning and preprocessing stage. Since missing values are handled depending on the nature of the problem to be solved, it would be extremely risky and inappropriate to choose filling in the missing values as an approach to handling missing values. We therefore avoided predicting missing values and instead removed all missing values for safety purposes. Using the PyCaret open-source library, we then performed anomaly detection to identify the outliers in the dataset. The safest way of handling outliers for water safety prediction was removing them since there should not be outliers in the dataset related to life. Over 110 anomalies were observed by evaluating the various dates which reflected the recorded cases that were juggled for water potability. This gave an insight of the potential existence of more instances of juggled cases of undrinkable water as drinkable. On performing datatype verification of the variables, it was observed that all features were numerical, and the target variable was imbalanced.

**Checking Feature Importance.** The feature importance of the various water quality metrics was checked in various ways. It is important to note that poor water quality can cause different diseases. Therefore, knowing which features are important when judging water quality will help in making public health decisions. We checked the feature importance in three different ways, namely feature importance with partial dependence, feature importance based on mean decrease in impurity, and feature importance based on feature permutation.

**Model Creation.** The new clean training dataset was used to train multiple classification algorithms, for example, decision trees, light gradient boosting machine

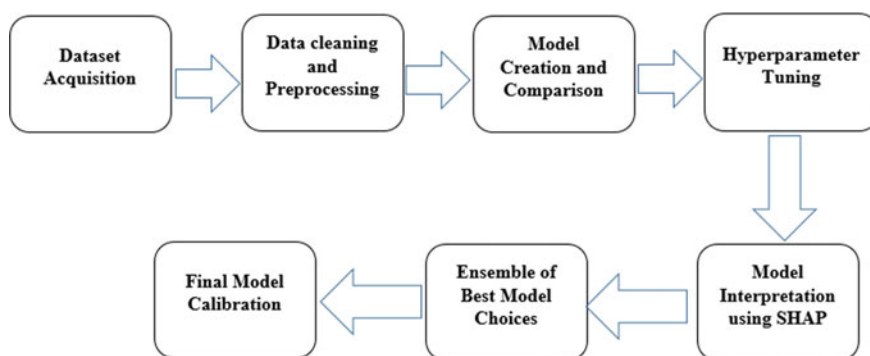


Fig. 1 Proposed approach

(LGBM), CatBoost, Naïve Bayes, random forests, extra trees, gradient boosting classifier, linear discriminant analysis, and the logistic regression models. Model comparison was performed and explainability done for a few outstanding tree-based classifiers.

**Ensemble Learning.** The best models that is LGBM, CatBoost, and random forest were ensembled to form a robust water quality prediction model which was trained on the same dataset to assess its binary classification performance before calibration for model generalizability.

## 4 Results and Discussion

### 4.1 *Feature Importance*

Feature importance provided insights into the dataset. We began by assessing the feature importance to water quality prediction in the perspective of the best classifiers for easy understanding during interpretation. From Fig. 2, it was observed that the pH feature has a large partial dependence at values between 6 and 8.5. For hardness, the value of partial dependence rapidly increases around 210. In Figs. 3 and 4, it was observed that pH and sulfate features were judged to be important features. The power of collective intelligence allows for a more accurate deduction that pH is a very important predictor of water quality since we have compared results from various approaches assessing feature importance.

### 4.2 *Model Creation and Comparison*

All the estimators in the model library were trained, and their performance is evaluated using cross-validation to output a score grid with all the average cross-validated scores as shown in Table 2.

The hyperparameters of the estimators were tuned to output a score grid with cross-validation scores by fold of the best selected model based on the optimization parameter. The performance is shown in Table 3.

### 4.3 *Model Interpretation by SHAP*

We deployed Shapley Additive Explanations (SHAP) to interpret the water quality predictions of the models. SHAP values show the impact of each feature whose comparative possession yields interpretation of predictions based on baseline values. Figures 5, 6, and 7 are SHAP summary plots which combine feature importance with

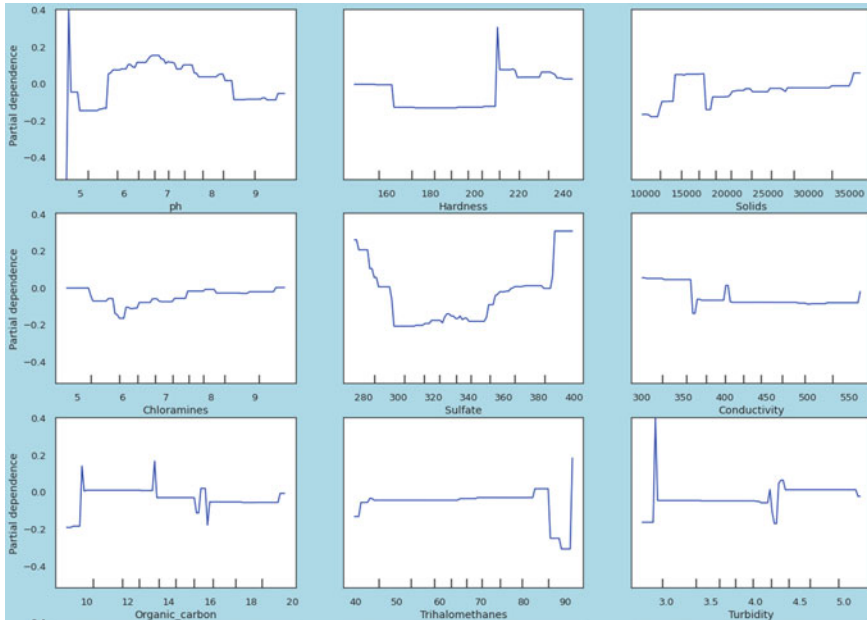


**Fig. 2** Partial display of feature importance with partial dependencies

feature impacts. Each point on the summary plot is a Shapley value of an instance per feature. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value of each instance. The color of the dots denotes the value of the feature (Blue: low value, red: higher value). Features are well-organized depending on their importance during the interaction. We observed that the feature sulfate was the most important feature for LGBM, while CatBoost and random forest indicated that pH is the most important of all. Therefore, from the summary plots, the top three influential variables in determining water potability are sulfate, pH, and hardness.

### 4.4 Ensemble Modeling

A soft voting technique was used to ensemble the 3 best models; LGBM, CatBoost, and random forest. Light gradient boosting machine (LGBM) is based on decision trees and boosting with a quicker training speed and higher efficiency while extending the gradient boosting algorithm by including an automatic feature selection and focusing on boosting larger gradient examples. Categorical boosting (CatBoost) utilizes the gradient descent framework, a complex ensemble learning technique that uses binary decision trees as base predictors in order to predict categorical features. And random forest is a decision trees-based classifier for predicting qualitative responses by dividing the predictor space into different and non-overlapping

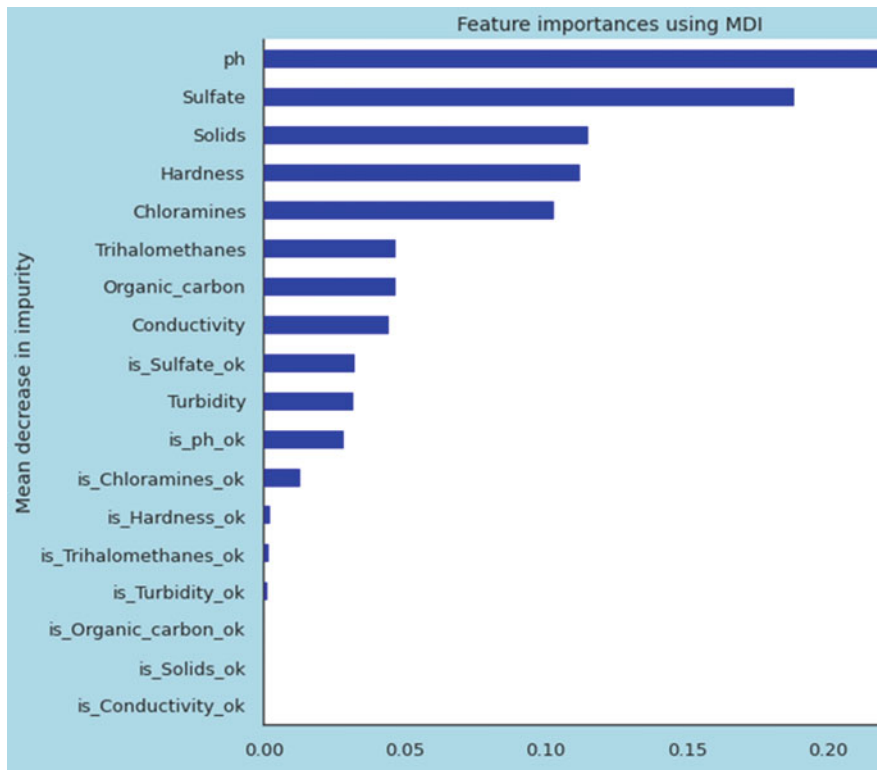


**Fig. 3** Feature importance using mean decrease in impurity

regions for the same prediction to be made for every observation in that region (majority group) during classification which can be regarded as Bayes classifier.

We plotted the ROC curve in Fig. 9 of the soft voting classifier that predicted water quality in comparison to other algorithms.

### 4.5 Final Model Calibration

On comparing performance of the individual models with the ensemble model, the ensemble soft voting model had the best performance. Therefore, the soft voting model is selected as the final model for performing calibration. Model calibration was done by calibrating probability of the estimators using isotonic or logistic regression. The accuracy of the model completed until calibration was over 90%. The final calibrated soft voting model trained the estimator on the entire dataset including the holdout set. The summary of results is presented in Table 4 with a confusion matrix to illustrate the classical results of the final calibrated model in Fig. 10.

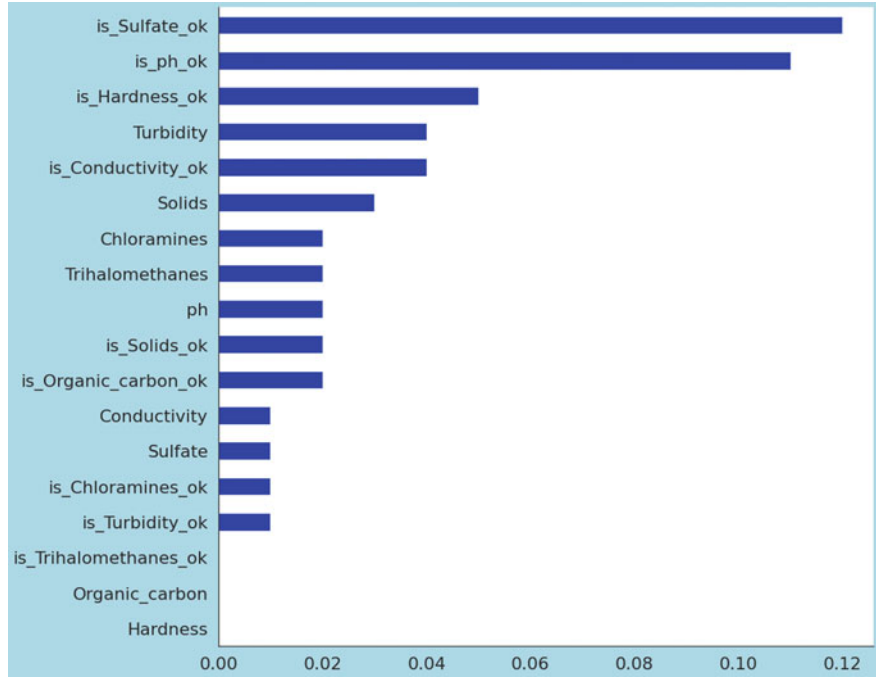


**Fig. 4** Feature importance based on feature permutation

**Table 2** Model comparison

| Model                           | Accuracy | AUC    | Recall | Precision | F1     |
|---------------------------------|----------|--------|--------|-----------|--------|
| Decision tree                   | 0.5988   | 0.5759 | 0.2949 | 0.4782    | 0.4756 |
| Light gradient boosting machine | 0.6423   | 0.6463 | 0.4120 | 0.5497    | 0.4667 |
| CatBoost                        | 0.6864   | 0.6857 | 0.3770 | 0.6609    | 0.4749 |
| Naïve Bayes                     | 0.6205   | 0.6106 | 0.3009 | 0.5057    | 0.3752 |
| Random forest                   | 0.6602   | 0.6586 | 0.4782 | 0.6191    | 0.3971 |
| Extra trees                     | 0.6310   | 0.6055 | 0.2949 | 0.5294    | 0.3751 |
| Gradient boosting               | 0.6378   | 0.6263 | 0.2716 | 0.5386    | 0.3582 |
| Linear discriminant analysis    | 0.6205   | 0.5709 | 0.1798 | 0.5239    | 0.2645 |
| Logistic regression             | 0.6175   | 0.4985 | 0.0038 | 0.0667    | 0.0073 |

## 5 Conclusion and Future Works

Investigative measures for evaluating water quality are of high relevance. This study proposed a robust generalizable calibrated ensemble learning model that is well suited for water quality prediction. To verify the proposed model’s accuracy and feasibility, a Kaggle stationed dataset is used. Results indicate high classification accuracy of

**Table 3** Performance evaluation using tuned parameters

| Fold | Accuracy | AUC    | Recall | Precision | <i>F1</i> |
|------|----------|--------|--------|-----------|-----------|
| 0    | 0.6418   | 0.6657 | 0.3333 | 0.5484    | 0.4146    |
| 1    | 0.6194   | 0.5781 | 0.2941 | 0.5000    | 0.3704    |
| 2    | 0.5896   | 0.6226 | 0.1373 | 0.3889    | 0.2029    |
| 3    | 0.5746   | 0.4759 | 0.1765 | 0.3750    | 0.2400    |
| 4    | 0.6194   | 0.6283 | 0.2115 | 0.5238    | 0.3014    |
| 5    | 0.5597   | 0.5210 | 0.2500 | 0.3939    | 0.3059    |
| 6    | 0.6466   | 0.5903 | 0.3922 | 0.5556    | 0.4598    |
| 7    | 0.6165   | 0.5527 | 0.1765 | 0.5000    | 0.2609    |
| 8    | 0.5714   | 0.5544 | 0.0980 | 0.3125    | 0.1493    |
| 9    | 0.6541   | 0.6786 | 0.4706 | 0.5581    | 0.5106    |
| Mean | 0.6093   | 0.5868 | 0.2540 | 0.4656    | 0.3216    |
| Std  | 0.0319   | 0.0605 | 0.1121 | 0.0848    | 0.1098    |

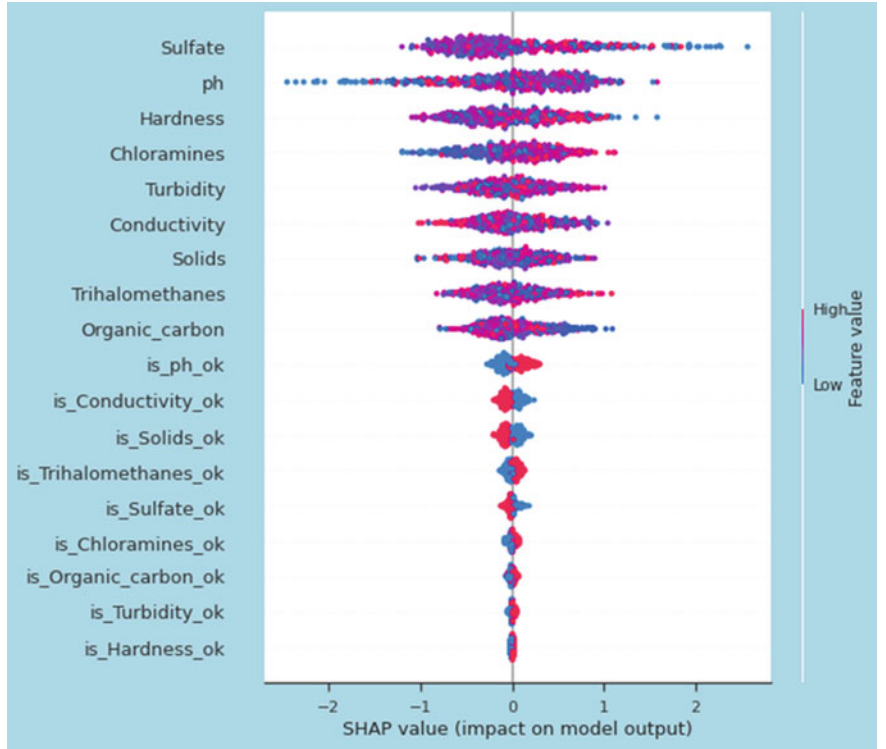


**Fig. 5** Tuned LGBM model interpretation



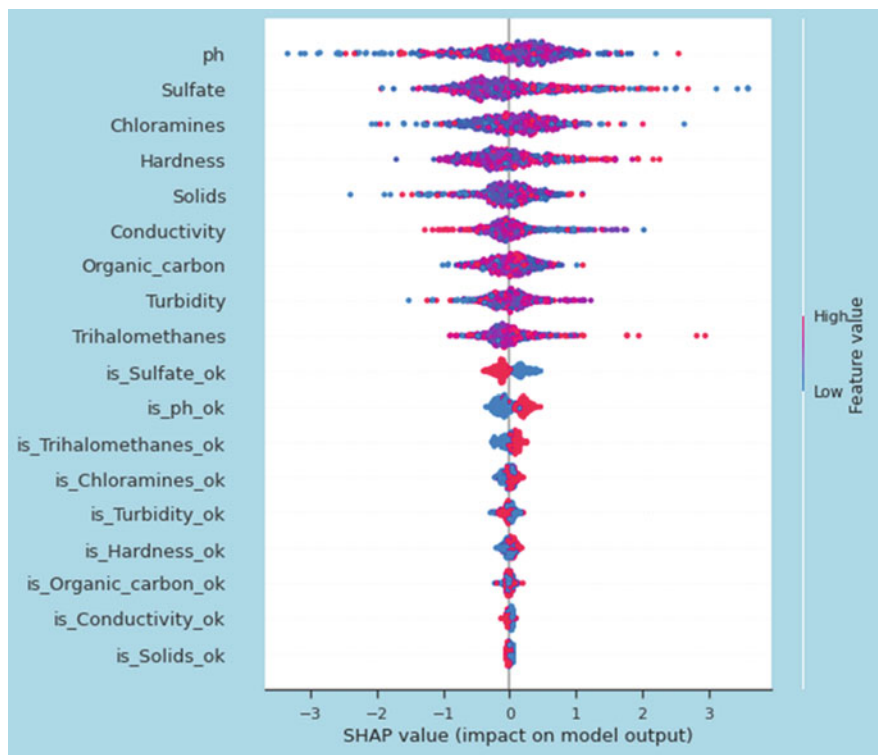


**Fig. 6** Tuned CatBoost model interpretation

the proposed model with a high recall and precision. Additionally, the ease of use with our SHAP framework is a means to make the low-level data easier to use and comprehend. However, the model may need extra investigations with other relevant datasets to have its general performance characteristics validated. We believe our initial results may warrant translation of the study outcomes into full-scale Public Health practice by guiding agencies and governments on management, policy, and decision-making concerning water resources.

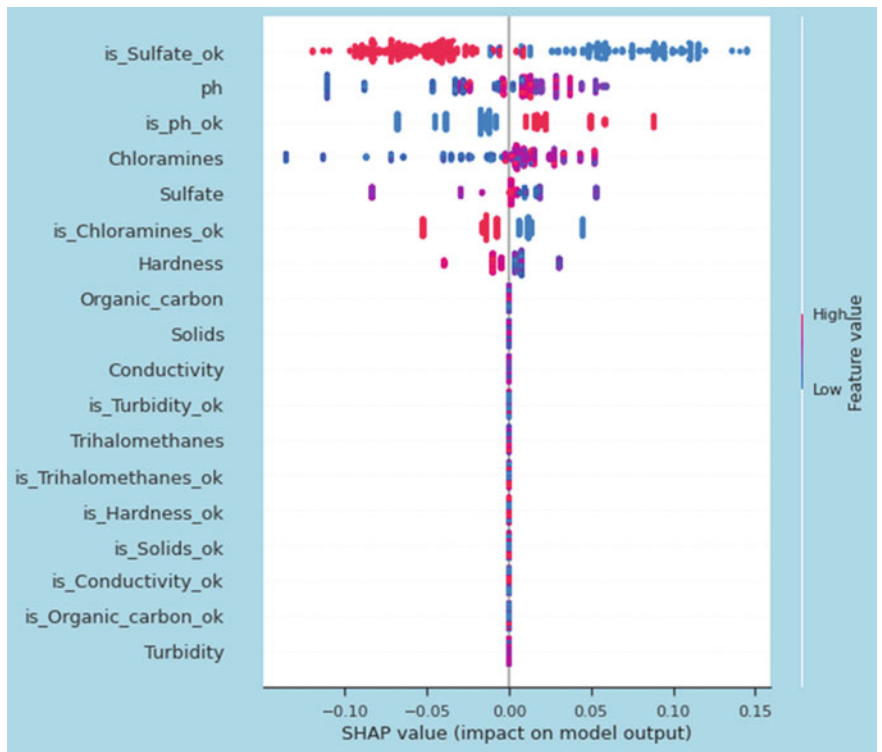


Fig. 7 Tuned random forest model interpretation

Fig. 8 Confusion matrix for the soft voting classifier

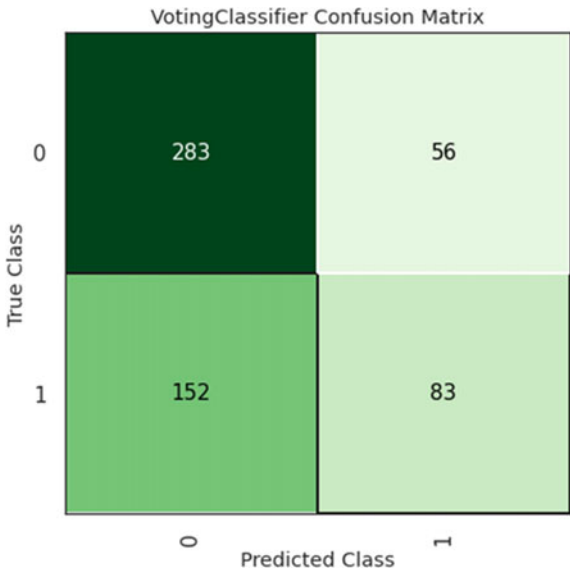


**Fig. 9** ROC curves for the soft voting classifier

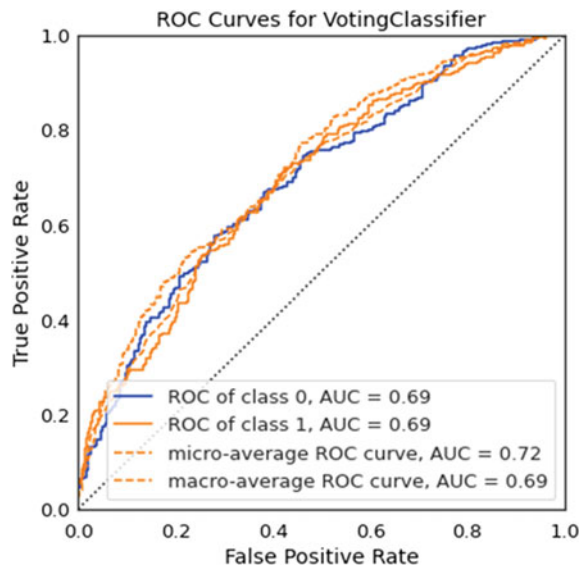
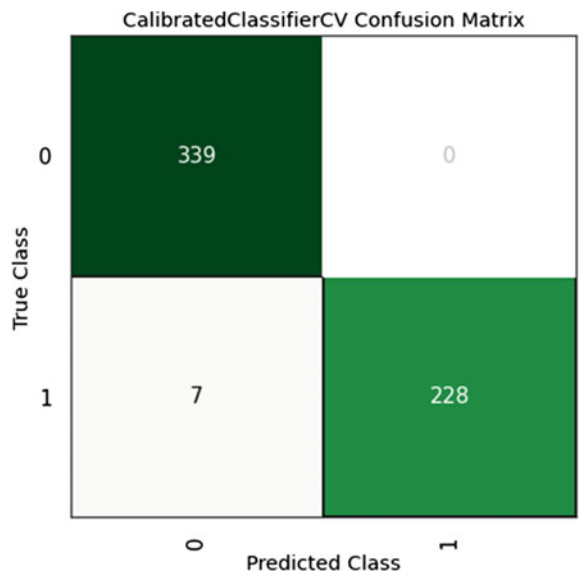


**Table 4** Classification report for the calibrated model

| Metric          | Precision | Recall | <i>F</i> 1-score | Support |
|-----------------|-----------|--------|------------------|---------|
| Not-potable (0) | 0.980     | 1.000  | 0.990            | 339     |
| Potable (1)     | 1.000     | 0.970  | 0.985            | 235     |

Accuracy score: 0.988

**Fig. 10** Confusion matrix for the final calibrated model



## References

1. Loucks DP, van Beek E (2017) Urban water systems. In: Water resource systems planning and management. Springer International Publishing, Cham, pp 527–565
2. [https://www.un.org/waterforlifedecade/pdf/unwdpac\\_biennial\\_report\\_2010\\_2011.pdf](https://www.un.org/waterforlifedecade/pdf/unwdpac_biennial_report_2010_2011.pdf)
3. Ubah JI, Orakwe LC, Ogbu KN, Awu JI, Ahaneku IE, Chukwuma EC (2021) Forecasting water quality parameters using artificial neural network for irrigation purposes. *Sci Rep* 11(1):24438
4. Sahaya Vasanthi S, Adish KS (2019) Application of artificial neural network techniques for predicting the water quality index in the Parakai lake, Tamil Nadu, India. *Appl Ecol Environ Res* 17(2):1947–1958
5. Ahmed U, Mumtaz R, Anwar H, Shah AA, Irfan R, García-Nieto J (2019) Efficient water quality prediction using supervised machine learning. *Water (Basel)* 11(11):2210
6. Daud MK, Nafees M, Ali S, Rizwan M, Bajwa RA, Shakoor MB et al (2017) Drinking water quality status and contamination in Pakistan. *Biomed Res Int* 2017:1–18
7. Ranković V, Radulović J, Radojević I, Ostojić A, Čomić L (2010) Neural network modeling of dissolved oxygen in the Gruža reservoir, Serbia. *Ecol Modell* 221(8):1239–1244
8. Shafi U, Mumtaz R, Anwar H, Qamar AM, Khurshid H (2018) Surface water pollution detection using internet of things. In: 2018 15th international conference on smart cities: improving quality of life using ICT & IoT (HONET-ICT). IEEE
9. Aldhyani THH, Al-Yaari M, Alkahtani H, Maashi M (2020) Water quality prediction using artificial intelligence algorithms. *Appl Bionics Biomech* 2020:6659314
10. Prasad DVV, Venkataramana LY, Kumar PS, Prasannamedha G, Harshana S, Srividya SJ et al (2022) Analysis and prediction of water quality using deep learning and auto deep learning techniques. *Sci Total Environ* 821:153311
11. Kouadri S, Elbeltagi A, Islam ARMT, Kateb S (2021) Performance of machine learning methods in predicting water quality index based on irregular data set: application on Illizi region (Algerian southeast). *Appl Water Sci* 11(12)
12. Haghiabi AH, Nasrolahi AH, Parsaie A (2018) Water quality prediction using machine learning methods. *Water Qual Res J Can* 53(1):3–13
13. Park J, Lee WH, Kim KT, Park CY, Lee S, Heo T-Y (2022) Interpretation of ensemble learning to predict water quality using explainable artificial intelligence. *Sci Total Environ* 832:155070
14. Shamshirband S, Jafari Nodoushan E, Adolf JE, Abdul Manaf A, Mosavi A, Chau K-W (2019) Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters. *Eng Appl Comput Fluid Mech* 13(1):91–101
15. Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A et al (2020) Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf Fusion* 58:82–115
16. Başağaoğlu H, Chakraborty D, Lago CD, Gutierrez L, Şahinli MA, Giacomoni M et al (2022) A review on interpretable and eXplainable Artificial Intelligence in hydroclimatic applications. *Water (Basel)* 14(8):1230
17. Wang R, Kim J-H, Li M-H (2021) Predicting stream water quality under different urban development pattern scenarios with an interpretable machine learning approach. *Sci Total Environ* 761:144057
18. Hellen N, Marvin G (2022) Explainable AI for safe water evaluation for public health in urban settings. In: 2022 international conference on innovations in science, engineering and technology (ICISSET). IEEE
19. Hellen N, Marvin G (2021) Interpretable feature learning framework for smoking behavior detection. *arXiv [csCV]*
20. Marvin G, Alam MGR (2021) Cardiotocogram biomedical signal classification and interpretation for fetal health evaluation. In: 2021 IEEE Asia-Pacific conference on computer science and data engineering (CSDE). IEEE

21. Vermeire T, Brughmans D, Goethals S, de Oliveira RMB, Martens D (2022) Explainable image classification with evidence counterfactual. *Pattern Anal Appl*. <http://dx.doi.org/10.1007/s10044-021-01055-y>
22. Kadiwal A (2021) Water quality

## Chapter 20

# Diagnosis of Autism Spectrum Disorder Through Eye Movement Tracking Using Deep Learning



Nasirul Mumenin , Md. Farhadul Islam ,  
Md. Reasad Zaman Chowdhury , and Mohammad Abu Yousuf

**Abstract** Autism spectrum disorder is a complex neurodevelopmental disease with many sub-types that make diagnosis difficult. Several studies have shown that children with autism have difficulty maintaining their attention and have a less-focused vision. The purpose of this article is to discuss the automated identification of autism spectrum disorder using eye-tracking scanpath images. A convolutional neural network has been implemented to extract important features from the images and classify them efficiently. Accuracy, sensitivity, specificity,  $F1$  score, AUC are used to evaluate the performance of the model. The study has been conducted on both RGB images obtained from a publicly available dataset and on converted grayscale images. The proposed model has achieved 97.41% accuracy while using grayscale images which outperforms the existing models.

**Keywords** Autism spectrum disorder · Eye movement tracking · Convolutional neural network · Data augmentation · Grayscale image · Scanpath

## 1 Introduction

Autism spectrum disorder (ASD) is a term used to describe a group of complex neurodevelopment conditions such as impairments in social communication, deficits in social interactions and restricted, and repetitive behavioral patterns [1]. It is considered among the most common developmental disorders, affecting about 1% of

---

N. Mumenin (✉)

Bangladesh University of Professionals, Dhaka, Bangladesh  
e-mail: [nmmouno@gmail.com](mailto:nmmouno@gmail.com)

Md. F. Islam

BRAC University, Dhaka, Bangladesh

Md. R. Z. Chowdhury

Ahsanullah University of Science and Technology, Dhaka, Bangladesh

M. A. Yousuf

Jahangirnagar University, Savar Union, Bangladesh

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_20](https://doi.org/10.1007/978-981-19-7528-8_20)

251

the world's population [2]. However, there are no uniform biomarkers for ASD, and individuals may pose different symptoms. Typically, persons diagnosed with ASD show social communication deficits and find it hard to retain their attention on certain tasks [3]. Unfortunately, such unpleasant symptoms may hinder social and personal development and substantially influence their lives.

Currently, there is no treatment method that has been created to fully recover from ASD. But, early detection and proper treatment can help improve the situation [4]. In addition, an early diagnosis might aid in the recovery of the symptoms and help prevent other psychological illnesses. As a result, adequate support and attention need to be given to autistic individuals. However, detection of ASD is still very challenging due to the wide variety of severity levels and signs found on ASD patients. Some widely used clinical methods to evaluate the symptoms include functional magnetic resonance imaging (fMRI), electroencephalography (EEG), metabolic disorder testing, genetic testing, facial expression analysis, etc. Recently, eye tracking has been used to diagnose ASD, proven to be an effective tool. Eye tracking is such a system that can be used for recording, monitoring, and taking measurement of eye movements or absolute point of gaze (POG), that is the point which indicates where the eye gaze is giving attention in the visual scene. The eye-tracking system aims to analyze the perceptual aspects of ASD and finds the unusual visual attention that contributes to the clinical features of ASD. Eye tracking has been very effective in diagnosis because of the fact that abnormal gaze patterns are one of the critical features of ASD. As a result, eye tracking can be utilized to identify ASD features and help improve the diagnosis.

Deep learning (DL) is a machine learning and artificial intelligence technique that mimics how people acquire information. Deep learning is viewed of as a method to automate predictive analytics at its most basic level. Deep learning algorithms are layered in a hierarchy of increasing complexity and abstraction, unlike conventional machine learning algorithms, which are linear. A convolutional neural network (CNN) is a part of deep learning that is able take an image as input, assign importance (learnable weights and biases) to various aspects, and distinguish between them [5]. When compared to the existing classification methods available, the amount of pre-processing needed by a CNN is comparatively less. In such cases where existing techniques require human intervention of filters, CNN has the ability learn these features/characteristics if provided with enough data and training. Through the use of relevant filters, a CNN can capture the spatial and temporal dependencies in an image. As the number of parameters are reduced and the ability to reuse the weights, the CNN can perform effective fitting to the picture dataset. In other words, the network may be taught to better comprehend the image's complexity. For such abilities, CNN is used for various disease detection tasks, specifically from image data [6]. In recent times, AI, machine learning, and deep learning are playing a big role in diagnosis and monitoring of autism patients [7].

Our research is based on the fact that individuals with ASD pose an unusual gaze pattern, and this pattern can be identified from the visual representation of eye tracking [8]. In this study, a deep CNN is developed to classify autism spectrum disorder from the eye-tracking scanpaths. One of the advantage of the proposed

approach is that the overall diagnosis process is automated and based on the eye-tracking scanpaths image of an individual. Therefore, the proposed model can be used to build an automatic and efficient diagnosis system of ASD. The main contribution of our research are as follows:

- Using CNN to automatically extract the important features from the scanpath images.
- Building a lightweight and effective deep learning model to diagnose ASD effectively. Our model has achieved higher accuracy than previous works.
- Testing both colored (RGB) and grayscale image to understand the impact of these two type of images on effective diagnosis.

The remainder of the manuscript is laid out as follows: Sect. 2 provides a literature review that includes studies on classification of ASD by eye tracking using various techniques, such as deep learning and machine learning. We present information about dataset, data preprocessing and a CNN model for detecting ASD from scanpath images in Sect. 3. Section 4 provides details on experiment and result analysis of the model. Finally, conclusion is drawn in Sect. 5.

## 2 Literature Review

Many researches on the identification of autism spectrum disorder have been conducted over the years, given the importance of the problem. However, some social and behavioral symptoms of ASD such as social interaction, lack of attention, eye contact, face recognition can be harder to identify in the early stages. Therefore, traditional clinical methods have their limitations in early diagnosis which can hinder the recovery process. Due to these varying degrees of symptoms severity, the problem is quite challenging and needs further research.

Recently, Akter et al. [9] established a method to identify ASD by analyzing facial features using transfer learning. They developed an improved transfer learning based face recognition framework that can identify individuals with ASD. Various machine learning and deep learning models were applied and out of those, improved MobileNet-V1 model achieved the best accuracy of 90.67%. Also, they used k-means clustering algorithm to find out the different ASD groups by varying the values of  $k$  from 2 to 10. Later, the proposed model is applied to predict the sub-types which showed accuracy of 92.10% when  $k = 2$ . Although the proposed model achieved a good accuracy, facial recognition-based system might not work in some cases, particularly in the earlier development stage. Raj et al. [10] performed different machine learning and deep learning techniques such as Naive Bayes, support vector machine, logistic regression, KNN, artificial neural network (ANN), and CNN to detect ASD in children. For evaluating the proposed techniques, they used three publicly available datasets from the UCI repository, each belonging to different age groups. The dataset used in this research contains twenty common attributes, primarily demographic and



screening-related information. From their experiment, CNN performed better than the other models with an accuracy of 99.53%, 98.30%, and 96.88% for the adult, children, and adolescents data, respectively. Xie et al. [11] developed a two-stream deep learning network to detect visual attention in ASD. The proposed model was adapted from two identical VGGNets modified from VGG16 architecture. For this study, data were collected from the OSIE database, which includes 700 images of eye movement patterns of those with ASD and TD. The proposed model achieved accuracies of 0.95 and 0.85 in the leave-one-out and 13-fold cross-validation, respectively. Also, they applied t-SNE visualization and identified some pixel-level features within the images.

Recently explainable AI (XAI) has been highly influential in interpreting models and being employed in various applications. For example, Biswas et al. [12] proposed an explainable AI-based (XAI) SVM model in a recent study to find the most relevant characteristics that influence the prediction of ASD in children. They used Log and Sine FT methods on the toddler dataset to evaluate the performance. Their proposed model achieved an accuracy of  $98.27 \pm 1.12$  and  $98.47 \pm 1.10$  using Log FT and Sine FT methods, respectively. Later, the feature importance score was generated to find the most important features contributing to the prediction. This research shows that correctly optimized ML models can accurately predict whether a person has ASD or not. Many researchers have conducted experiment with eye tracking to study ASD. Shihab et al. [13] conducted a study to detect ASD in children and adults. They applied principal component analysis (PCA) and used an unsupervised classification method to identify ASD and TD. The dataset used in this work includes eye-tracking data of individuals aged between 4 to 60 years who were diagnosed with ASD. The proposed method achieved a sensitivity of 78.6% and specificity of 82.47% while detecting adults with ASD. On the other hand, classifying children with ASD gave a sensitivity of 87.5% and specificity of 95.7%. In recent studies, eye tracking has been coupled with AI to identify ASD. For instance, Carette et al. [14] used machine learning to predict autism spectrum disorder based on the visual representation of eye-tracking scanpaths. Dataset was constructed from the eye-tracking scanpaths of children of two groups: ASD diagnosed and non-ASD diagnosed. Several machine learning models including Naïve Bayes, logistic regression, SVM, and random forests and artificial neural network (ANN) were used to test the predictability of ASD. The experimental result suggests that simple ML models combined with eye-tracking scanpaths can predict ASD with high accuracy. With ANN classifiers, they achieved maximum accuracy of beyond 90%. In Bangladesh, Satu et al. [15] looked at the differences between normal and autism in divisional areas. They took individual samples of children ranging in age from 16 to 30 months from a variety of residents. The data was then preprocessed, and frequent characteristics were classified based on particular areas. Different tree-based methods were investigated in order to determine which one was the most effective classifier. J48 outperformed the rest of the classifiers in terms of results. Finally, 8 out of 23 characteristics were needed to distinguish normal and autism in different Bangladeshi areas.

Some diagnosis procedures on ASD such as interview settings have the possibility to exhibit undesirable bias toward the patient. Also, facial recognition-based system

has its limitations in detecting ASD in the early stages. On the contrary, recent researches on ASD showed a strong correlation between eye movements and ASD. In addition, some studies found that children with ASD often show unusual gaze patterns. In our work, we combined deep learning techniques with the eye movement data to overcome the existing issues in traditional methods and build a system that can accurately classify ASD.

3    Methodology

In this section, we have described the workflow of our work along with a brief discussion of our proposed method. The workflow is divided into some key parts.

Figure 1 represents workflow of the following method. In this section, we discuss about the dataset collection and analysis, data processing, deep learning model creation. Training and testing of the model and performance analysis is discussed in the next section.

3.1    Dataset Collection and Analysis

The dataset has been collected from the Figshare data repository. It is a publicly available dataset for researchers to conduct research on this topic. The dataset has been created by Carrette et al. [16]. The authors gathered 59 individuals that includes, 38 males and 21 females for collecting data. The participants were 30 ASD and 29 non-ASD children. One of the main objectives of their work was to early detection of autism, for that the age range of the participants was kept from 3–13 years. The primary equipment for eye-tracking studies was the SMI RED mobile. The raw eye-tracking data consisted of a set of (x, y) coordinates corresponding to the participant’s eye travel across the screen. The authors then created visualization from the coordi-

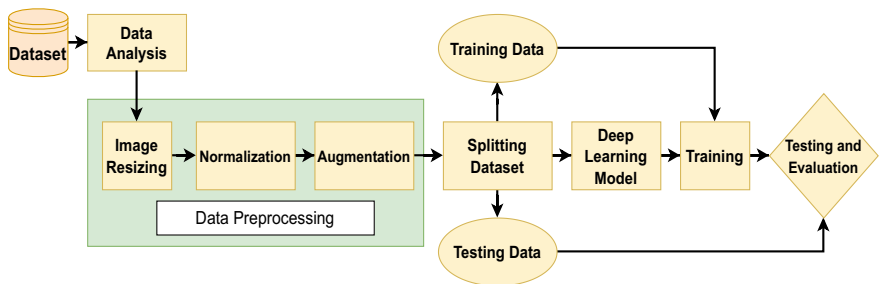


Fig. 1    Illustration of workflow

notes. Python was used to completely implement the data transformation procedure. The visualizers were created by Matplotlib library. Scanpath visualization resulted in a collection of 547 pictures that have 219 ASD and 328 non-ASD images.

### 3.2 Data Preprocessing

After the data collection, the pictures are randomly divided into 8:1:1 training, validation, and test datasets, which also maintains the ratio of autism and non-autism classes in each sub-datasets. We have preprocessed the eye-tracking scanpaths images from the dataset by converting them from RGB (three channel) to grayscale (one channel) images, increasing their brightness and resizing them into  $100 \times 100$  format. In Fig. 2, augmented images have been shown along with original image. This channel reduction scales the shape of input image from  $(100 \times 100 \times 3)$  to  $(100 \times 100 \times 1)$  which also reduces the number of parameters in the input layer.

Augmentation has been applied to generate images with various changes of the eye-scanpath visualizations. Additional 2933 images were added to the collection, after the augmentation process. The Keras library [17] provides an easy-to-use API for data augmentation, which makes the augmentation procedure significantly simplified. In our augmentation process, we use 3 different augmentation techniques which are rotation, flipping, and scaling. The method from Keras library randomly flips and rotates the images. But, we zoomed in the images with a fixed percentage of 10, which enables our model to classify the scanpaths more easily. After that, the images were converted into gray scale, while original rgb images are saved in a separate file. In Fig. 3, grayscale image has been shown along with original rgb image.

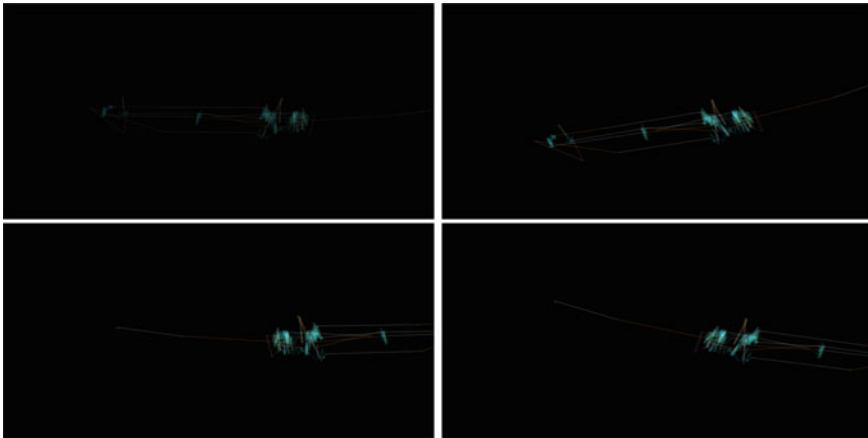


**Fig. 2** Samples of data augmentation. The top left-sided image represents original image from dataset, and other images are example of augmentation

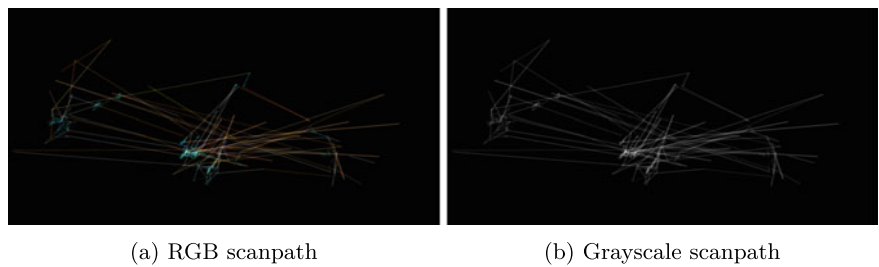


**Fig. 3** Illustration of RGB and grayscale scanpath

**3.3    Model Architecture**

The goal of this work is to get efficient result while keeping the numbers of parameters as low as possible. Our CNN model starts with a picture with the resolution of  $100 \times 100$  pixels. Then, we use five 2D convolutional layers and with  $3 \times 3$  kernel size. Also, to avoid overfitting in the model, we added a batch normalization layer after each convolutional layer. Besides that, we use a 2D max pooling layer after each batch normalization layer with a pool size of  $2 \times 2$  to reduce the computational cost. And for all convolutional layers, we use default strides which is (1,1).

We have trained the model on both grayscale (one channel) and RGB (three channel) images, which makes a difference in the size of parameters of the input layer of the model. As the activation function, we use ReLU as the non-saturation of its gradient accelerates the convergence of stochastic gradient descent (SGD) compared to the Sigmoid/Tanh function. After that, we add fully connected (FC) layers with 384 nodes to the CNN and use 30% dropout from that layer. In the output layer, we use the sigmoid activation function as a classifier for the network. Sigmoid function is added at the end of the output since here all the nodes are classified. The input to the function is transformed into a value between 0 and 1. In the sigmoid

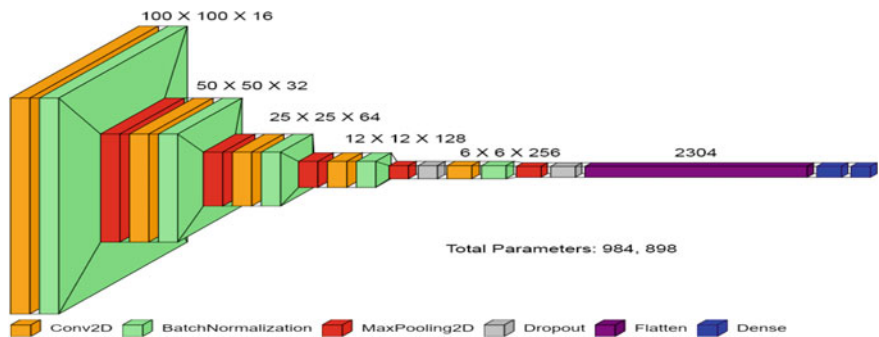


**Fig. 4** Visual representation of proposed CNN architecture

activation function, inputs that are much larger than 1 are transformed to the value 1; similarly, values much smaller than 0 are scaled to 0. The model consists of 984,898 parameters for grayscale images and 985,186 parameters for RGB images. Figure 4 is the visual representation of our proposed architecture.

4 Result Analysis and Discussion

4.1 Experimental Setup

The training and testing procedures for this CNN architecture are built using the Python packages including Tensorflow, Keras, Matplotlib, Pillow, and OpenCV. The models are trained and tested on an NVIDIA GeForce RTX 3080 Ti with a performance of 34.1 TeraFLOPs.

4.2 Experimentation

The model has been trained and tested on both color and grayscale images. Converting the colored images from the dataset into grayscale images reduced the size or number of features of the images. The training dataset, validation dataset, and testing dataset consisted of, respectively, 2784 images, 348 images, and 348 images. The hyperparameters of the model, which are number of training epochs, mini batch sizes, learning rate, have been tuned according to the results obtained. Table 1 provides the optimal hyperparameters used for this experiment.

4.3 Result Analysis

Several performance evaluation metrics such as accuracy, specificity, sensitivity, loss, *F1* score, AUC are used to compare and validate the model’s performance. Although accuracy is the most common metric used in classification tasks, we used several met-

Table 1 Hyperparameters of the proposed model

| Hyper parameter  | Value    |
|------------------|----------|
| Training epochs  | 35       |
| Mini batch sizes | 32       |
| Learning rate    | 0.000005 |
| Dropout          | 30%      |

rics to evaluate our model from different perspectives. Different evaluation metrics used in this research can be expressed using the following equations:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (2)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$F1 \text{ Score} = \frac{2 \times \text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}} \quad (5)$$

Here, TP and TN represent the data points that are correctly classified as true and false, respectively. On the other hand, FP and FN refer to the data points that the model inaccurately classifies as true or false.

We evaluated the performance for both color and grayscale images. Table 2 depicts the value of different evaluation metrics for color and grayscale images. It is clearly evident from different evaluation metric scores that grayscale images can be as useful as colored images in ASD prediction with a reduced number of channels.

After training and validating the model, it has been tested using the test set. The model has achieved **99.10%** training accuracy, **97.68%** validation accuracy, and **97.41%** testing accuracy for grayscale images. For the RGB images, the training accuracy, validation accuracy, and testing accuracy are **99.02%**, **97.50%**, and **96.55%**, respectively. Our model achieves **97.14%** test specificity and **97.60%** test sensitivity values for grayscale images and **95.17%** test specificity and **97.12%** test sensitivity values for RGB images.

The *F1* score and AUC score are also dominant than the methods mentioned in table 2. having **0.9783** and **0.9966** for grayscale images and **0.9712** and **0.9926** for RGB images, respectively. Figure 5 shows the comparison between training and validation where the experiment was conducted using grayscale images. While Fig. 6 shows the comparison between training and validation where the experiment was conducted using rgb images. After analyzing the result, it is found that model per-

**Table 2** Results obtained

| Image type | Accuracy | Specificity | Sensitivity | Loss   | <i>F1</i> Score | AUC Score |
|------------|----------|-------------|-------------|--------|-----------------|-----------|
| Grayscale  | 0.9741   | 0.9714      | 0.9760      | 0.0643 | 0.9783          | 0.9966    |
| RGB        | 0.9655   | 0.9571      | 0.9712      | 0.0687 | 0.9712          | 0.9926    |

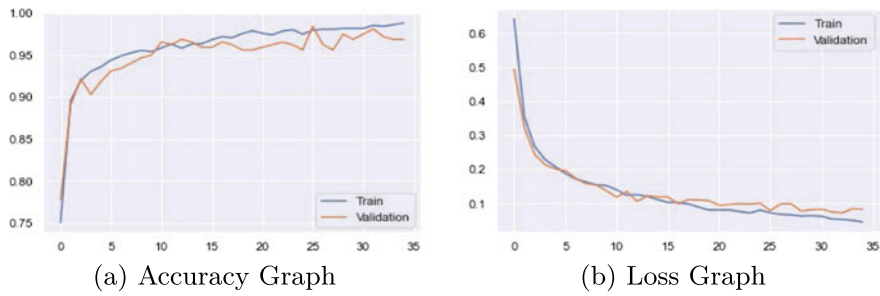


**Fig. 5** Training and validation graph on grayscale images

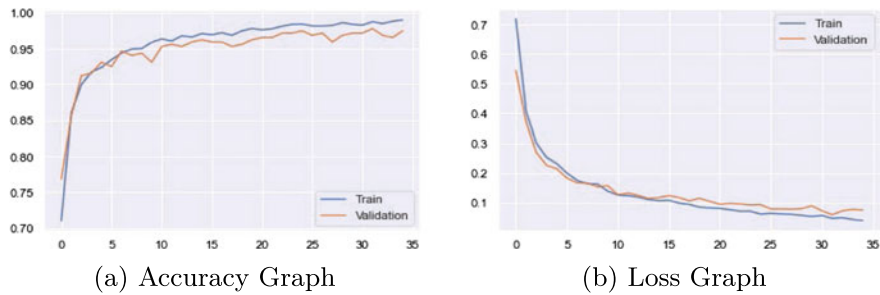


**Fig. 6** Training and validation graph on RGB images

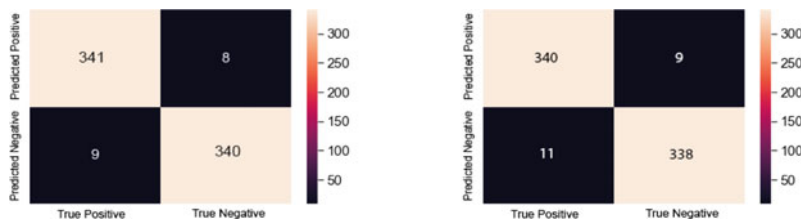


**Fig. 7** Confusion matrix of our work

forms slightly better with grayscale images then the rgb images even though the rgb images contains a greater number of parameters.

Figure 7a shows the confusion matrix while grayscale images were used, and Fig. 7b shows the confusion matrix while rgb images was used.

In previous works, the authors have assessed their work using just a few measures, like as accuracy and AUC. And also, all of the previous experiments were conducted on grayscale images. Carette et al. [16] employed a logistic regression model with an AUC of 81.9%. Again, they examined the eye-tracking scanpaths dataset and discovered an AUC of around 92% for a 200-neuron artificial neural network (ANN) [14]. Elbattah et al. [18] achieved a 94% accuracy rate when using k-means clustering

**Table 3** Comparison among different works.

| Author                            | Model information                      | Accuracy | AUC   |
|-----------------------------------|----------------------------------------|----------|-------|
| Proposed method (grayscale image) | Deep CNN                               | 0.9741   | 0.996 |
| Carette et al. [16]               | Logistic regression                    | –        | 0.819 |
| Carette et al. [14]               | Single layer artificial neural network | –        | 0.92  |
| Elbattah et al. [18]              | k-means clustering                     | 0.94     | –     |
| Elbattah et al. [19]              | Variational autoencoders               | 0.70     | 0.76  |
| Akter et al. [7]                  | Clustering and multilayer perceptron   | 0.87     | 0.79  |

methods. Then, Elbattah et al. [19] experimented using a variational autoencoder for implementing image augmentation and later classification. They achieved 70% accuracy on average. Lastly, Akter et al. [7] conducted experiment using a number of machine learning algorithms and multilayer perceptron (MLP). They divided the dataset into four different cluster using k-means clustering. The algorithms are then applied to check their efficiency. In cluster 1, MLP achieved 87% accuracy. They used a number of evaluation metrics, including accuracy, sensitivity, specificity,  $f1$ -measure, g-mean, AUC, fall out, and miss rate. Table 3 provides comparison among results obtained by existing works.

## 5 Conclusion and Future Works

In this paper, we have presented a CNN model that is able to classify ASD people from non-ASD people using grayscale eye-tracking scanpath images. Various evaluation metrics have been used to evaluate the performance of the model. We have tested the model with both rgb and grayscale images and found that the model performs better with grayscale images. The proposed method may be beneficial to provide a early indications regarding further testing that should be considered for a kid who exhibits symptoms of this condition. The proposed method can be used in conjunction with a desktop computer or laptop to screen children at an early age. It would be a way to identify children with ASD more quickly and efficiently by sending relevant data to a specialist, who would then conduct a more in-depth exam if required. However, the method described here is not intended to take the role of medical experts. The assistance provided is limited to an automated system that focuses on the most vulnerable children as quickly as possible and may point to erroneous diagnoses.

Additionally, a research comparing to other machine learning methods, such as recurrent neural networks should be performed to determine their overall efficiency. Another difficulty is the little quantity of training data required for a neural network method. Uncertainty and overfitting may be minimized or eliminated by collecting more data for training. Additionally, we must keep in mind that incidence of autism



spectrum condition may vary. Our current solution simply provides information on an individual's autistic status, whether they are diagnosed or not.

## References

1. Wing L, Gould J (1979) Severe impairments of social interaction and associated abnormalities in children: epidemiology and classification. *J Autism Dev Disord* 9:11–29
2. CDC. *Data and Statistics on Autism Spectrum Disorder* | CDC en-us. Sept 2020. <https://www.cdc.gov/ncbddd/autism/data.html> (2021)
3. Christensen DL et al (2018) Prevalence and characteristics of autism spectrum disorder among children aged 8 years-autism and developmental disabilities monitoring network, 11 sites, United States, 2012. *MMWR Surveill Summ* 65:1
4. Koegel LK, Koegel RL, Ashbaugh K, Bradshaw J (2014) The importance of early identification and intervention for children with or at risk for autism spectrum disorders. *Int J Speech-Lang Pathol* 16:50–56
5. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: 2017 International conference on engineering and technology (ICET), pp 1–6
6. Miah MBA, Yousuf MA (2015) Detection of lung cancer from CT image using image processing and neural network. In: 2015 International conference on electrical engineering and information communication technology (ICEEICT), pp 1–6
7. Akter T, Ali MH, Khan MI, Satu MS, Moni MA (2021) Machine learning model to predict autism investigating eye-tracking dataset. In: 2021 2nd International conference on robotics, electrical and signal processing techniques (ICREST), pp 383–387
8. Falck-Ytter T, Bölte S, Gredebäck G (2013) Eye tracking in early autism research. *J Neurodevelopmental Disord* 5:1–13
9. Akter T et al (2021) Improved transfer-learning-based facial recognition framework to detect autistic children at an early stage. *Brain Sci* 11:734
10. Raj S, Masood S (2020) Analysis and detection of autism spectrum disorder using machine learning techniques. *Procedia Comput Sci* 167:994–1004
11. Xie J et al (2019) A two-stream end-to-end deep learning network for recognizing atypical visual attention in autism spectrum disorder. arXiv preprint [arXiv:1911.11393](https://arxiv.org/abs/1911.11393)
12. Biswas M, Kaiser MS, Al Mamun S, Hossain M, Rahman M (2021) An XAI based autism detection: the context behind the detection in (Sept 2021). ISBN: 978-3-030-86992-2
13. Shihab AI, Dawood FA, Kashmar AH (2020) Data analysis and classification of autism spectrum disorder using principal component analysis. *Adv Bioinf*
14. Carette R et al (2019) Learning to predict autism spectrum disorder based on the visual patterns of eye-tracking scanpaths. In: HEALTHINF, pp 103–112
15. Satu MS, Sathi FF, Arifen MS, Ali MH, Moni MA (2019) Early detection of autism by extracting features: a case study in Bangladesh. In: 2019 International conference on robotics, electrical and signal processing techniques (ICREST), pp 400–405
16. Carette R, Elbattah M, Dequen G, Guérin J-L, Cilia F (2018) Visualization of eye-tracking patterns in autism spectrum disorder: method and dataset. In: 2018 Thirteenth international conference on digital information management (ICDIM), pp 248–253
17. Chollet F et al, Keras <https://github.com/fchollet/keras>
18. Elbattah M, Carette R, Dequen G, Guérin J-L, Cilia F (2019) Learning clusters in autism spectrum disorder: image-based clustering of eye-tracking scanpaths with deep autoencoder. In: 2019 41st Annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp 1417–1420
19. Elbattah M et al (2021) Variational autoencoder for image-based augmentation of eye-tracking data. *J Imaging* 7:83

## Chapter 21

# Real-Time Lightweight Bangla Sign Language Recognition Model Using Pre-trained MobileNetV2 and Conditional DCGAN



Abdullah Al Rafi, Rakibul Hassan, Md. Rabiul Islam,  
and Md. Nahiduzzaman

**Abstract** Sign language is one of the major communication mediums for deaf and dumb people. An interpreter can help disabled people in their daily life. Most of the works on Bangla sign language (BdSL) recognition are either on grayscale images or need heavy preprocessing. This paper proposes a lightweight, less preprocessing-based BdSL recognizing model with good accuracy. We used a dataset that has 38 classes. We used a pre-trained MobileNetV2 because of the small dataset, which is a lightweight model. To generalize the model, we used conditional deep convolutional generative adversarial network. We focused on creating a lightweight Bangla sign language recognizing model that smartphones can run and does not rely on preprocessing. We achieved 94.74% testing accuracy. Although our proposed model could not surpass the overall state-of-the-art accuracy, the model achieved better accuracy in classifying all individual signs, which was one of the major problems of many research works.

**Keywords** BdSL · Bangla sign language · MobileNetV2 · Conditional DCGAN · CNN · Computer vision

## 1 Introduction

Sign languages are just like natural languages that have their grammar and vocabulary. There is no universal sign language. Almost every country has its native sign language. It is difficult to tell precisely how many sign languages are there in the world. According to an article written by Santanu Dey in 2014 [1], there are almost

---

A. Al Rafi (✉) · R. Hassan · Md. Nahiduzzaman

Department of Electrical and Computer Engineering, Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh  
e-mail: [alrafikp@gmail.com](mailto:alrafikp@gmail.com)

Md. Rabiul Islam

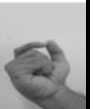
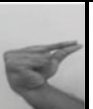
Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi 6204, Bangladesh

2.6 million deaf peoples in Bangladesh. Not only do deaf and dumb people use sign language, their parents and relatives also learn sign language to communicate with them.

However, outside a deaf or dumb person’s circle, there are hardly people who know sign language. So, one faces lots of difficulties in expressing himself. Hiring an interpreter is expensive and not always available. A sign language recognizer plays a vital role here. A vision-based sign recognizer can create a bridge between a deaf and dumb person and the world. Table 1 shows the BdSL alphabets.

Few research works have been carried out to develop Bangla sign language (BdSL) recognizer. Because of the lack of a publicly available dataset and many classes (9

Table 1 BdSL alphabets

| বাংলা বর্ণমালার সংকেত<br>(Bengali manual alphabet) |                                                                                   |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                      |                                                                                      |
|----------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| স্ব<br>র<br>ব<br>র্ণ                               |  |    |    |    |    |    |   |                                                                                      |                                                                                      |
|                                                    | অ/য়                                                                              | আ/া                                                                                 | ই/ঈ/ি/<br>া                                                                         | উ/ঊ/া/<br>া                                                                         | ঋ/ৠ/ড়/ঢ়/<br>ঢ়া                                                                   | এ/এ                                                                                 | ঐ/ঐ                                                                                 |                                                                                      |                                                                                      |
|                                                    |  |    |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                      |                                                                                      |
|                                                    | ও/ও                                                                               | ঔ/ঔ                                                                                 |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                      |                                                                                      |
|                                                    | ব<br>্য<br>ন<br>ব<br>র্ণ                                                          |   |   |   |   |   |   |   |                                                                                      |
|                                                    |                                                                                   | ক                                                                                   | খ/ক্ষ                                                                               | গ                                                                                   | ঘ                                                                                   | ঙ                                                                                   | চ                                                                                   | ছ                                                                                    |                                                                                      |
|                                                    |                                                                                   |  |  |  |  |  |  |  |                                                                                      |
|                                                    |                                                                                   | জ/য                                                                                 | ঝ                                                                                   | ঞ                                                                                   | ট                                                                                   | ঠ                                                                                   | ড                                                                                   | ণ/ন                                                                                  |                                                                                      |
|                                                    |                                                                                   |  |  |  |  |  |  |   |  |
|                                                    |                                                                                   | ত                                                                                   | থ                                                                                   | দ                                                                                   | ধ                                                                                   | প                                                                                   | ফ                                                                                   | ব/ভ                                                                                  | ম                                                                                    |
|                                                    |                                                                                   |  |  |  |  |  |  |                                                                                      |                                                                                      |
|                                                    |                                                                                   | ল                                                                                   | শ/ষ/স                                                                               | হ                                                                                   | ঐ                                                                                   | ঐ                                                                                   | ঐ                                                                                   |                                                                                      |                                                                                      |

for vowels, 29 for consonants) in Bangla sign language (BdSL), it is not easy to build a good classifier.

Most of the BdSL-related research works were focused on grayscale images, and small-sized datasets were used. Many of them used computationally costly preprocessing. Those approaches are suitable for use with a computer and Webcam. However, most people use their smartphones when they are outside. Those models are hard to implement on mobile devices to classify signs in real time.

If we want to use those heavy preprocessing-based methods in smartphones, we need to use cloud computational power to work. As we need to detect in real time, we have to stream the video to the cloud server, which needs heavy Internet usage. As Internet packages are costly in Bangladesh, we need mobile applications to detect sign language in real-time offline. Thus, in the proposed model, we have focused on the lightness of the model without reducing the classifier's performance.

This paper consists of five sections. Section 2 gives a literature review of the works done on BdSL recognition. Section 3 describes our proposed method. Section 4 focuses on experimental result evaluation. Section 5 gives the conclusion of this paper.

## 2 Literature Review

Many researchers have achieved remarkable results on sign language classification. However, here, we focused mainly on vision-based Bangla sign language recognition models.

Begum and Hasanuzzaman [2] used principal component analysis (PCA) in their paper, one of the earliest works on vision-based BdSL recognition. They built this model to classify 6 vowels and 10 numbers.

Deb et al. [3] used two wrist bands to find the hand region and then the template matching technique for classification. They achieved good accuracy but faced few problems with different backgrounds and hand sign rotation.

Several works used the support vector machine (SVM) combined with other techniques. Yasir et al. [4] used scale invariant feature transform (SIFT) for feature extraction and SVM for classification. Uddin and Chowdhury [5] applied the Gabor filter to extract features and SVM for classification. However, their accuracy dropped a little when they rotated the picture.

Some researchers used K-nearest neighbors (KNNs) in their papers. Rahaman et al. [6] recognized BdSL in real time by converting images to binary images and using the K-nearest neighbors (KNNs) classifier. Though they used a pretty good amount of data, they used images of only 10 persons. So, there was a scope of lack of variation. They had difficulties segmenting the hand area if the background had a color similar to skin and faced difficulty distinguishing some letters. Haque et al. [7] combined PCA and KNN for classification. They used 4 images of each class to train the model to classify 26 classes.

Neural network-based works are famous in vision-based BdSL recognition models. ChandraKarmokar et al. [8] used neural network ensemble (NNE) for recognition which they trained using the negative correlation learning (NCL) algorithm. Yasir and Khan [9] used linear discriminant analysis (LDA) and artificial neural network (ANN). Ahmed and Akhand [10] used fingertip positioning for BdSL classification. This model used the canny edge detection algorithm to detect each fingertip and passed the data to a three-layer ANN for classification.

Convolutional neural network (CNN) brought high accuracy in neural network-based works. Yasir et al. [11] used a virtual reality-based hand tracking controller known as the leap motion controller (LMC), convolutional neural network (CNN), and a multilayer neural network. Islam et al. [12] used preprocessing and convolutional neural network (CNN). However, we can see from the confusion matrix that the model gave low accuracy in some digits. For example, digit 9 got 52% accuracy. Hoque et al. [13] used faster R-CNN to classify BdSL in real time. Though they achieved great accuracy, they faced problem recognizing letters that had similar patterns.

As BdSL has many classes and most BdSL dataset sizes are small, pre-trained models became very popular. Rafi et al. [14] used a pre-trained VGG19 model and convolutional neural network, which does not rely on preprocessing. However, their model faced difficulties recognizing some digits. Nishat and Shopon [15] created an unsupervised pre-training and a transfer learning-based Bangla sign language recognition model. They proposed two models. The unsupervised pre-training model took an image, used autoencoder to get the most critical features, and passed them to a fully connected layer. In the transfer learning-based model, they used a pre-trained VGG16 model.

Researchers have done notable works on BdSL recognition. Many of the works are rely on preprocessing, grayscale images or reducing the image size. Most of the works used small datasets or data collected from very few people. As a result, some might have suffered minor variation, and models were not generalized enough. They performed less well in case of rotation or background with a color similar to skin color. One of the main problems they faced was that although they gained good overall performance, some of the models showed weak performance in classifying some classes. That is why we propose a model that works on RGB images without relying on preprocessing, grayscale images, or reducing the image size.

### 3 Proposed Method

The dataset we used in our research was initially small in size. To increase variation and make our model more generalized, we used conditional deep convolutional generative adversarial network (Conditional DCGAN) to generate images. We trained the conditional DCGAN to generate an image of any class label given. After that, we added 1900 generated images in the dataset with real images. We combined a pre-trained MobileNetV2 and few layers and trained the model after freezing the

**Fig. 1** Proposed method

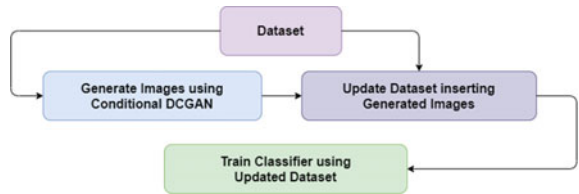


**Table 2** Dataset overview

| Gender |        | Hearing imparity |        | Age range |
|--------|--------|------------------|--------|-----------|
| Male   | Female | Deaf             | Normal | 12–30     |
| 193    | 143    | 42               | 298    |           |

MobileNetV2 using Adam optimizer. We then trained the model unfreezing all layers except the first 100 layers using Adam optimizer. We deleted the previously generated images of each class, generated new 380 images, added them to our dataset, and trained the model using Adam optimizer (Fig. 1).

3.1 Dataset

Most of the datasets on BdSL are not public, incomplete, or grayscale images. We used the dataset provided by Rafi et al. [14] on [kaggle.com](#), which was used in their paper [14]. There are 38 different signs and a total of 12,582 images in the dataset. The images were 224 × 224 px RGB images. Table 2 collected from the paper [14] describes the participants in data collection.

3.2 Choosing Parameters for Conditional DCGAN

Conditional deep convolutional generative adversarial network is a very famous tool for generating an image of a given label. As we have a relatively small dataset for image generation, we generated an image in 28 × 28 px gray scale and then converted it to a 224 × 224 px black and white image. Because of the dataset’s limitation, we could not produce 224 × 224 px RGB images as it resulted in Mode Collapse and generated the same image repeatedly. So, we chose the generated image size to be 28 × 28 px gray scale.

### 3.3 Conditional DCGAN Architecture

Generative model architecture is given in Fig. 2. The generative model input was a noise vector of size 100, and 1 labeled the class we want to generate. Then, we passed the noise to a fully connected layer of size 6272 with LeakyReLU activation. The mathematical expression of LeakyReLU is given in Eq. 1:

$$F(z) = \begin{cases} 0.01z, & z < 0 \\ z, & z \geq 0 \end{cases} \quad (1)$$

Then, we reshaped it to  $7 \times 7 \times 128$ . On the other hand, we passed the class label of size 1 to the embedding layer of size 50. Then, we passed it through a fully connected layer of size 49. Then, we reshaped it to  $7 \times 7 \times 1$  and stacked it on that  $7 \times 7 \times 128$ . Then, we had  $7 \times 7 \times 129$ . After that, we used transposed convolution with kernel size  $4 \times 4$  and strides (2, 2) and filter of 128, which we used for upsampling the input to  $14 \times 14 \times 128$ , followed by LeakyReLU activation. We repeated the same process. Thus, we got  $28 \times 28 \times 128$ . Then, we used a convolutional layer with filter 1, kernel size  $7 \times 7$ , and Tanh activation. The mathematical expression of Tanh is given in Eq. 2:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

After that, we had  $28 \times 28 \times 1$ , which was our desired  $28 \times 28$  px generated grayscale image.

Discriminator model architecture is given in Fig. 3. The input of the discriminator was a  $28 \times 28 \times 1$  image and the label of the image. We passed the label to the

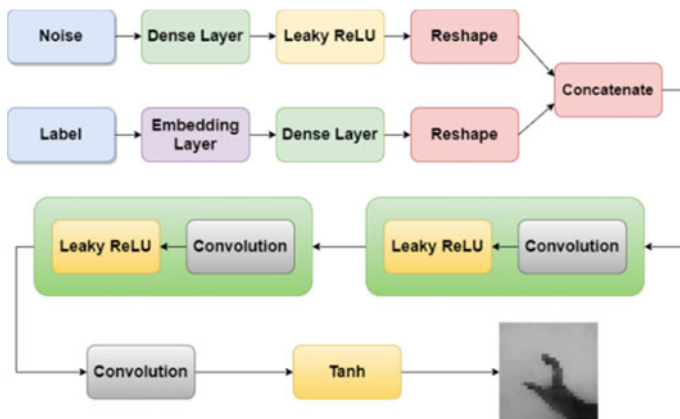


**Fig. 2** Generative model architecture

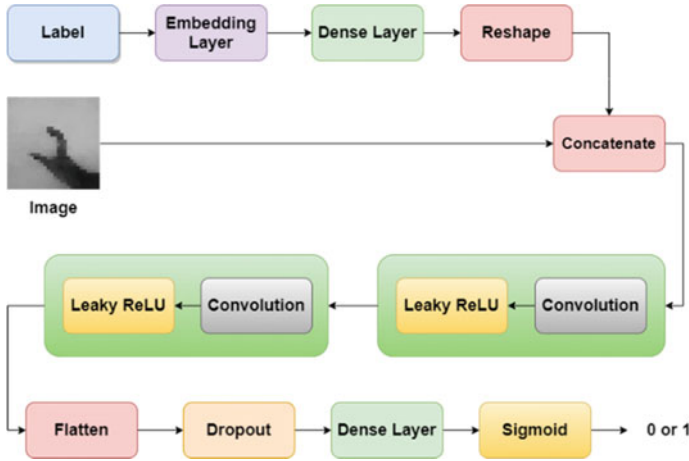


**Fig. 3** Discriminator model architecture

embedding layer, which converted the label to a vector of size 50. We passed it through a fully connected layer of size 784. Then, we reshaped it to  $28 \times 28 \times 1$  and stacked it on that image of size  $28 \times 28 \times 1$ . After the process, we had  $28 \times 28 \times 2$ . We used a convolutional layer with filter 128, kernel size  $3 \times 3$ , and strides 2, 2 with LeakyReLU activation. We used this technique for downsampling. Then, we had  $14 \times 14 \times 128$ . We repeated the same process. After that, we had  $7 \times 7 \times 128$ . We flattened the layer and added dropout 0.4. We added a fully connected layer of size 1 with sigmoid activation. The mathematical expression of sigmoid is given in Eq. 3:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

### 3.4 Choosing Parameters for Classification Model

For choosing parameters for our classification model, we took a small portion of the training dataset. We trained and validated the model using a different combination of numbers in dense layers and different activation functions. Below is the parameter of the model that we found worked best.



### 3.5 Classification Model

Classification model architecture is given in Fig. 4. We chose a pre-trained MobileNetV2 model trained on ImageNet. Where VGG19 is 549 MB with 143,667,240 parameters [16], MobileNetV2 is only 14 MB with 3,538,984 parameters [16]. We needed a light model for our purpose. After that, we added global average pooling and batch normalization. Then, we added a fully connected layer of size 512 with Glorot uniform initializer, also called Xavier uniform initializer, to be the kernel initializer. We then initialized bias with zeros. We added ReLU activation. After that, we added batch normalization. We added the same fully connected layer and batch normalization layer again. Lastly, we added a fully connected layer of 38 with random uniform kernel initializer and bias initialized with zeros and Softmax activation. The mathematical expression of Softmax is given in Eq. 4:

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (4)$$

We carefully avoided adding batch normalization and dropout together as authors of the paper [17] showed that combining these techniques often leads to worse performance.

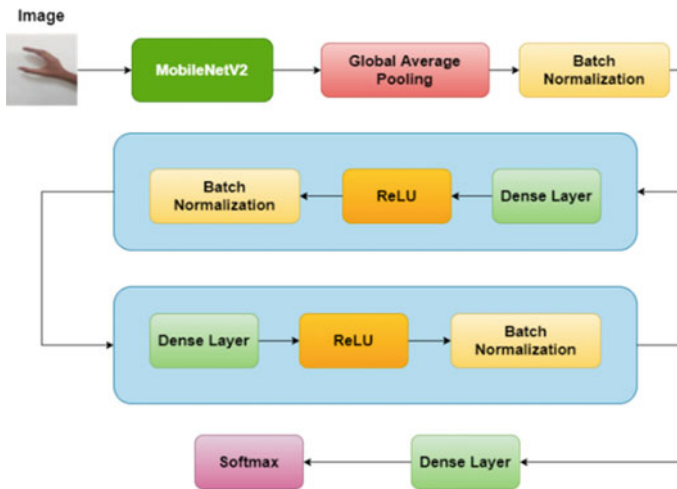


**Fig. 4** Classification model architecture

## 4 Experimental Evaluation

### 4.1 *Experimental Setup*

We implemented this model using Python 3.6.9 and TensorFlow 2.3.0. The hardware we used for this project was Intel(R) Xeon(R) CPU 2.00 GHz with 12 GB RAM and 7 GB NVIDIA Tesla P4.

### 4.2 *Splitting Dataset*

We took 10,302 images for training, 1140 images for validating, and 1140 images for testing.

### 4.3 *Dataset Augmentation for Classification Model*

We used different augmentation techniques for both offline and online augmentation.

We used Gaussian blur and Blocky artifact [18] techniques for offline and rotation, rescale, width shift, height shift, shear, zoom, channel shift for online augmentation.

### 4.4 *Training the Conditional DCGAN*

We trained the model for 1500 epochs. We took 64 real images, trained the discriminator, generated 64 fake images, and trained the discriminator. We froze the discriminator and connected the generator's output and label of input noise to the discriminator's input. We generated 128 fake images, passed them through the combined model and updated the generator, not the discriminator. We used Adam optimizer with a learning rate of 0.0002 and Beta1 0.5.

After completing these processes, we had a generator ready to generate an image by giving any label. A generated image of label 0 is given in Fig. 5.

### 4.5 *Training the Classification Model*

We froze MobileNetV2 and made layers after that trainable. We added 1900 generated images in the dataset with real images, used Adam optimizer with a learning rate of 0.001 and categorical cross-entropy for loss. For 20 epochs, we trained the model. The batch size was 32. When validation loss did not decrease for 5 epochs, we

**Fig. 5** Generated picture of label 0



decreased the learning rate by a factor of 0.1. If for 10 epochs, validation loss did not decrease, we stopped the training early. After the training, the model restored the parameters for which we got the lowest validation loss.

Then, we made all layers trainable except for the first 100 layers. We used Adam optimizer with a learning rate of 0.00001 and decay of 0.00001 and categorical cross-entropy. For 30 epochs, we trained the model. The batch size was 32. If the validation loss did not decrease for 7 epochs, we decreased the learning rate by a factor of 0.1. If for 10 epochs, validation loss did not decrease, we stopped the training early. After the training, the model restored the parameters for which we got the lowest validation loss.

We deleted the generated images, generated 380 images, and added to dataset. We used Adam optimizer with a learning rate of 0.00001 and decay of 0.00001. We used the decay rate as the learning rate/epoch and used categorical cross-entropy to calculate the loss. For 20 epochs, we trained the model. The batch size was 32. If the validation loss did not decrease for 5 epochs, we decreased the learning rate by a factor of 0.1. If for 8 epochs, validation loss did not decrease, we stopped the training early. After the training, the model restored the parameters for which we got the lowest validation loss.

## 5 Result Analysis

In each training, the model restored the parameters for which we got the lowest validation loss. So, we plotted the graph to that epoch where we got the lowest parameters.

As we can see in Fig. 6, we got 83.95% testing accuracy after the first training process. After the second training process, we got 94.47% testing accuracy which is shown in Fig. 7. After the last training, we can see in Fig. 8, got 94.74% testing accuracy.

We had a small amount of data to classify a high number of classes. Still, our model performed well without any preprocessing. It takes 7 ms to classify a sign. As shown in Table 3, we got 94.74% testing accuracy, where Rafi et al.'s [14] model got 89.6% accuracy. The model in Rafi et al.'s [14] paper trained on the same dataset as ours did not perform well when it tried to classify labels 3, 4, 5, and 9. It got 55%,

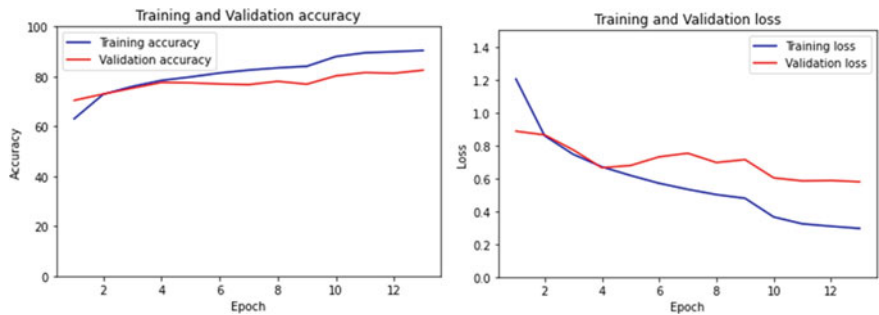


Fig. 6 Accuracy versus epoch and loss versus epoch graph during the first training

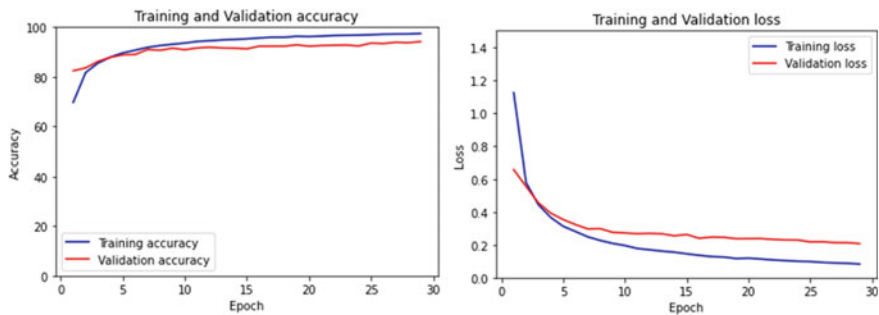


Fig. 7 Accuracy versus epoch and loss versus epoch graph during the second training

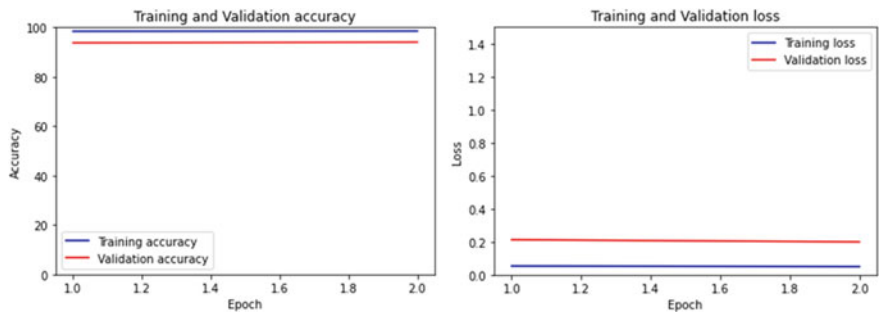


Fig. 8 Accuracy versus epoch and loss versus epoch graph during the third training

57%, 65%, and 68% accuracy in those cases. Our model overcame that problem. The worst accuracy in our model was 77% for label 32. Our model got better overall accuracy and is lighter and faster than them which is shown in Table 3.

**Table 3** Comparison with some models

| Researchers            | Method                                          | Classification time       | Model size        | Test accuracy (%) |
|------------------------|-------------------------------------------------|---------------------------|-------------------|-------------------|
| Rahaman et al. [6]     | KNN                                             | Real-time<br>93.55 ms     | –                 | 96.46             |
| Hoque et al. [13]      | Faster R-CNN                                    | Real-time<br>90.03 ms     | –                 | 98.20             |
| Rafi et al. [14]       | Pre-trained                                     | 30 ms                     | 549 MB [16]       | 89.6              |
| Nishat and Shopon [15] | Unsupervised pre-training and transfer learning | –                         | 528 MB [16]       | 95.86<br>97.87    |
| Proposed method        | Pre-trained MobileNetV2 and conditional DCGAN   | <b>Real-time<br/>7 ms</b> | <b>14 MB [16]</b> | 94.74             |

Bold indicates the best ones in those categories

## 6 Conclusion

In this research work, we have developed a lightweight model to classify BdSL in real time. It provides very good accuracy in all classes and uses very little memory. It can be used in mobile application or any other sector where less memory and computational power is available. It performed well with a background that was not in the training dataset, but the performance decreased a little if we use a background with many variations. A large dataset may make the model more generalized to overcome this problem. We made a model that only classifies character. However, in the future, works can be done to create a model to build words and complete a sentence (Fig. 9).

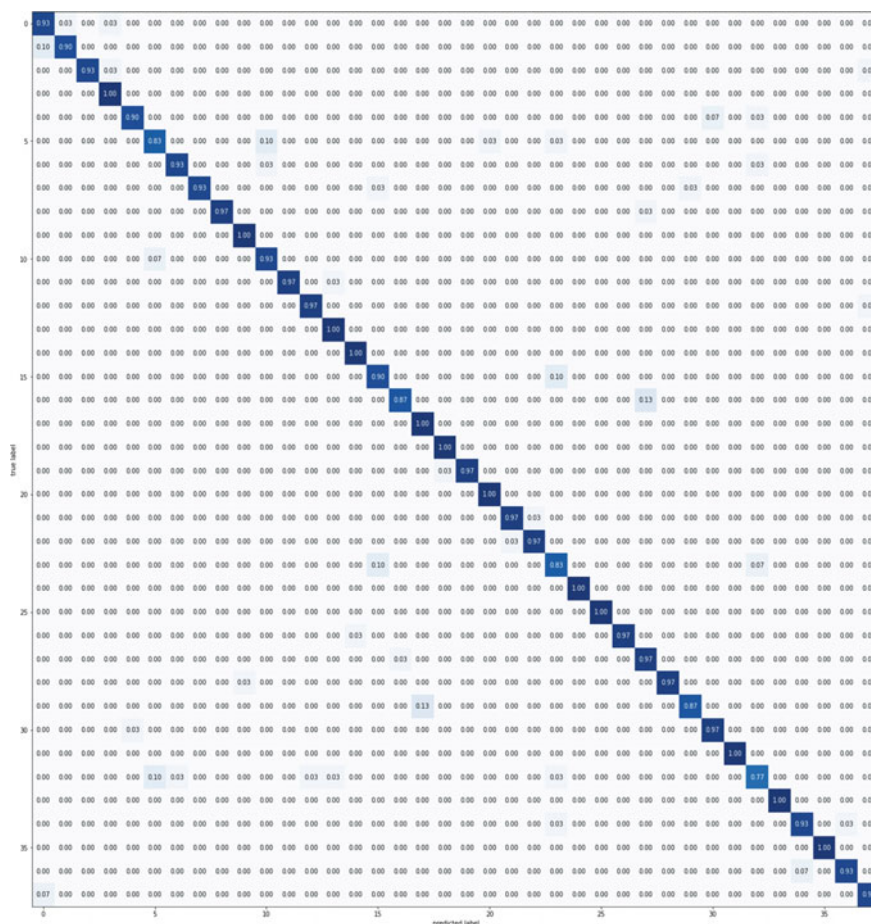


Fig. 9 Confusion matrix

## References

1. Dey S (2014) National activities—Bangladesh sign language day
2. Begum S, Hasanuzzaman Md (2009) Computer vision-based Bangladeshi sign language recognition system. In: 2009 12th international conference on computers and information technology. <http://doi.org/10.1109/iccit.2009.5407274>
3. Deb K, Khan MI, Mony HP, Chowdhury S (2012) Two-handed sign language recognition for bangla character using normalized cross correlation. Glob J Comput Sci Technol 12
4. Yasir F, Prasad PWC, Alsadoon A, Elchouemi A (2015) SIFT based approach on Bangla sign language recognition. In: 2015 IEEE 8th international workshop on computational intelligence and applications (IWCIA). <http://doi.org/10.1109/iwcia.2015.7449458>
5. Uddin MA, Chowdhury SA (2016) Hand sign language recognition for Bangla alphabet using support vector machine. In: 2016 international conference on innovations in science, engineering and technology (ICISSET). <http://doi.org/10.1109/iciset.2016.7856479>

6. Rahaman MA, Jasim M, Ali MdH, Hasanuzzaman Md (2014) Real-time computer vision based Bengali sign language recognition. In: 2014 17th international conference on computer and information technology (ICCIT). <http://doi.org/10.1109/iccitechn.2014.7073150>
7. Haque P, Das B, Kaspy NN (2019) Two-handed Bangla sign language recognition using principal component analysis (PCA) and KNN algorithm. In: 2019 international conference on electrical, computer and communication engineering (ECCE). <http://doi.org/10.1109/ecace.2019.8679185>
8. ChandraKarmokar B, Rokibul Alam KMd, Kibria Siddiquee Md (2012) Bangladeshi sign language recognition employing neural network ensemble. *Int J Comput Appl* 58:43–46. <http://doi.org/10.5120/9370-3846>
9. Yasir R, Khan RA (2014) Two-handed hand gesture recognition for Bangla sign language using LDA and ANN. In: The 8th international conference on software, knowledge, information management and applications (SKIMA 2014). <http://doi.org/10.1109/skima.2014.7083527>
10. Ahmed ST, Akhand MAH (2016) Bangladeshi sign language recognition using fingertip position. In: 2016 international conference on medical engineering, health informatics and technology (MediTec). <http://doi.org/10.1109/meditec.2016.7835364>
11. Yasir F, Prasad PWC, Alsadoon A, Elchouemi A, Sreedharan S (2017) Bangla sign language recognition using convolutional neural network. In: 2017 international conference on intelligent computing, instrumentation and control technologies (ICICICT). <http://doi.org/10.1109/iciciet1.2017.8342533>
12. Islam S, Sultana Sharmin Mousumi S, Rabby AKMSA, Hossain SA, Abujar S (2018) A potent model to recognize Bangla sign language digits using convolutional neural network. *Procedia Comput Sci* 143:611–618. <http://doi.org/10.1016/j.procs.2018.10.438>
13. Hoque OB, Jubair MI, Islam MS, Akash AF, Paulson AS (2019) Real time Bangladeshi sign language detection using faster R-CNN. <http://doi.org/10.1109/CIET.2018.8660780>
14. Rafi AM, Nawal N, Bayev NSN, Nima L, Shahnaz C, Fattah SA (2019) Image-based Bengali sign language alphabet recognition for deaf and dumb community. In: 2019 IEEE global humanitarian technology conference (GHTC). <http://doi.org/10.1109/ghc46095.2019.9033031>
15. Nishat ZK, Shopon Md (2020) Unsupervised pretraining and transfer learning-based Bangla sign language recognition. In: Proceedings of international joint conference on computational intelligence, pp 529–540. [http://doi.org/10.1007/978-981-15-3607-6\\_42](http://doi.org/10.1007/978-981-15-3607-6_42)
16. K. Team (2021) Keras documentation: Keras applications
17. Li X, Chen S, Hu X, Yang J (2019) Understanding the disharmony between dropout and batch normalization by variance shift. *Thecvf.com*, pp 2682–2690
18. Shopon M, Mohammed N, Abedin MA (2016) Bangla handwritten digit recognition using autoencoder and deep convolutional neural network. In: 2016 international workshop on computational intelligence (IWCI). <http://doi.org/10.1109/iwci.2016.7860340>

# Chapter 22

## A Novel Method of Thresholding for Brain Tumor Segmentation and Detection



Tanber Hasan Shemanto, Lubaba Binte Billah, and Md. Abrar Ibtesham

**Abstract** Human-assisted manual categorization of brain tumors is one of the most difficult tasks in medical image processing because incorrect prognosis and diagnosis might occur. Brain tumors vary in appearance, and there is a strong relationship between tumors and normal tissues. In this study, a thresholding approach was used to eliminate brain malignancies from 2D magnetic resonance brain images (MRI). We suggested an HOFILTER for preprocessing MRI images that eliminates unnecessary noise and prepares the picture for tumor segmentation. We also compared our findings to those of other researchers on segmentation and detection and discovered that ours were better for a variety of methodologies. We got better results after using the highlighted object filter (HOFILTER). To extract the appropriately segmented tumor from MRI images, edge detection, segmentation, and morphological techniques were applied. In our proposed model, we obtained an accuracy rate of 96.46% and a precision rate of 96.19%.

**Keywords** MRI · Thresholding · Segmentation · HOFILTER · Brain tumor detection

### 1 Introduction

The human brain is a very complex and sensitive organ. Every element of the human body is controlled by the brain which makes it a vital organ. The tumor of this organ is now a common occurrence which both children and adults are now at risk of developing. The most lethal and incurable disease is a brain tumor [1]. Tumors are solid masses of tissue that grow and replicate uncontrollably, ignoring normal cell regulating processes [2]. They can be both benign and malignant. Malignant

---

T. H. Shemanto (✉) · L. B. Billah  
Military Institute of Science and Technology (MIST), Mirpur Cantonment, Dhaka 1216,  
Bangladesh  
e-mail: [hasantanber519@gmail.com](mailto:hasantanber519@gmail.com)

Md. A. Ibtesham  
Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_22](https://doi.org/10.1007/978-981-19-7528-8_22)

277



tumors are cancerous. A benign tumor poses less risk. These appear gradually and are easily spotted. On the other side, malignant tumors are exceedingly aggressive and can be lethal [3]. Pre-malignant tumors are precancerous growths that can proceed to malignancy if left untreated [1]. Estimates for brain and spinal cord tumors in the United States by 2021 are based on data for adults and children. On average, there are 24,530 malignant tumors (13,840 males and 10,690 females) diagnosed. Inclusion of benign (non-cancerous) tumors would raise these numbers even higher. Approximately 18,600 people will die from brain and spinal cord tumors. There are some challenges involved in analyzing brain tumors due to differences in the shapes, sizes, and locations of each tumor as well as their presence and appearance in the brain.

Brain tumors are difficult to detect because accurate measurements are difficult to obtain. Brain tumors can be easily treated and cured if they are detected early. For clinical and scientific research purposes, medical imaging technology is being used to examine visual representations within the body. MRIs, CT scans, ultrasounds, SPECTs, PETs, and X-rays are among the noninvasive medical imaging technologies available. As it generates higher contrast images of the brain and malignant tissues than any other method, magnetic resonance imaging (MRI) is the most often utilized medical imaging tool [3]. As part of our research, we examined brain tumors utilizing MRI brain imaging. MRI is a common and effective method for identifying brain cancer. It is noninvasive and produces high-quality images of the body's organs. An MRI scan may reveal a great deal about the human brain [1]. Brain tumors can be identified using image processing methods. In medical image processing, image processing and image enhancement technologies are employed to improve image accuracy. Image processing is a broad field with a wide range of applications. To emphasize MRI image qualities, contrast modification and threshold techniques are applied. Edge detection, histogram, segmentation, and morphology are all significant in the diagnosis and classification of brain cancers [3].

## 2 Literature Review

Several previous and recent research related works conducted by others to segment the brain tumor process are presented in this sector. In order to understand the tumor segmentation process, some related works need to be understood, including fuzzy C means algorithms, geometric algebra, wavelet-based image fusion, watershed segmentation, and finally K means clustering. With the help of fuzzy logic, Ahmed et al. [4] present a customized algorithm for estimating intensity inhomogeneity, which can be used to support fuzzy MRI image segmentation. Their work implements a BCFCM algorithm, which converges to the correct classification faster, by changing the objective function used in the traditional FCM algorithm. Using the Gaussian multi-resolution EM algorithm, Tolba et al. [5] developed an MR brain image segmentation algorithm based on the EM algorithm and multi-resolution image analysis. Simulating the proposed algorithm shows that it performs better than traditional EM.

Based on Yu et al. [6], image segmentation is used for removing meaningful objects from an image. The authors suggest breaking an image down into three sections: black, gray, and white. In order to compute a fuzzy division of 2D histograms, Z-functions and s-functions are used. Their work is significant in that three-level image segmentation is adopted by following a maximum fuzzy partition of 2D histograms. According to Bayro Corrochano et al. [7], the use of boundary and texture information is imperative for the segmentation of medical images. In this paper, geometric algebra is used for sphere registration, real-time tracking, and volumetric data representation. The proposed method is of considerable relevance. It is difficult to detect and segment edges, segment images, and match images of long focal length optics, as Li et al. [8] demonstrate. Research has previously proposed wavelet-based image fusion as a technique for this mechanism. An enhanced wavelet function can be created by using a discrete wavelet frame transform (DWFT) and a support vector machine (SVM). Kumar and Mehta [9] point out that segmentation is incorrect when the edges of the tumor are not sharp, which is the case when a tumor is in its infancy. An algorithm based on texture is proposed in their paper. Automatic detection and segmentation of brain tumors is performed using this technique.

### 3 Methodology

#### 3.1 *HOFilter*

The majority of images have noise or distortion, which makes image processing difficult. It results in a decrease in resolution and quality. A filter reduces the amount of noise in a photograph. The ROI of a photograph can be increased or hidden with filters. Median filter, Gaussian filter, and box filter are some of the most prevalent types of filters. Thresholds are commonly used to identify tumors. Tumors in the input brain picture are brighter than the average threshold gray point. Because we know that an image may be represented as a two-dimensional function,  $f(x, y)$ . This filter is based on the gray threshold difference between input photographs and cancer. The  $X$  and  $Y$  numbers represent the picture's intensity level. Images are nothing more than a collection of little tiles. Image elements are abbreviated as pixels (Pix=picture and el=element). To conclude, a pixel is a unit of measurement that may be shown as the smallest unit of a digital image on a digital display device. HOFilter starts with the input picture and the gray threshold point of the image. Each pixel in the filter will be examined to ensure that it is greater than the threshold value previously determined. Let us consider the first pixel to be a major pixel to better understand the filtering process. If the primary pixel's value falls below the threshold, the filter will set the pixel to zero. If the primary pixel's value exceeds the threshold, the filter will advance to the next step. The value of the neighbor pixel of the primary pixel will be examined in the next phase. The primary pixel is surrounded by 8 pixels. HOFilter examines each one independently, as seen in Fig. 1.

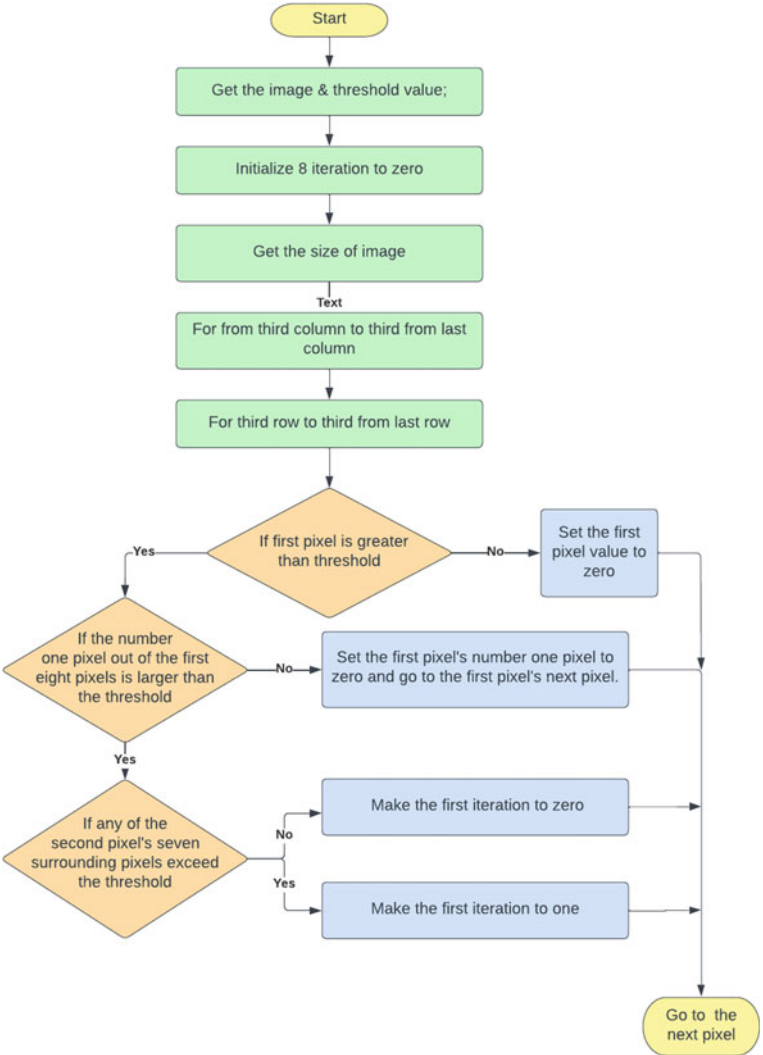


Fig. 1 HOFILTER methodology for the first pixel

3.2 Preprocessing

This is an important stage before picture processing. Noise reduction, picture type conversion, and image enhancement are all activities that must be done at this stage. A photograph may be appropriately processed once any unnecessary staff has been removed. The first step in image processing is the modification of the image [10]. Converting the raw image to grayscale is the most common approach for image

**Fig. 2** **a** Tumor affected brain MRI Image **b** Filtered image by HOFILTER

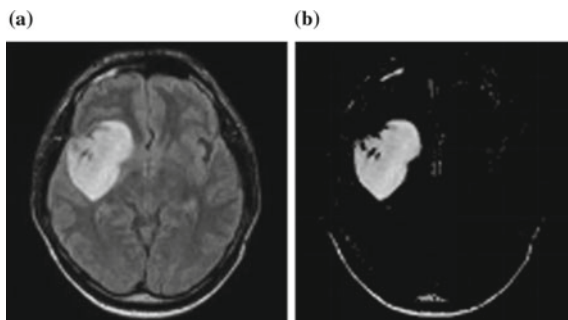
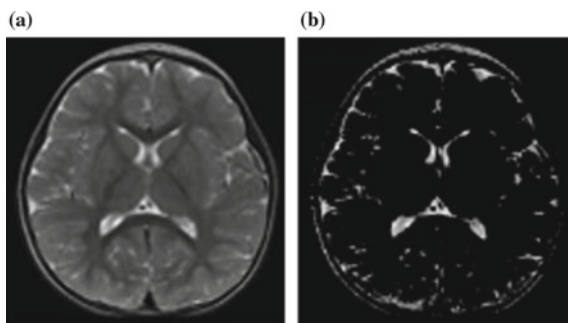


**Fig. 3** **a** Non-tumor affected brain MRI Image **b** Filtered image by HOFILTER



processing to this extent [11]. After converting the image to grayscale, it is necessary to decrease the noise. A number of filtering algorithms can be used to execute this noise reduction strategy. In our suggested model, we first converted the MRI image to gray scale. Following conversion, the picture was filtered five times with a median filter, which is the most common method of noise reduction. It is a filtering technique known as “non-linear”. It is simple to implement and has excellent results. This filter’s main benefit is its ability to effectively decrease salt and pepper noise. As a consequence, the image is enhanced, and noise is effectively reduced (Figs. 2 and 3).

### 3.3 Processing

The picture is sent to HOFILTER for additional processing after it has been suitably improved. However, before using HOFILTER, the threshold point must be defined. Threshold point is calculated using the graythresh function. After acquiring the threshold value, we double it by 255. Our analysis revealed that all photos failed to deliver an accurate result for only one specific threshold value. Raising the threshold became necessary. The rise, however, was not uniform across all threshold levels. Various increases were required at various threshold levels. Thus, we devised a series of ranges that could be added to the threshold value. In the first scenario, if the thresh-

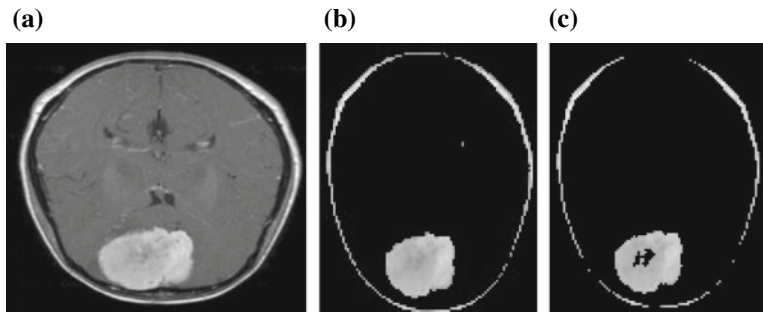


**Fig. 4** **a** Brain MRI image having threshold 54. **b** 70 added with threshold. **c** 90 added with threshold

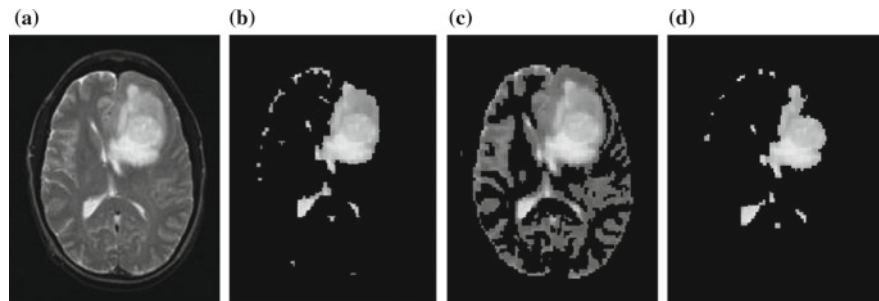


**Fig. 5** **a** Brain MRI image having threshold 63. **b** 80 added with threshold. **c** 40 added with threshold. **d** 100 added with threshold

old value is between 22 and 41, 60 must be added to the threshold, else the output image will be unacceptable. 70 needs to be added to the second example, where the threshold values vary from 40 to 55. If any value other than 70 is entered, some of the tumor pictures are lost, as demonstrated in the accompanying example (Fig. 4).

In the third scenario, photos with threshold values ranging from 55 to 110 and 80 should be included. According to our study, when an image's threshold value is less than 22, it comes within this third discipline. The lower threshold value indicates that the majority of the brain picture is below the threshold, with only the tumor above it. This third component specifies the tumor retrieval threshold, which must be less than 22. In order to obtain the tumor, a significant value must be added to the threshold.

The fourth and final example contains photos with a threshold value greater than 110. A threshold value of this type is required to add 30 with it. The threshold value in this situation is the smallest of the four parts. The pictures show threshold values more than 110, indicating that the intensities of brain images are high, resulting in not much difference in intensity between the tumor and other brain areas. Therefore, we need to add a low number to this threshold, which we set to 30 (Fig. 5).

Table 1 shows that the threshold value of the input picture is 135; thus, we must add 30 to it. Figure 6b illustrates that 30 has been added to the threshold value, making the output picture appropriate for additional image processing in tumor detection.

**Table 1** Adding values with threshold values for different ranges

| Ranges of threshold ( <i>t</i> )                 | Value to be added with threshold |
|--------------------------------------------------|----------------------------------|
| $t > 22 \ \& \ t < 41$                           | 60                               |
| $t > 40 \ \& \ t < 55$                           | 70                               |
| $t < 22 \ \text{or} \ t > = 55 \ \& \ t < = 110$ | 80                               |
| $t > 110$                                        | 30                               |

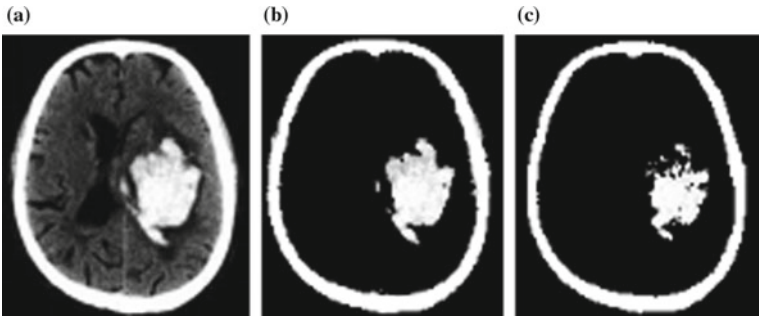


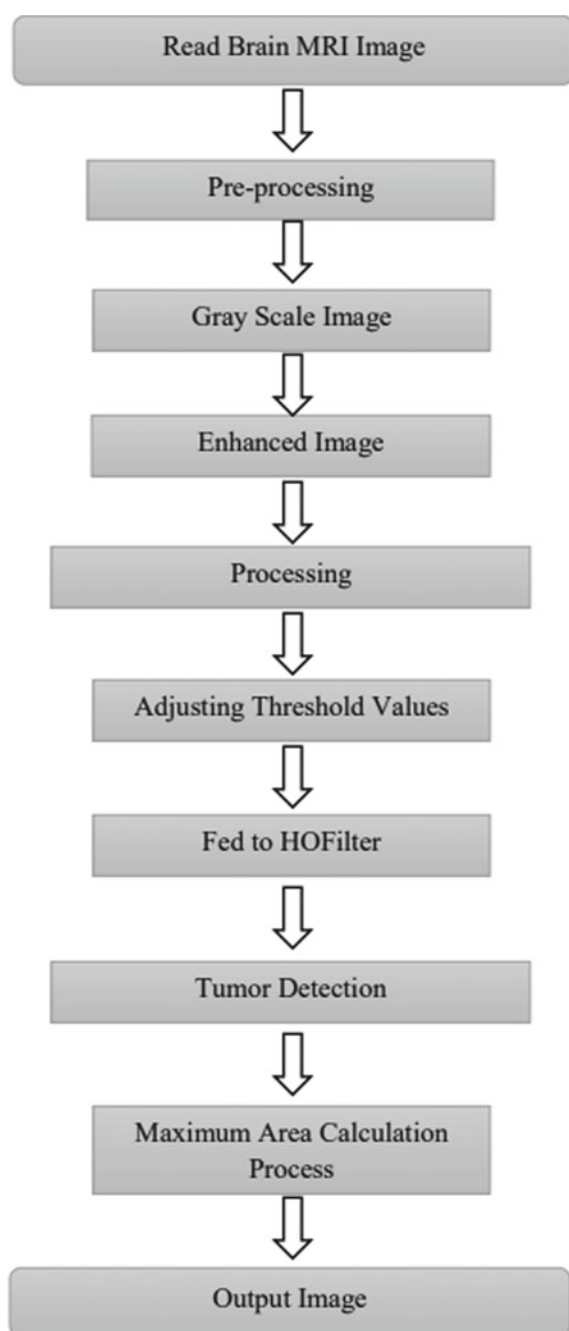
**Fig. 6** **a** Brain MRI image having threshold 135. **b** 30 added with threshold. **c** 90 added with threshold

Figure 6c increases the threshold by 90, and portion of the tumor is gone. As a result, 30 is the ideal value to add. If we keep Table 1, HOFILTER will eliminate unwanted areas of brain MRI pictures.

**3.4    Tumor Detection**

The generated picture must now be translated to binary format before the tumor can be detected by calculating the area of the elements in the image. To aid processing, we use the “clean border” function to remove the brain’s boundaries. To compute the area, we utilized the regionprops function. The function regionprops removes borders from a tagged picture. A binary image, such as “g,” has values of zero or one. Use the bwlabel function to label the image, which provides an ID number for each connected blob. This function may be used to count how many clusters and groupings of ones and zeros are deemed background data. There are various properties provided by the regionprops function, including area, bounding box, centroid, solidity, eccentricity, filled area, and filled image. Solidity, area, bounding box, and filled area are some of the ways used to identify tumors. Because tumor intensity is higher than total brain imaging intensity, the density of the tumor section should be higher. As a consequence, a dense zone should be more dense than the density value. The outcome will be bad if the value is less than 0.4. An image with a density higher than 0.4 may include a large number of elements. As a result, we must disentangle the tumor

**Fig. 7** Stages of tumor detection



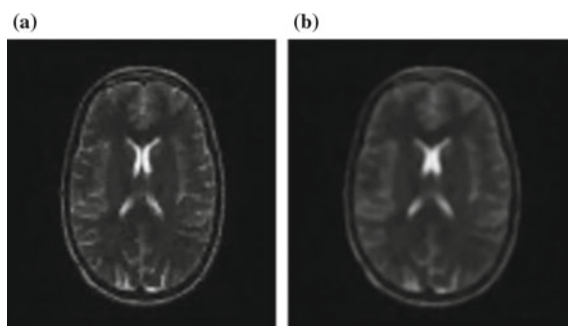


**Fig. 8** **a** Input brain MRI image. **b** Filtered image.

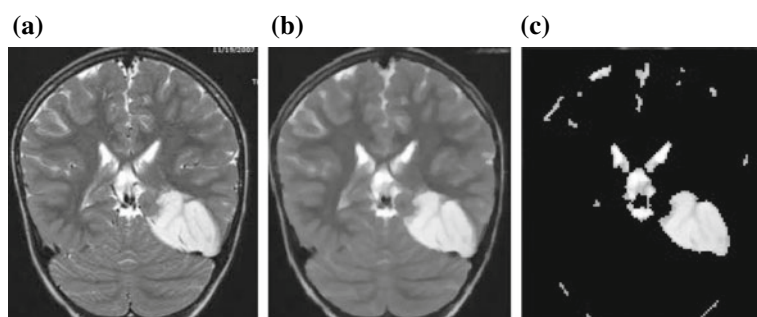


**Fig. 9** **a** Input brain MRI image **b** Filtered image **c** Output image of HOFILTER

from these objects. Tumor pixels, unlike pixels in other abnormalities, are constantly connected to one another. Other objects' pixels may have intensities larger than 0.4, but they are not continuous. After running HOFILTER, non-continuous pixel intensities that indicate significant variation between close pixels are removed. Certain objects, however, have densities larger than 0.4 and continuous pixel values. However, the size of these items is far smaller than that of tumors.

As a result, area calculations using regionprops may quickly differentiate malignancies. If an object has a high density area, it may be found using the 'Area' property. Tumors should be bigger than any other item in the filtered image, and our study found that things with areas smaller than 480–500 were not tumors. These objects are typically brain components. Tumors may therefore be defined as areas greater than 480–500 (Fig. 8).

### 3.5 Experimental Outcome

sec:exp Fig. 9a depicts an input brain MRI picture that is tumor affected. This image has been processed using the recommended methods. Figure 9b shows the filtered



**Table 2** Calculated parameters from theoretical simulation

| Accuracy | Precision | Recall | <i>F</i> -score | Specificity |
|----------|-----------|--------|-----------------|-------------|
| 96.46%   | 96.19%    | 97.12% | 96.65%          | 95.74%      |

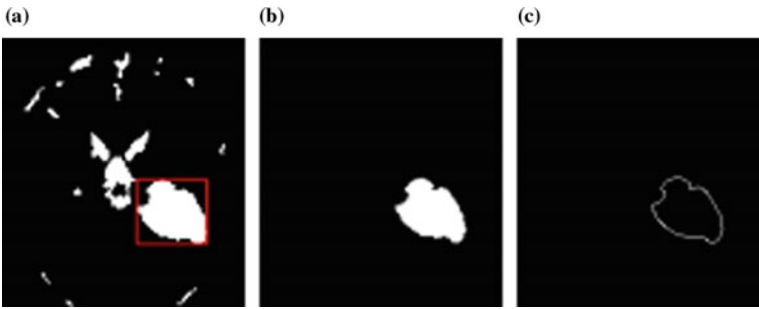
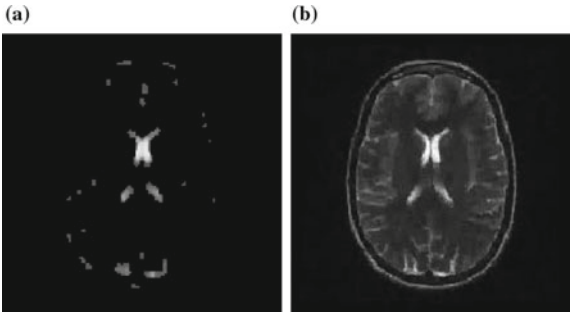


**Fig. 10** a Tumor location on input image. b Tumor alone. c Edge of tumor

**Fig. 11** a Output image of HOFILTER. b No tumor found in the image



input picture after it has been filtered using the median filter. The gray threshold value for this picture is 63. According to Table 3, 80 is added to this threshold, and then supplied to the HOFILTER, as seen in Fig. 9c (Table 2).

The tumor may be discovered by calculating the maximum area with the region-props function. Figure 10a depicts tumor localization using bounding boxes. Figure 10b depicts a newly discovered tumor, and Fig. 10c depicts the tumor's border.

This technology will now be used to a brain MRI that is not influenced by a tumor. Figure 8a depicts a typical brain MRI picture.

Figure 8b depicts a filtered image of the input image that was filtered using the median filter. The grayscale threshold for this image is 32. As seen in Table 3, 60 is added to this threshold before it, and the filtered image is sent into HOFILTER.

Figure 11b depicts the image following the maximum area computation method. This picture contains no tumors.

**Table 3** Existing related work

| References            | Number of pictures used | Methods                                                                           |
|-----------------------|-------------------------|-----------------------------------------------------------------------------------|
| Debnath et al. [13]   | 12 Images               | Detected by mean, median, standard deviation, and number of white pixels measured |
| Ihan et al. [14]      | 100 Images              | Binary thresholding                                                               |
| Mustaqeem et al. [15] | 100 Images              | Hybrid segmentation algorithm                                                     |
| Akram et al. [16]     | 100 Images              | Binary thresholding                                                               |
| Badran et al. [17]    | 102 Images              | Adaptive thresholding technique                                                   |
| Mittal et al. [18]    | Unspecified numbers     | Based on which the tumor was correctly located                                    |

## 4 Result and Discussion

### 4.1 Datasheet Acquisition

This study made use of Kaggle [12] MRI brain tumor segmentation datasets. Kaggle is a big online community of experts in information science and machine learning. This dataset comprises two datasets: one for tumorous MRIs (104 images) and one for non-tumorous MRIs (94 images).

### 4.2 Performance Comparison

In our suggested model, we employed 198 photos, both tumorous and non-tumorous, and we acquired an accuracy rate of 96.46%, precision of 96.19%, recall of 97.12%, *F*-score rate of 96.65%, and specificity rate of 95.74%. The existing works of other scholars that are similar are outlined in Table 3. These findings show that our method is capable of detecting brain from 2D MRI pictures.

## 5 Conclusion

A strategy was created in this study to assist medical practitioners in detecting brain tumors using MRI images. To filter the input brain MRI, we utilized Kaggle brain tumor datasets and typical filtering approaches as well as our newly developed filtering methodology. We also used both traditional filtering methods and our novel filtering strategy. A tumor was then detected utilizing the object's area properties. The suggested approach has a 97.12% identification rate on tumor photographs and a 95.74% recognition rate on non-tumor photos. Overall, at 96.46%, the accuracy rate

is higher than the industry average. Parallelization and the use of high-performance computing platforms are essential for best efficiency in the context of a comprehensive dataset. Despite our best efforts to identify tumors effectively, we uncovered situations where tumors could not be identified or were incorrectly diagnosed. As a result, we will try to work on both those images and the whole dataset.

## 6 Limitation and Future Work

Some future projects or research should be implemented as there are many fascinating scopes to work on in this project. Not all of the images in the incoming data are of excellent quality. Only two-dimensional images were used. We have not yet been able to identify the kind or size of tumor using our suggested approach. We will have greater opportunities for advancement in the future. The number of photos that may be used as input can be increased. More images will improve the model's performance. In the future, we want to work with 3D photos as well. Our study will focus on tumor size. As a result, small malignant tumors may be recognized. We wish to utilize our suggested HOFiler to detect and filter malignancies such as breast cancer. Standard classifiers may be used to compare the accuracy of our proposed model to that of other models. We shall try to determine if a tumor is benign or malignant after identifying it.

## References

1. Oo SZ, Khaing AS (2014) Brain tumor detection and segmentation using watershed segmentation and morphological operation. *Int J Res Eng Technol* 3(03):367–374
2. Patil RC, Bhalchandra AS (2012) Brain tumour extraction from mri images using matlab. *Int J Electron Commun Soft Comput Sci Eng (IJECSCE)* 2(1):1
3. Borole VY, Nimbhore SS, Kawthekar DS (2015) Image processing techniques for brain tumor detection: a review. *Int J Emerging Trends Technol Comput Sci (IJETTCS)* 4(2):1–14
4. Ahmed MN, Yamany SM, Mohamed N, Farag AA, Moriarty T (2002) A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Trans Med Imaging* 21(3):193–199
5. Tolba MF, Mostafa MG, Gharib TF, Salem MAM (2003) Mr-brain image segmentation using gaussian multiresolution analysis and the EM algorithm. In: *ICEIS* (2), pp 165–170
6. Yu HY, Fan JL (2008) Three-level image segmentation based on maximum fuzzy partition entropy of 2-d histogram and quantum genetic algorithm. In: *International conference on intelligent computing*. Springer, pp 484–493
7. Rivera-Rovelo J, Bayro E, Orozco-Aguirre R (2005) Medical image segmentation and the use of geometric algebras in medical applications. In: *Proceedings of the 10th Iberoamerican congress conference on progress in pattern recognition, image analysis and applications (CIARP'05)*
8. Li S, Kwok JTY, Tsang IWH, Wang Y (2004) Fusing images with different focuses using support vector machines. *IEEE Trans Neural Netw* 15(6):1555–1561
9. Kumar M, Mehta KK (2011) A texture based tumor detection and automatic segmentation using seeded region growing method. *Int J Comput Technol Appl* 2(4)

10. Gupta S, Hebli AP (2016) Brain tumor detection using image processing: a review. In: 65th IRF international conference
11. Kapoor L, Thakur S (2017) A survey on brain tumor detection using image processing techniques. In: 2017 7th international conference on cloud computing, data science and engineering-confluence. IEEE, pp 582–585
12. Your machine learning and data science community
13. Bhattacharyya D, Kim TH (2011) Brain tumor detection using mri image analysis. In: International conference on ubiquitous computing and multimedia applications. Springer, pp 307–314
14. Umit I, Ahmet I (2017) Brain tumor segmentation based on a new threshold approach. *Procedia Comput Sci* 120:580–587
15. Anam M, Ali J, Tehseen F (2012) An efficient brain tumor detection algorithm using watershed and thresholding based segmentation. *Int J Image Graph Signal Process* 4(10):34
16. Akram MU, Usman A (2011) Computer aided system for brain tumor detection and segmentation. In: International conference on computer networks and information technology. IEEE, pp 299–302
17. Badran EF, Mahmoud EG, Hamdy N (2010) An algorithm for detecting brain tumors in MRI images. In: The 2010 International conference on computer engineering and systems. IEEE, pp 368–373
18. Mittal K, Shekhar A, Singh P, Kumar M (2017) Brain tumour extraction using OTSU based threshold segmentation. *Int J Adv Res Comput Sci Softw Eng* 7(4)

# Chapter 23

## Developing Wearable Human–Computer Interfacing System Based on EMG and Gyro for Amputees



Md. Rokib Raihan and Mohiuddin Ahmad 

**Abstract** Electronics with low power consumption and the Internet of Things (IoT) are promoting the development of wearable medical devices. Persons who have amputated limbs by accident or by born and also want to operate practical devices and systems mostly personal computers then they have to be required a human–computer interface (HCI) system. As a result, innovations are essential for amputated persons in this area. To operate a computer, a keyboard and mouse are essential as control tools. Some may use the virtual keyboard to type or do multiple tasks, but that operation is quite slow or challenging for high-performance work mostly for amputees' person. In this study, we present a wireless EMG data acquisition system that has a multichannel data transmission capability and an EMG and gyro-controlled combined mouse and keyboard that is acceptable for persons who have had their hands amputated up to the elbow. Here, we use left-hand and right-hand biceps muscles for mouse clicking and keyboard button pressing functions. Hand coordination is detected by a gyroscope sensor, and this hand coordination data are used to mouse cursor positioning and selecting keywords or other functionality of the keyboard. In this paper, we design a signal acquisition system with a wireless multichannel transmission circuit and a keyboard that is operated with EMG and gyroscope signals. The device described in this article is capable of performing all of the functions of a standard computer mouse and keyboard.

**Keywords** Wearable · Computer mouse · sEMG · Human–computer interaction · Gyroscope · Amputee · EMG keyboard · Shift register · Decoder · NRF module · Microcontroller (uC)

---

Md. Rokib Raihan (✉)

Department of Biomedical Engineering (BME), Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh  
e-mail: [abdurrokbiraihan@gmail.com](mailto:abdurrokbiraihan@gmail.com)

M. Ahmad

Department of Electrical and Electronic Engineering (EEE), KUET, Khulna 9203, Bangladesh  
e-mail: [ahmad@eee.kuet.ac.bd](mailto:ahmad@eee.kuet.ac.bd)

# 1 Introduction

As a consequence of an accident or a severe infection, there are over a million people around the globe who have lost a limb, and these individuals lack the fundamental ability of the arm to operate any device or normal daily activity. To solve this problem, significant research is being conducted to establish the human-machine interface system for impaired individuals. As a result, different types of control system are developed in this area [1]. With the advancement of technology, the computer has recently become a vital tool in our society. These devices can be beneficial to the average individual (Fig. 1).

The handicapped, for example, who lack a forearm, cannot operate a computer without some assistance. Many researchers have created alternate interfaces to help people with upper limb disabilities use computers. In recent times, extracting user intentions from brain signals has gotten a lot of interest for the researcher since the signals convey information about body movements faster than other methods, and a variety of methods also have been developed to execute human thoughts [2]. Signals from brain activity can be used as a control signal at the central nervous system (CNS) level. To monitor brain activity non-invasively from the scalp, electroencephalogram (EEG) method is used [3], but signal extract by this method provides low spatial resolution and a low signal-to-noise ratio (SNR). Brain activity-based interfaces often need extensive training, and solving the technical difficulty is quite challenging [4, 5]. Electrooculography (EOG) has been investigated for use in the control of a computer [6, 7] or other vehicles for disabled persons. However, there are several disadvantages to using EOG instead of a computer keyboard. Some researchers also have developed device controllers using a combined signal from the brain, EMG, and EOG [8] or only combined EEG and EOG [9] or controllers for virtual keyboard by combined EOG and EMG signal [10].

Irrespective of the difficulties, we can use the electromyogram (EMG) signal as a control signal for computer operation. In comparison to neurological signals, EMG signals are easier and safer to monitor, and the SNR of this non-invasive monitoring technique is rather high. Even if a crippled individual is unable to utilize his or her hands, his or her residual muscle can often create an EMG signal. As a result, people

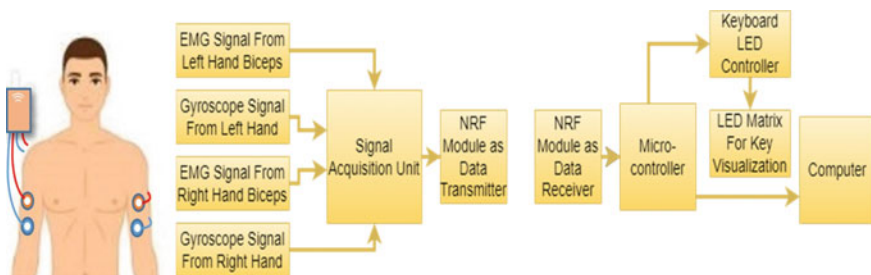


**Fig. 1** Overall block diagram

can utilize a computer if certain types of EMG generated from surviving muscle can be used as a determiner for computer instruction. The researcher developed an adaptive real-time HCI system to access a computer using residual muscle activity for some specific computer task. Some use different wrist movements to generate mouse and keyboard operating signals [2], and for visualization, they use an on-screen keyboard [2, 11]. An on-screen keyboard or virtual keyboard is a piece of software that mimics a regular keyboard. In this structure, the keyboard image is presented on a computer screen, and the users enter letters or text by pointing and clicking on the pictures of keys using the mouse cursor. People with substantial mobility problems can utilize computers with virtual keyboards. Some virtual keyboards use word prediction to help operators type faster. But, this is for that kind of user who can use a conventional mouse efficiently.

It also has some disadvantages the most of them is it occupies computer screen which can be on our working space. During our work, this type of keyboard needs to be repositioned on the screen which may cause slow down the working performance. For example, if we want to see the full-screen view of our computer screen, we need to minimize the on-screen keyboard, or if we want to type something related to previous data or text where the on-screen keyboard is positioned, then repeated repositioning, minimizing, reopening of the keyboard are required. This will hamper the working ability of the user. Since on-screen keyboard keys are very close to each other than EMG and gyro-controlled mouse efficiency may decrease due to the user's unusual spatial movement. So, the on-screen keyboard is not an effective development in a computer interfacing system. Since researchers have used different types of muscle activity to extract usable signals, then a complex algorithm is required to classify the signals [2, 12], and some of them also have used pattern recognition software to detect the muscle signal [13]. In some methods, researchers have used 12 channel electrodes [12] or 8 pairs of electrodes [13] to recognize the signal pattern and signal classification which is quite difficult for wearable devices. There is another issue regarding this vast number of electrode positioning on specific muscles for users, and it is quite difficult to maintain proper electrode placing.

To overcome those mentioned problems, we designed a combined EMG and gyroscope-controlled mouse and keyboard that functions like a computer keyboard and mouse. In this design, we only use biceps muscle activity to keep the design simple and user-friendly, and this activity will be used as a signal for a mouse click or keyboard button pressing action. For mouse cursor positioning or keyboard-letter indication, hand coordination is used which is detected by the gyroscope sensor.

The design contains:

- EMG signal detection
- EMG signal acquisition and signal processing
- Hand coordination tracking by gyroscope
- Wireless data transmission
- Visual keyboard.

## 2 System Design

### 2.1 EMG Signal Detection

An electrical signal that indicates a physical variable is known as a biomedical signal, which can be generated by any organ. In most cases, this signal may be described in terms of its amplitude, frequency, and phase, which are all time dependent. During muscle contraction, electrical currents in the muscle are created, which is a biological signal that reflects neuromuscular activity. The EMG signal's power spectral density is within 0–500 Hz. Because the signal strength is smaller than the electrical noise power outside of the 0–500 Hz frequency band, it cannot be used, and the frequency range of motion artifacts with electrical signals is 0–20 Hz [1]. Then, to extract a useable EMG signal, we use a band-pass filter with an instrumentation amplifier where we set low-pass filter and high-pass filter corner frequency by 18.55 Hz and 741 Hz, respectively, and gain 2468. Here, we use a part of the EMG detection circuit and its figure from the reference paper [1]. Figure 2 represents the EMG signal detection unit [1], and  $V_{\text{GND}}$  (Fig. 5) will be calculated by Eq. (1).

$$V_{\text{GND}} = \frac{V_S}{2}. \quad (1)$$

The second-order active high-pass filter and low-pass filter corner frequency ( $f_C$ ) can be set by Eq. (2) and the gain by Eq. (3) (Figs. 3 and 4).

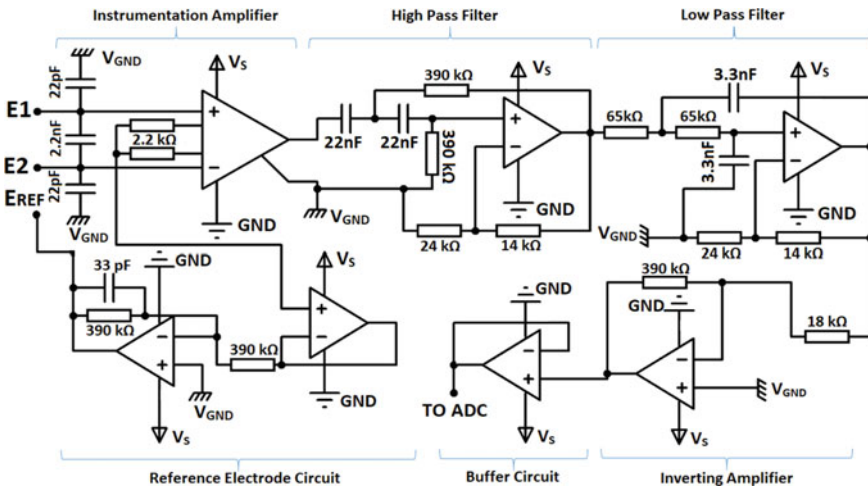


Fig. 2 EMG signal detection unit



**Fig. 3** **a** Active high-pass filter and **b** active low-pass filter

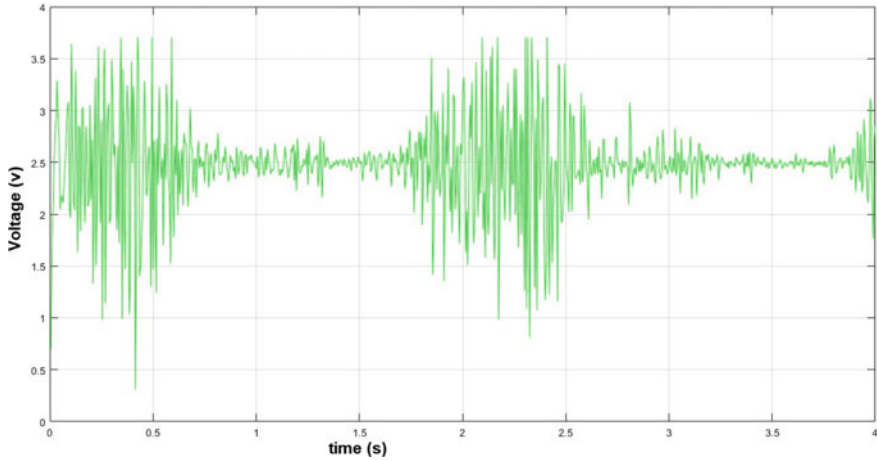
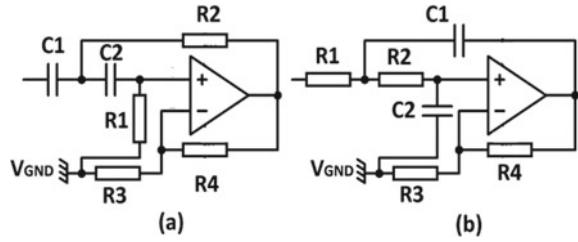


**Fig. 4** Raw EMG signal during muscle contraction

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (2)$$

$$G = 1 + \frac{R_4}{R_3} \quad (3)$$

## 2.2 Signal Acquisition Unit

For signal acquisition, we have used an 8-bit AVR microcontroller (Atmega-328p) (Fig. 5) which is good enough to signal acquisition of both analogs to digital conversion and processing gyroscope (MPU6050) signals. We have used an external 16 MHz crystal oscillator with the AVR microcontroller. Since researchers show that a 1 kHz sampling frequency is enough to detect biceps muscle contraction [1], then we use this frequency for EMG data sampling. Another reason to select this frequency is that we are using a microcontroller whose storage memory is low, but a minimum of

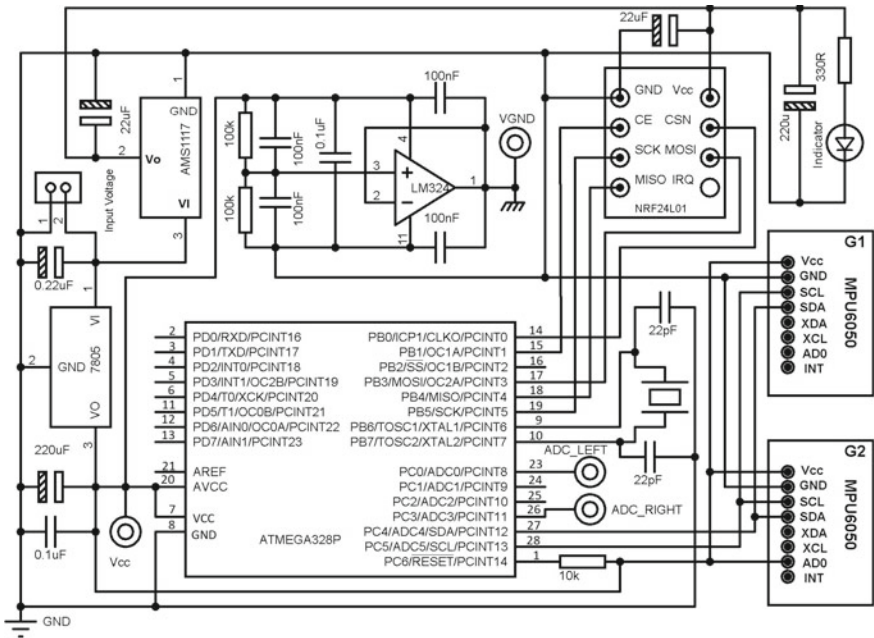


Fig. 5 Signal acquisition and transmission circuit

real-time data is need to be stored. But higher the sampling frequency, the higher the sampled data for the same specific time. In Fig. 5, we use AMS1117 IC as a 3.3-V voltage regulator because the nRF24L01 module’s maximum powering voltage is 3.6 V, and a 22  $\mu$ F capacitor is used at the output of the regulator for output voltage stability. A coupling capacitor (0.1  $\mu$ F) is used for noise compensation. In Fig. 5, we use two gyroscopes in parallel, and this module is based on the I2C communication protocol. In Figs. 2 and 5, all  $V_{cc}$  and  $V_s$  will be connected, and also, all  $V_{GND}$  will be connected (Fig. 6).

### 2.3 Hand Coordination Tracking by Gyroscope

A gyroscope is a device that measures or keeps track of orientation and angular velocity. In this paper, we used these properties to control the computer mouse and keyboard. In our experiment, two 3-axis gyroscopes are used which model is MPU 6050. Gyroscopes G1 and G2 (Fig. 5) are in parallel connected, but they have different data bus addresses since gyroscope (G2) AD0 pin is connected to the  $V_{cc}$ . G1 address is  $0 \times 68$ , and G2 address is  $0 \times 69$ . We used G1 and G2 for the left hand and right hand, respectively, to measure the orientation of our hand, and these will be our control signals to control the mouse and keyboard. After calibrating the maximum

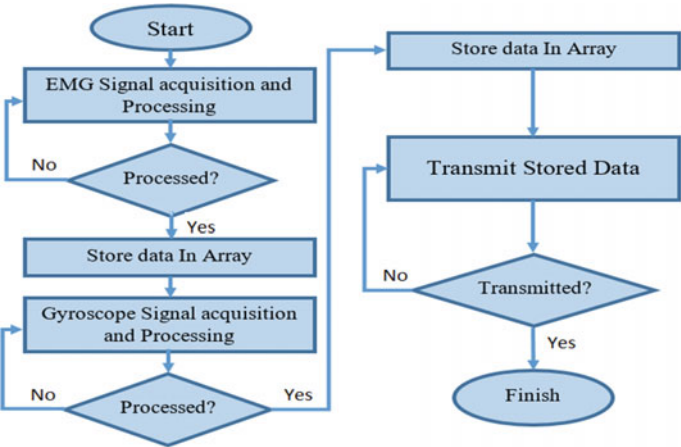


**Fig. 6** Flowchart of signal acquisition and transmission programming

and minimum values of our hand orientation, we map these values within a range from  $-6$  to  $6$  to control the mouse. To control the keyboard, we converted obtained  $Y$ -axis value into a range from  $1$  to  $6$  and the  $X$ -axis value into a range from  $1$  to  $8$  because our keyboard was parted by the  $(6 \times 8)$  matrix (Fig. 7). Our hand movement up-down will control the row, and left-right will control the column.

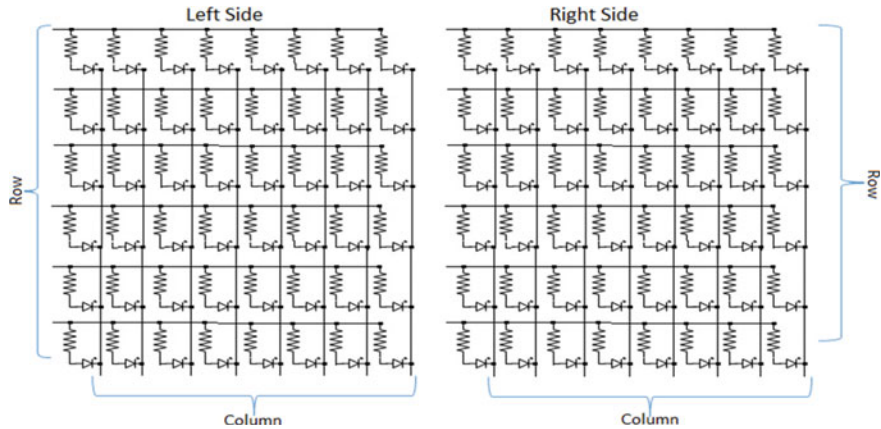


**Fig. 7** LED arrangement of the keyboard

**Table 1** Data stored in an array

| Type          | Mode | Left-hand data |        |     | Right-hand data |        |     |         |         |
|---------------|------|----------------|--------|-----|-----------------|--------|-----|---------|---------|
|               |      | G1             |        |     | G2              |        |     | G2      |         |
|               |      | X              | Y      | EMG | X               | Y      | EMG | X       | Y       |
| Array address | 0    | 1              | 2      | 3   | 4               | 5      | 6   | 7       | 8       |
| Value         | K/M  | 1 to 8         | 1 to 6 | 1/0 | 1 to 8          | 1 to 6 | 1/0 | −6 to 6 | −6 to 6 |

## 2.4 Wireless Data Transmission

For wireless data transmission, we use the nRF24L01 module. The nRF24L01 is a single-chip radio transceiver for the global ISM band of 2.4–2.5 GHz, and its data transmitting rate on-air is 1–2 Mbps. The power-up voltage for this module ranges from 1.9 to 3.6 V, and this module works through SPI communication. From Fig. 5, chip enabler (CE) if connected with HIGH, then it will transmit or receive data according to its selection mode; serial clock (SCK) synchronizes with SPI bus master, and MOSI and MISO are SPI input and output, respectively. We transmitted data as a data array through this module. The transmitted data and receiving data sequence and data array address are given in Table 1 where K = Keyboard, M = Mouse, EMG = 1 mean muscle contraction, and 0 means no contraction.

## 2.5 Visual Keyboard

We demonstrate a visual keyboard that contains all normal functionality but as a key as a light indicator to visualize letters or special characters. For user comfort, we divided the keyboard into two parts left side and the right side. The left side of the keyboard will be controlled with the right hand, and the right side of the keyboard will be controlled with the left hand. We have to do that because key detection has occurred by the spatial movement of the hand and pressing action will occur by muscle contraction. Here, we have to select the right-side key by the movement of the left hand, and to press the selected key, right-hand muscle contraction is required, and for the left-side key, it is vice versa.

To identify the key, we have to glow led. So, our keyboard has a large number of led matrices. In our designed keyboard, we have 6 rows and 16 columns of led to represent the keys, spatial characters, and other functionality. To increase user smoothness and user operating accuracy, we divided the all led by  $(6 \times 8)$  and  $(6 \times 8)$  matrices (Fig. 7). If we want to operate this LED only using microcontroller pins, then 22 microcontroller output pins are required. But, our microcontroller pins are limited, and some of the output pins have a special work function, so we can't use those pins for our design. To overcome this problem, we have designed a controller circuit that required only 4 microcontroller pins to operate the LED matrices. The designed circuit can work with upcoming EMG and gyro data from the user.

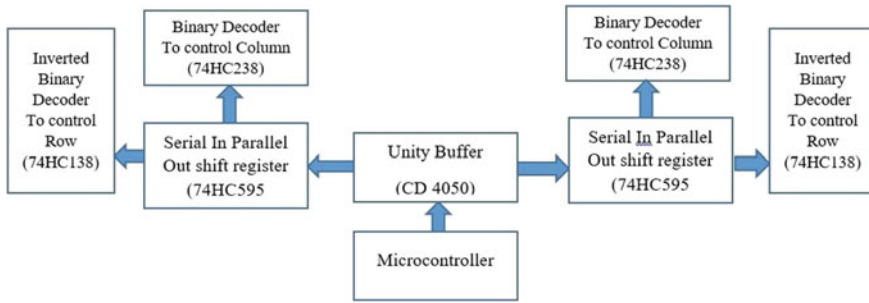


**Fig. 8** Block diagram of the LED controller

### Unity Buffer Circuit

The microcontroller's output pin can supply a very low current of about 20 mA. If we connect multiple inputs with a single microcontroller output pin, it will be unable to supply the necessary current to the other components of the circuit. Due to this limitation, system will be unstable. To overcome this, we use an operational amplifier as a voltage follower configuration which is known as a unity buffer or isolator circuit because of its properties. The OP-Amp has a very high input impedance so that they don't draw much current at the inputs, and it also has a very low output impedance. It is also a protection circuit for the microcontroller. Components connected with this circuit output can get a proper signal because its output draws current from the supply voltage of the IC. In our design, we use CD4050 OP-Amp IC. We chose this IC because it can operate from 3 to 15 V, and it has high source and sink current capability and also has a special input protection system (Fig. 8).

### Shift Register

A shift register is a form of a digital circuit that is formed by a combination of cascaded flip-flops where one flip-flop IC's output is connected to another flip-flop IC's input. Flip-flop IC takes input with pulse, and cascaded all flip-flops share the same clock, so data feed on one flip-flop to another. Using this property, we minimize more microcontroller output pin usage. In our design, we use 74HC595 8-bit shift register IC which is serial in parallel out. We chose this IC because its operating voltage is very low (3–5 V), and it is capable of high frequency in the MHz range. Through this IC, we can control 8 pins only using 3 pins of the microcontroller. Here, we use two-shift register IC to control the left side and the right side of the keyboard. Each shift register contains three parts: inputs, control pins, and output pins. Input pins are data pins; ST\_CP, SH\_CP, and control pins are OE, MR, and output pins are Q0 to Q7 (Fig. 9). To activate the chip, MR pin will be high which means it will connect with  $V_{cc}$ , and to see the output in parallel, OE pin will be connected with GND. In this configuration, the IC is ready to read and write data to its output. Now, we control this chip using a microcontroller through ST\_CP, SH\_CP, and DATA pins. 74HC595 IC read data when SH\_CP gets LOW to HIGH gate pulse.

Here are two options we can see instant reading data to the output or when we want to see it. If we connect SH\_CP and ST\_CP, then instant reading data will be shown in the output. But, in our design, we want to read 8-bit data first and then want to write to the output. So, we required two pins to control ST\_CP and SH\_CP. For this configuration, we read 8-bit data first by data pin for data, and pulse in SH\_CP pin is LOW to HIGH because it read data only positive edge of the pulse signal. During this reading process, the ST\_CP pin will be either zero or one. If we want to see the received data to the output, then we need to provide a HIGH to LOW pulse to the ST\_CP pin (Table 2).

Since our keyboard is divided into two parts left side and right side, then two-shift register IC is used to control both sides individually. So, we required 6 pins of the microcontroller to control the two-shift register. But, we use only 4 pins because shift register pulses are the same for both IC only data is different. For this reason, we connect our uC pin by buffer circuit to draw a minimum current from uC and

**Fig. 9** Pin-out of 74HC595 IC

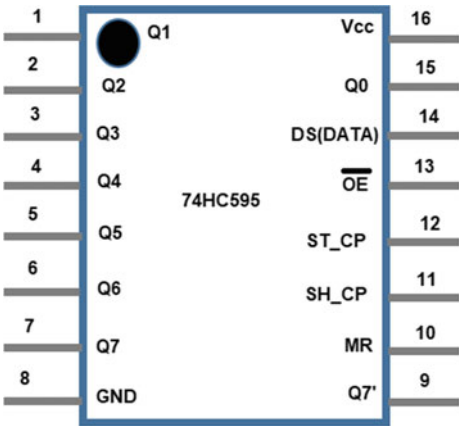


**Table 2** Important functions of 74HC595 IC

| IC pin number | PIN name | Function                   | Action                                                                              |
|---------------|----------|----------------------------|-------------------------------------------------------------------------------------|
| 10            | MR       | IC enabler                 | Keep high                                                                           |
| 11            | SH_CP    | Data reading pulse         |  |
| 12            | ST_CP    | Serial to parallel writing |  |
| 13            | OE       | Output enabler             | Keep low                                                                            |

system stability. In this configuration, we can control 16 individual pins using only 4 uC pins.

### Binary Decoder

Since in our designed keyboard, we need to control the  $(6 \times 8)$  led matrix on the left side and right side; then, we need 14 individual output pins for each side, but from the shift register, we can only get 8 pins to control. To overcome this problem, we use 4 bit binary to BCD decoder. Here, we use two decoders; one of them is inverted because to glow a led anode, and cathode is required. The binary input will get from the shift register output. These two-shift registers work like indexing the key. In our design, we use binary decoder IC 74HC238 and inverted binary decoder IC 74HC138. We select these ICs because it is CMOS low-power dissipation and has a wide supply voltage range from 2 to 6 V. The 74HC238 and 74HC138 have 3 input controller pins (E1, E2, and E3) (Fig. 10a, b) which can be used as a data protector. Address input pin will be enabled if E1 and E2 are connected to GND and E3 in  $V_{cc}$ . Without this configuration in enabler pins (E1, E2, E3), output pins are remaining LOW for 74HC238 and HIGH for 74HC138 whatever the inputs are. In (Fig. 10a, b), A0, A1, and A2 are input address pins, and Y0 to Y7 are output pins where A0 is LSB and A2 is MSB. For 74HC238 IC, if we connect  $A0 = 0$ ,  $A1 = 1$ , and  $A2 = 0$ , then output Y1 will be HIGH which means binary input will be converted into decimal output. For 74HC138 IC, if we connect  $A0 = 0$ ,  $A1 = 1$ , and  $A2 = 0$ , then output Y1 will be LOW, and the rest of the output pins will be high since it is an inverted decoder.

### LED Controller Circuit

On the receiver side, we used Arduino micro to control the LED matrix, and Arduino micro has a USB port to function keyboard and mouse. In Fig. 11, ST\_CP, SH\_CP, DS\_Right, and DS\_Left are connected with microcontroller digital output pins.

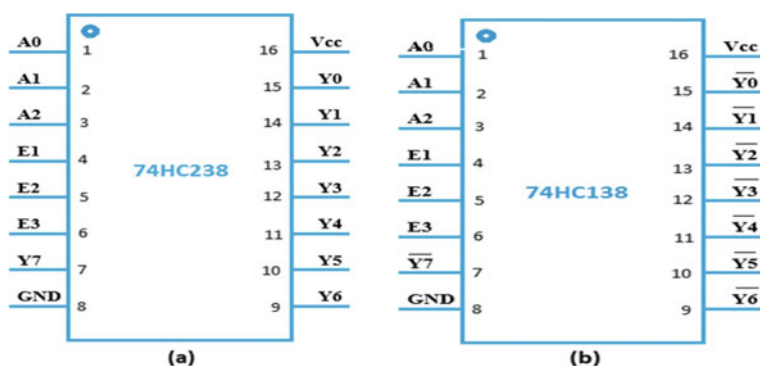


**Fig. 10** a Pin-out of 74HC238 IC and b pin-out of 74HC138 IC

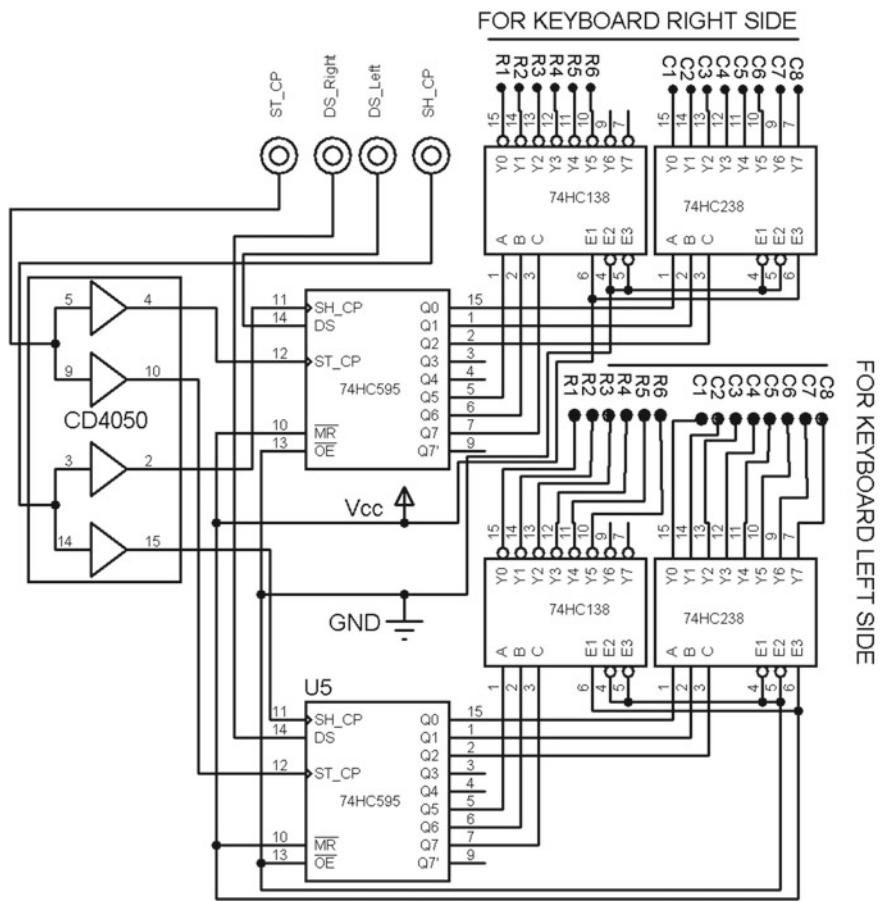


Fig. 11 Led controller circuit

3 Functionality

Our designed keyboard can be used either mouse or keyboard by muscle command. If we contract both hands' biceps muscles at the same time, then its functionality will be changed from keyboard to mouse or mouse to keyboard. When it will work as a mouse, then right-hand coordination will control the mouse cursor, and left-hand bicep muscle contraction will be worked as a click command. For keyboard interfacing, when we will want to access the keyboard left-side keys, then left-hand coordination will be used to visualize the keyword, and right-hand muscle contraction will take action to press the key. For keyboard right-side key accessing, the previous action will be reversed.



## 4 Conclusion

In this paper, we demonstrate an EMG-controlled combined keyboard and mouse suitable for people with hand amputation up to their elbow in which the device will work on a combination of signals from gyroscope and EMG. The algorithm used in this system is simpler and easy to implement, and the developed device completes the task like a conventional mouse and keyboard. As a result, we concluded that the described system is more straightforward to implement for real-time human–computer interaction (HCI) applications.

## References

1. Raihan MR, Shams AB, Ahmad M (2020) Wearable multifunctional computer mouse based on EMG and gyro for amputees. In: 2020 2nd international conference on advanced information and communication technology (ICAICT), pp 129–134. <http://doi.org/10.1109/ICAICT51780.2020.9333476>
2. Choi C, Kim J (2007) A real-time EMG-based assistive computer interface for the upper limb disabled. In: 2007 IEEE 10th international conference on rehabilitation robotics, pp 459–462. <http://doi.org/10.1109/ICORR.2007.4428465>
3. Millan JR, Renkens F, Mourino J, Gerstner W (2004) Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Trans Biomed Eng* 51(6):1026–1033. <https://doi.org/10.1109/TBME.2004.827086>
4. Nicoletis MAL (2001) Action from thoughts. *Nature* 409:403–407. <https://doi.org/10.1038/35053191>
5. Cheng M, Gao X, Gao S, Xu D (2002) Design and implementation of a brain-computer interface with high transfer rates. *IEEE Trans Biomed Eng* 49(10):1181–1186. <https://doi.org/10.1109/TBME.2002.803536>
6. Borghetti D, Bruni A, Fabbrini M, Murri L, Sartucci F (2007) A low-cost interface for control of computer functions by means of eye movements. *Comput Biol Med* 37(12):1765–1770. <https://doi.org/10.1016/j.compbiomed.2007.05.003>
7. Usakli AB, Gurkan S (2010) Design of a novel efficient human-computer interface: an electrooculogram based virtual keyboard. *IEEE Trans Instrum Meas* 59(8):2099–2108. <https://doi.org/10.1109/TIM.2009.2030923>
8. Fatourehchi M, Bashashati A, Ward RK, Birch GE (2007) EMG and EOG artifacts in brain-computer interface systems: a survey. *Clin Neurophysiol* 118(3):480–494. <http://doi.org/10.1016/j.clinph.2006.10.019>
9. Hosni SM, Shedeed HA, Mabrouk MS et al (2019) EEG-EOG based virtual keyboard: toward hybrid brain-computer interface. *Neuroinform* 17:323–341. <https://doi.org/10.1007/s12021-018-9402-0>
10. Dhillon HS, Singla R, Rekhi NS, Jha R (2009) EOG and EMG based virtual keyboard: a brain-computer interface. In: 2009 2nd IEEE international conference on computer science and information technology, pp 259–262. <http://doi.org/10.1109/ICCSIT.2009.5234951>
11. Merino M, Gómez I, Molina AJ, Guzman K. Assessment of biosignals for managing a virtual keyboard. Part of the lecture notes in computer science book series, vol 7383. [http://doi.org/10.1007/978-3-642-31534-3\\_50](http://doi.org/10.1007/978-3-642-31534-3_50)
12. Ando K, Nagata K, Kitagawa D, Shibata N, Yamada M, Magatani K (2006) Development of the input equipment for a computer using surface EMG. In: 2006 international conference of the IEEE engineering in medicine and biology society, pp 1331–1334. <http://doi.org/10.1109/IEMBS.2006.259222>

13. Wheeler KR (2003) Device control using gestures sensed from EMG. In: Conference: soft computing in industrial applications, 2003. SMCia/03. Proceedings of the 2003 IEEE international workshop. <http://doi.org/10.1109/SMCIA.2003.1231338>

# Chapter 24

## Implementation of Twisted Edwards Curve-Based Unified Group Operation for ECC Processor



Asif Faisal Khan and Md. Jahirul Islam

**Abstract** Exchanging information throughout the online or virtual media is one of the rapidly growing issues. Importantly, there must have security concerns for exchanging the information and thanks to cryptography to handle the concerned points. The group operation along with the elliptic curve cryptography (ECC) processor ensures the desired security. This work focuses on the newly invented twisted Edwards curve for executing group operations to enable enhanced security. Two novel architectures, referred to as first and second architectures of unified group operation, are designed as 256-bit over the prime field. On the Virtex-7 field-programmable gate array, the proposed architectures are realized. Interestingly, the second design reduces 128 clock cycles than the first one and also compared to other available designs in the literature. From the best of the author's knowledge, the designed architectures provide better performance with respect to throughput and slice area than the other related work, and thus, it is realizable as an effective design for the ECC processor on FPGA.

**Keywords** Twisted Edwards curve · ECC processor · Field-programmable gate array · Unified group operation

## 1 Introduction

In recent times, there is a lot of information around the world that needs security. Therefore, there is a demand for a secured network that can be applied to related appliances [1]. In addition, here comes the Internet of Things (IoTs) which in other words, refers to the global network. The Internet connects billions of electrical gadgets, and they share extensive amount of data [2]. The facts that inspire researchers to create lightweight cryptographic algorithms due to the necessity of data security and management in the cloud computing environment.

---

A. Faisal Khan · Md. Jahirul Islam (✉)  
Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh  
e-mail: [jahirul@eee.kuet.ac.bd](mailto:jahirul@eee.kuet.ac.bd)

In recent days, elliptic curve cryptography (ECC) becomes very much promising for security and in public domain applications. Notably, using fewer keys, ECC becomes more powerful compared to the traditional Rivest-Shamir-Adleman (RSA) algorithm and digital signature algorithm [2]. ECC requires lower hardware resources as well as low usage of energy [1].

ECC processor can also be implemented on FPGA to facilitate the easy update of the cryptographic algorithm. It is noted that, the FPGA implementation is better for prototype design due to their reduction of hardware cost, time, and test of development and most importantly eliminate the extra fabrication cost [3].

On the other hand, Edwards curves are a type of elliptic curve that is gaining popularity among security researchers due to their ease of use and strong resistance to side-channel attacks [2]. In addition, it is much faster and more secure than other techniques in the literature. Due to its unified point addition, the secret key is unrecognizable from the power tracing of the simple power analysis attack. Currently reported, a digital signature scheme called Ed25519 is a high-speed, high-security scheme that uses the Edwards25519 curve [2]. The twisted Edwards curve has less operation than any other curve among other Edwards curves [4]. These curves are at the heart of an electronics signature scheme called EdDSA [5]. It offers high performance and is faster than any other signature algorithm and avoids security problem. The core operation for ECC processor based on twisted Edwards curve is unified group operation, which includes both point doubling and point addition.

In this work, we demonstrate a unified group operation of the Twisted Edwards Curve algorithm with its efficient hardware architecture which requires fewer components. Two algorithms of the unified group operation are tested, and the second one utilizes fewer resources area, less computational time, and yield high throughput. The main challenge in this work is to compel the data speed efficient and reduce the area for compactness. If the data speed becomes high, quicker encryption and decryption will be fast enough for detection from spying.

## 2 Mathematical Background

### 2.1 Twisted Edwards Curve

The twisted Edwards curve is a version that is derived from the Edwards curve [4]. A non-binary field with a twisted Edwards curve in which  $\text{char}(k) \neq 2$  with fix prime field  $k$  and non-zero element  $a, d \in k$  can be written as:

$$E_{E,a,d} = ax^2 + y^2 = 1 + dx^2y^2 \quad (1)$$

If  $a = 1$ , that is untwisted because the curve turned into an ordinary Edwards curve [4]. Every twisted Edwards curve has a birational equivalent in Montgomery form to an elliptic curve [4].

The twisted Edwards curve has a unified point addition law that is combined point addition and point doubling which is good for identity. Another feature noted that it saves computational time by reducing arithmetic operation [1].

Equation (1) is in affine coordinates. For  $Z_1 \neq 0$ , the homogeneous point is  $(X_1 : Y_1 : Z_1)$ . Projection of that point in the affine point  $(X_1/Z_1, Y_1/Z_1)$  over  $E_{E,a,d}$  [4]. This turns into projective coordinates:

$$(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2 \quad (1)$$

The neutral point on twisted Edwards curve is  $(0, 1)$ , and an inverse point of  $(x, y)$  is  $(-x, y)$ .

Curve point description can be represented in different forms such as projective coordinates, Jacobian coordinates, affine coordinates, and Chudnovsky coordinates. In this work, we chose projective coordinates as it eliminates the inversion or division operation on Edwards curves [1].

## 2.2 Unified Group Operation Theory

The unified point addition formula including point addition and point doubling can be represented by the following equation [2]:

$$P_1(X_1, Y_1, Z_1) + P_2(X_2, Y_2, Z_2) = P_3(X_3, Y_3, Z_3) \quad (2)$$

where the parameters  $X_3, Y_3, Z_3$  are defined as [1]:

$$X_3 = Z_1 Z_2 (X_1 Y_2 + X_2 Y_1) (Z_1^2 Z_2^2 - d X_1 X_2 Z_1 Z_2)$$

$$Y_3 = Z_1 Z_2 (Y_1 Y_2 - a X_1 X_2) (Z_1^2 Z_2^2 + d X_1 X_2 Z_1 Z_2)$$

$$Z_3 = (Z_1^2 Z_2^2 + d X_1 X_2 Z_1 Z_2) (Z_1^2 Z_2^2 - d X_1 X_2 Z_1 Z_2)$$

## 3 Algorithms

The unified group operation for twisted Edwards curve in projective coordinates is given by the following algorithms:

**Algorithm 1** First unified group operation on twisted Edwards curve

---

Input:  $P_1(X_1, Y_1, Z_1)$  and  $P_2(X_2, Y_2, Z_2) \in k$   
Output:  $P_1 + P_2 = P_3(X_3, Y_3, Z_3) \in k$

---

$A = Z_1 \cdot Z_2, C = X_1 \cdot X_2, K = Y_1 \cdot Y_2, A_1 = X_1 + Y_1, A_2 = X_2 + Y_2,$   
 $B = A^2, E_1 = C \cdot K, C_1 = C + K, C_2 = a \cdot C, H = A_1 \cdot A_2,$   
 $E = d \cdot E_1, J = K - C_2, K_1 = H - C_1,$   
 $F = B - E, G = B + E, M = A \cdot K_1, N = A \cdot J,$   
 $X_3 = M \cdot F, Y_3 = G \cdot N, Z_3 = G \cdot F$

---

In the second algorithm,  $a = -1$  is selected to reduce arithmetic operations.

**Algorithm 2** Second unified group operation on twisted Edwards curve

---

Input:  $P_1(X_1, Y_1, Z_1)$  and  $P_2(X_2, Y_2, Z_2) \in k$   
Output:  $P_1 + P_2 = P_3(X_3, Y_3, Z_3) \in k$

---

$A = X_1 \cdot X_2, B = Y_1 \cdot Y_2, C = Z_1 \cdot Z_2, A_1 = X_1 \cdot Y_2, B_1 = X_2 \cdot Y_1,$   
 $AB_1 = A + B, AB = A \cdot B, C_1 = C^2, AB_2 = A_1 + B_1,$   
 $D_1 = d \cdot AB, E_1 = C \cdot AB_1, E_2 = C \cdot AB_2,$   
 $M = C_1 - D_1, N = C_1 + D_1,$   
 $Y_3 = E_1 \cdot N, Z_3 = M \cdot N, X_3 = E_2 \cdot M$

---

## 4 Architecture of Hardware

The high-speed, area-efficient, and humble architecture of the unified group operation unit improves the performance of the ECC processor hardware. The unified group operation or unified point addition formula of the twisted Edwards curve has been used to generate hardware design in this study.

There are two architectures of unified group operation. Both consist of modular multiplication, modular subtraction, modular addition, and modular squaring. Also, both have five consecutive levels of arithmetic operation, in which the arithmetic modules are connected successively. Those units are fixed up in such a way that modules can achieve the smallest path.

The hardware architecture of the first one is shown in Fig. 1. Modular multiplication is a costly operation concerning hardware architecture design. There are interleave and Montgomery modular multipliers. Interleave has both Radix-2 and Radix-4 operations, on the other hand, Montgomery has only Radix-2. We chose Radix-4 interleaved among others because of its high-speed operation and operation on two bits in each clock cycle.

The required arithmetic operation of the first hardware architecture is shown in Table 1. It shows in total 12 modular multiplication, 1 modular squaring, 4 modular addition, and 3 modular subtraction. They are important mathematical units for

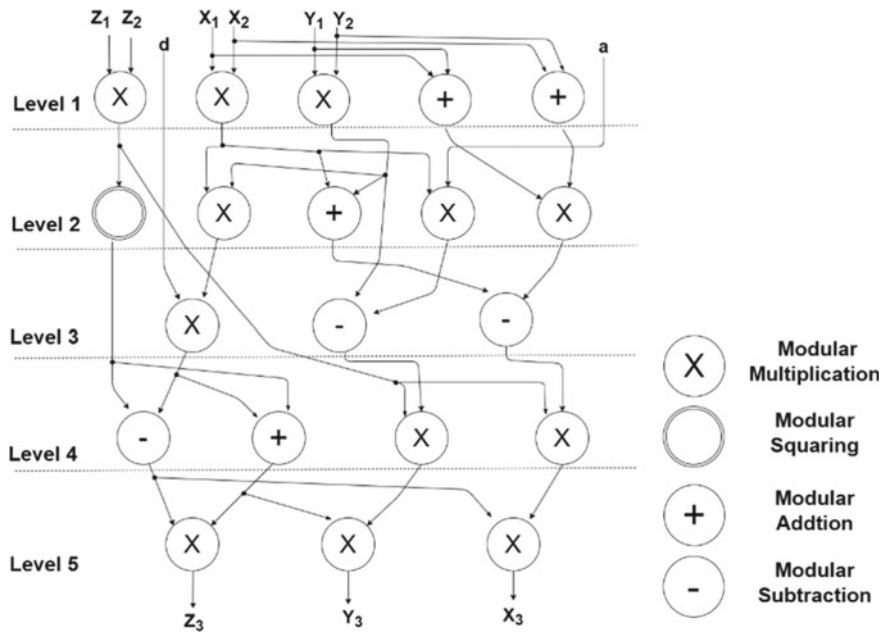


Fig. 1 First hardware architecture for unified group operation

conducting group operations. Table 2 represents the required clock cycles of latency in each level. Due to every level having at least a modular multiplier, each level requires 129 clock cycles. The modular squaring is another version of the modular multiplier.

The second hardware architecture is shown in Fig. 2. The essential arithmetic unit operation of the second hardware architecture for unified group operation is depicted in Table 3. It shows arithmetic units overall 12 modular multiplication, 1 modular squaring, 3 modular addition, and 1 modular subtraction. There is a reduction in arithmetic units in the second hardware architecture than in the first one.

Table 2 shows the required clock cycles or latency at every level. Four levels having at least a modular multiplier required 129 clock cycles. In Level-4, there required one clock cycle. Due to overall clock cycles being reduced, it makes faster data operation for the ECC processor. As from clock cycles requirements, modular multiplication or squaring needs more time than other modular units. The modular inversion is the most costly and time-consuming arithmetic operation which is obstructed for selecting projective coordinates (Table 4).

**Table 1** Overall essential modular operation of first unified group operation

| Operations          | Multiplication | Squaring | Addition | Subtraction |
|---------------------|----------------|----------|----------|-------------|
| $A = Z_1 \cdot Z_2$ | 1              |          |          |             |
| $C = X_1 \cdot X_2$ | 1              |          |          |             |
| $K = Y_1 \cdot Y_2$ | 1              |          |          |             |
| $A_1 = X_1 + Y_1$   |                |          | 1        |             |
| $A_2 = X_2 + Y_2$   |                |          | 1        |             |
| $B = A^2$           |                | 1        |          |             |
| $E_1 = C \cdot K$   | 1              |          |          |             |
| $C_1 = C + K$       |                |          | 1        |             |
| $C_2 = a \cdot C$   | 1              |          |          |             |
| $H = A_1 \cdot A_2$ | 1              |          |          |             |
| $E = d \cdot E_1$   | 1              |          |          |             |
| $J = K - C_2$       |                |          |          | 1           |
| $K_1 = H - C_1$     |                |          |          | 1           |
| $F = B - E$         |                |          |          | 1           |
| $G = B + E$         |                |          | 1        |             |
| $M = A \cdot K_1$   | 1              |          |          |             |
| $N = A \cdot J$     | 1              |          |          |             |
| $X_3 = M \cdot F$   | 1              |          |          |             |
| $Y_3 = G \cdot N$   | 1              |          |          |             |
| $Z_3 = G \cdot F$   | 1              |          |          |             |
| Total               | 12             | 1        | 4        | 3           |

**Table 2** Clock cycle requirement of each level of first unified group operation

| Level | Cycle | Operations                                                                                  |
|-------|-------|---------------------------------------------------------------------------------------------|
| 1     | 129   | $A = Z_1 \cdot Z_2; C = X_1 \cdot X_2; K = Y_1 \cdot Y_2; A_1 = X_1 + Y_1; A_2 = X_2 + Y_2$ |
| 2     | 129   | $B = A^2; E_1 = C \cdot K; C_1 = C + K; C_2 = a \cdot C; H = A_1 \cdot A_2$                 |
| 3     | 129   | $E = d \cdot E_1; J = K - C_2; K_1 = H - C_1$                                               |
| 4     | 129   | $F = B - E; G = B + E; M = A \cdot K_1; N = A \cdot J$                                      |
| 5     | 129   | $X_3 = M \cdot F; Y_3 = G \cdot N; Z_3 = G \cdot F$                                         |
|       |       | Total clock cycles = $129 + 129 + 129 + 129 + 129 = 645$                                    |

## 5 Results of Hardware Simulations and Performance Analyses

This section is for the results of hardware simulation and its analysis of performance for the proposed unified group operation hardware architectures in the prime field.



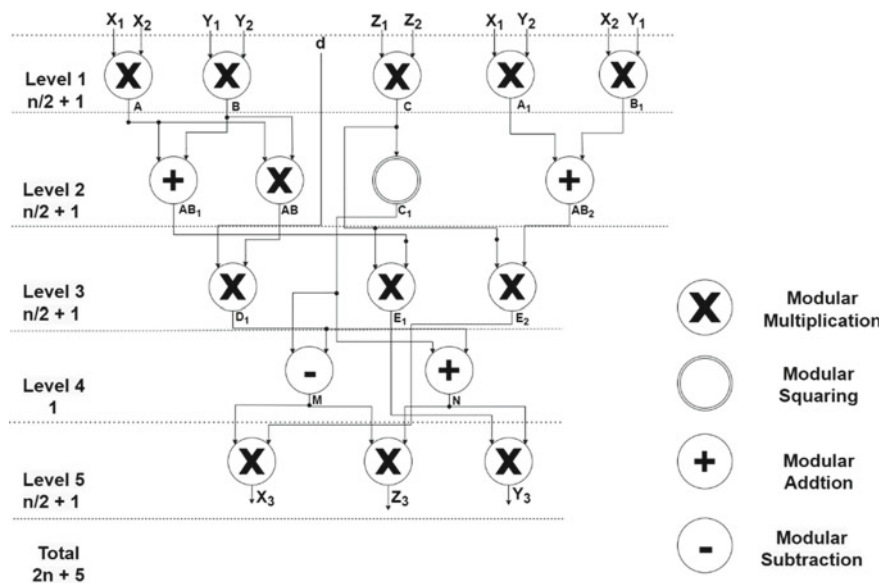


**Fig. 2** Second hardware architecture of unified group operation

**Table 3** Overall essential operation of second unified group operation

| Operations            | Multiplication | Squaring | Addition | Subtraction |
|-----------------------|----------------|----------|----------|-------------|
| $A = X_1 \cdot X_2$   | 1              |          |          |             |
| $B = Y_1 \cdot Y_2$   | 1              |          |          |             |
| $C = Z_1 \cdot Z_2$   | 1              |          |          |             |
| $A_1 = X_1 \cdot Y_2$ | 1              |          |          |             |
| $B_1 = X_2 \cdot Y_1$ | 1              |          |          |             |
| $AB_1 = A + B$        |                |          | 1        |             |
| $AB = A \cdot B$      | 1              |          |          |             |
| $C_1 = C^2$           |                | 1        |          |             |
| $AB_2 = A_1 + B_1$    |                |          | 1        |             |
| $D_1 = d \cdot AB$    | 1              |          |          |             |
| $E_1 = C \cdot AB_1$  | 1              |          |          |             |
| $E_2 = C \cdot AB_2$  | 1              |          |          |             |
| $M = C_1 - D_1$       |                |          |          | 1           |
| $N = C_1 + D_1$       |                |          | 1        |             |
| $Y_3 = E_1 \cdot N$   | 1              |          |          |             |
| $Z_3 = M \cdot N$     | 1              |          |          |             |
| $X_3 = E_2 \cdot M$   | 1              |          |          |             |
| Total                 | 12             | 1        | 3        | 1           |

**Table 4** Clock cycle requirement of each level of second unified group operation

| Level | Cycle | Operations                                                                                          |
|-------|-------|-----------------------------------------------------------------------------------------------------|
| 1     | 129   | $A = X_1 \cdot X_2; B = Y_1 \cdot Y_2; C = Z_1 \cdot Z_2; A_1 = X_1 \cdot Y_2; B_1 = X_2 \cdot Y_1$ |
| 2     | 129   | $AB_1 = A + B; AB = A \cdot B; C_1 = C^2; AB_2 = A_1 + B_1$                                         |
| 3     | 129   | $D_1 = d \cdot AB; E_1 = C \cdot AB_1; E_2 = C \cdot AB_2$                                          |
| 4     | 1     | $M = C_1 - D_1; N = C_1 + D_1$                                                                      |
| 5     | 129   | $Y_3 = E_1 \cdot N; Z_3 = M \cdot N; X_3 = E_2 \cdot M$                                             |
|       |       | Total clock cycles = $129 + 129 + 1 + 129 + 129 = 517$                                              |

**Table 5** Radix-2 and Radix-4 multiplier algorithms comparison

| Parameters                         | Radix-2<br>Interleaved | Radix-2<br>Montgomery | Radix-4<br>Interleaved |
|------------------------------------|------------------------|-----------------------|------------------------|
| Number of slice registers          | 523                    | 1031                  | 1029                   |
| Number of slice LUTs               | 1482                   | 2360                  | 1609                   |
| Number of occupied slices          | 424                    | N/A                   | 668                    |
| Number of LUT flip flop pairs used | 523                    | 1028                  | 353                    |
| Number of bonded IOBs              | 772                    | 773                   | 772                    |
| Clock cycles                       | 258                    | 258                   | 129                    |

Every architecture is synthesized in Xilinx ISE design suite 14.7. We also did Radix-4 and Radix-2 modular multiplication algorithm comparisons. The device family is Virtex-5, where the device and package number are XC5VLX330T and FF1738, respectively. In addition, we chose Radix-4 for data speed as mentioned in Table 5.

Table 6 shows the comparison between the first hardware architecture and the second hardware architecture. At most of the parameters, the second architecture is better than the first one. The second architecture is more area efficient, has fewer clock cycles, less average time, and has good throughput than the first one. Both architectures are suitable for 256 bits operations. The designs are implemented on the FPGA called Virtex-7. The device name is XC7VX690T, and the package name is FFG1157. Throughput and average time are calculated as follows:

$$\text{Throughput} = (\text{Maximum frequency} \times 256) \div \text{Clock cycles}$$

$$\text{Average time} = \text{Clock cycles} \div \text{Maximum frequency}$$

Table 7 shows the comparison between our work and other related published works in the literature in terms of clock cycles requirements with the area and data speed for group operation. Analytic data for other works derived to form different FPGA boards like Virtex-6, Kintex-7, and Virtex-5. The current design offers less area with respect to clock cycles. As shown in Table 7, the second design presented in this paper has better performance in terms of area (slices), area (LUTs), speed/time throughput (Mbps) compared to other references.

**Table 6** Comparison between first and second architectures

| Parameters                         | First architecture | Second architecture |
|------------------------------------|--------------------|---------------------|
| Bits                               | 256                | 256                 |
| Latency (CCs)                      | 645                | 517                 |
| Frequency (MHz)                    | 95.856             | 104.323             |
| Average time ( $\mu$ s)            | 6.72               | 4.96                |
| Throughput rate (Mbps)             | 38.04              | 51.7                |
| Number of slice registers          | 12,039             | 8427                |
| Number of slice LUTs               | 16,630             | 13,609              |
| Number of occupied slices          | 4269               | 3673                |
| Number of LUT flip-flop pairs used | 9814               | 6624                |
| Number of bonded IOBs              | 4                  | 4                   |

## 6 Conclusion

In this research work, 256-bit highly efficient twisted Edwards curve-based unified group operation for ECC processor unit has been proposed. The group operation is one of the core parts of the arithmetic formula of cryptography. We redesigned two architectures from the unified group operation hardware architecture to illustrate the improved performances. A reduction in the hardware architecture as well as areas are identified. The second architecture is quite impressive with respect to the slices, LUTs, maximum frequency, latency, and throughput rate compared to the first architecture. The most important parameters such as slices, LUTs, and clock cycles are reduced in the second design of hardware architecture. Based on a review of overall performance and a comparison of results with related works, our proposed hardware architecture of unified group operation on twisted Edwards curve design provides the best result. For instances, the slices for the first and second architecture were 4269 and 3673, respectively, which is one of the most suitable Virtex-7 architectures.

**Table 7** Performance comparison of unified group operation architecture with other research works over the prime field of 256 bits

| Research work          | Operation               | Platform | CCs        | Area (slices) | Area (LUTs)   | Time ( $\mu$ s) | Throughput (Mbps) |
|------------------------|-------------------------|----------|------------|---------------|---------------|-----------------|-------------------|
| [2]                    | Unified point operation | Virtex-7 | 517        | 4159          | 15,594        | 4.95            | 51.7              |
|                        | Unified point operation | Virtex-6 | 517        | 4292          | 15,593        | 5.55            | 46.7              |
| [1]                    | Unified point operation | Virtex-5 | 646        | 3102          | 17,388        | 5.48            | 46.69             |
| [6]                    | Point doubling          | Kintex-7 | 1556       | 4597          | N/A           | 7.65            | 33.46             |
|                        | Point addition          | Kintex-7 | 1556       | 3861          | N/A           | 7.57            | 33.82             |
| [7]                    | Point doubling          | Kintex-7 | 1290       | 2707          | N/A           | 8.94            | 28.63             |
|                        | Point addition          | Kintex-7 | 1034       | 2577          | N/A           | 6.82            | 37.53             |
| [8]                    | Point doubling          | Virtex-7 | 1032       | N/A           | 19,095        | 8.49            | 30.17             |
|                        | Point addition          | Virtex-7 | 1547       | N/A           | 30,039        | 12.73           | 20.11             |
| Proposed first design  | Unified group operation | Virtex-7 | 645        | 4269          | 16,630        | 6.72            | 38.04             |
| Proposed second design | Unified group operation | Virtex-7 | <b>517</b> | <b>3673</b>   | <b>13,609</b> | <b>4.96</b>     | <b>51.7</b>       |

## References

1. Hossain MR, Hossain MS, Kong Y (2019) Efficient FPGA Implementation of unified point operation for twisted Edward curve cryptography. In: 2019 international conference on computer, communication, chemical, materials and electronic engineering (IC4ME2), pp 1–4
2. Islam M, Hossain M, Hasnan M, Shahjalal M, Jang YM et al (2020) Design and implementation of high-performance ECC processor with unified point addition on twisted Edwards curve. *Sensors* 20:5148
3. Hossain MS (2017) High-performance hardware implementation of elliptic curve cryptography. Ph.D. dissertation, Macquarie University, Sydney, Australia
4. Bernstein DJ, Birkner P, Joye M, Lange T, Peters C (2008) Twisted Edwards curves. In: International conference on cryptology in Africa, pp 389–405
5. [https://en.wikipedia.org/wiki/Twisted\\_Edwards\\_curve](https://en.wikipedia.org/wiki/Twisted_Edwards_curve)
6. Hossain MS, Kong Y, Saeedi E, Vayalil NC (2016) High-performance elliptic curve cryptography processor over NIST prime fields. *IET Comput Digit Tech* 11:33–42

7. Kudithi T, Sakthivel R (2019) High-performance ECC processor architecture design for IoT security applications. *J Supercomput* 75:447–474
8. Rahman MS, Hossain MS, Rahat EH, Dipta DR, Faruque HMR, Fattah FK (2019) Efficient hardware implementation of 256-bit ECC processor over prime field. In: 2019 international conference on electrical, computer and communication engineering (ECCE), pp 1–6

## Chapter 25

# IoT-Based Smart Office and Parking System with the Implementation of LoRa



Robi Paul  and Junayed Bin Nazir 

**Abstract** In recent times, the Internet of Things (IoT) has been getting a lot of traction. Embedded systems have become ingrained in our daily lives connecting day-to-day devices to a network. IoT-enabled devices have helped us operate, monitor, and do various tasks efficiently from afar. This is achieved while consuming less space with each iteration due to technological advancements. Being compact and modular, IoT-based devices are frequently battery-powered and require a large battery backup. Although in an ideal situation, a low power consuming device that also traverses great distances is expected. However, many current technologies, such as Zig-Bee, Wi-Fi, and Bluetooth, take a lot of power and aren't ideal for battery-powered systems. LoRa is a relatively new technology that is rapidly gaining traction. LoRa can consume a fraction of energy while maintaining a higher coverage area, making it a great contender for IoT applications. This paper covers a combinational work of implementing smart office and parking systems using IoT-based technology, while the communication between different nodes is handled using LoRa. The implementations of LoRa have proven to show greater performance while consuming significantly less power. Therefore, implementing the work can surely ease the overall office management while being energy efficient.

**Keywords** IoT · Embedded systems · Zig-Bee · Wi-Fi · Bluetooth · LoRa · Low-powered device

## 1 Introduction

Modern tools and technology have already become a driving factor in decreasing human effort for regulating everyday activities and increasing effectiveness with limited infrastructure. A prediction given by Enterprise Management Associates (EMA) states that around 98 percent of total enterprises in the current time will

---

R. Paul (✉) · J. Bin Nazir

Department of Electrical & Electronic Engineering, Shahjalal University of Science and Technology, Sylhet, Bangladesh

e-mail: [robipaul01733@gmail.com](mailto:robipaul01733@gmail.com)

depend on the automated system to develop smart cities, smart workplaces, smart homes, smart institutes, smart industries, and so on in the next several years. In 2019, researchers have seen an increase in market value with an amount of \$745 billion regarding the Internet of Things (IoT) which is a key technology for establishing smart technologies [1]. The International Data Corporation (IDC) estimates that the Internet of Things (IoT) will develop at a double-digit yearly rate globally from 2017 to 2022, surpassing the \$1 trillion industry in 2022 [2]. However, one of the fundamental drawbacks of the present system is that they consume a considerable amount of power to operate at an optimum level. It increases the maintenance cost and discourages a large portion of the industries and offices from embracing smart systems. As a result, one of Industry 4.0's main concerns is reducing power consumption and expanding the communication range of smart devices [3–5].

Over the past years, long range-based telecommunication or LoRa-based telecommunication has gained much attention due to its lower power consumption and considerably higher area coverage. LoRa is a CSS or chirp spread spectrum-based technology which is a wireless modulation method [6]. It uses chirp pulses to encode radio waves. This works similarly like the dolphins and bats usually communicate. Due to its advantages in area coverage and power management, large-scale companies with vast infrastructures slowly adapt to different LoRa-based communication systems. This has tremendously helped them to monitor and control appliances at competitively lower cost and power efficiently while being in a fault-free, hazard-free manner from a large distance. A LoRa-based automation technology has a simple and cost-efficient option [7, 8]. M. Shahjalal et al. have shown that around 45.12% of people worldwide with over 3.5 billion smartphones are currently controlling their appliances remotely [9]. With the constant improvement of lifestyle and technology, this number is growing each day by day. Here, Table 1 represents a comparison view among currently available various technology aspects with their limitations and advantages.

With the advantages LoRa holds over the traditional communication devices customarily used for IoT applications, researchers are now relentlessly trying to integrate into newer sectors of normal and industrial sectors. They have pursued different approaches from time to time over the recent decade. M. S. Islam et al. took an approach toward medical implementation. In this work, they have represented an integration of sensor nodes that collect human body conditional data and then send it to the doctor for further treatment [14]. Their work showed the possibilities of decentralizing medical treatment and spreading quality medical treatment to people using LoRa-based technology. Again, Y. W. Ma et al. took initiatives toward integrating the LoRa in the agriculture system [15]. Here, they showed a system where the sensor nodes take various data from the soil and the environment and then send them to the central nodes via the LoRa. Then, these data are processed and taken measures accordingly. Although there are diverse approaches, and each serves different aspects, their integration of LoRa-based technology has proved it is an alternative in low power data transfer while covering a large area.

In this paper, we have taken an approach toward developing a smart office and parking system with the implementation of LoRa-based IoT technology. The

**Table 1** Comparison view between existing and proposed technologies

| Technology             | Coverage range, power consumption | Radio band, radio spectrum                          | Transmit-receive data size, network technology | Limitation/advantage                      |
|------------------------|-----------------------------------|-----------------------------------------------------|------------------------------------------------|-------------------------------------------|
| Bluetooth [10]         | 10–100 m, 15–30 mA per packet     | UHF, 2.4–2.4835 GHz                                 | 1–3 Mbps, PANs                                 | Short range                               |
| Wi-Fi [11]             | 50–100 m, 2–20 W                  | UHF and SHF ISM, 2.4–5 GHz                          | 1–9608 Mbps, WLANs                             | Short distance, high battery power        |
| Zigbee [12]            | 10–100 m, 150 mA                  | Unlicensed ISM bands, 2.4 GHz, 915, 868 and 784 MHz | 20–250 kbps, WPANs                             | Short distance, maintenance cost too much |
| Sigfox [13]            | 20–25 km, 78 mA                   | Licensed ISM bands, 868, 915 and 433 MHz            | 100 bps (4–140 data/day), LPWAN                | High module cost, high battery power      |
| Proposed system (LoRa) | 10–20 km, 32 mA                   | Unlicensed ISM bands, 868, 915 and 433 MHz          | 290 bps–50 kbps, LPWAN                         | Longer range, low battery power           |



work covers a broad structure for parking management as well as monitoring and maintaining security for the employees. As the LoRa is proven to have a lower power consumption rate while having a greater area coverage, a significant improvement is observed in the received signal quality while maintaining communication significantly greater and secured manner.

## 2 Methodology

As LoRa is a relatively new and comparatively dynamic technology, a limited number of studies have highlighted the device's practical implementation and performance. In this study, we will be representing a smart office and parking system with the implementation of LoRa-based communication. We have divided the system into three portions for efficient work, each sharing data for seeming less connectivity. The overall process for the data transmission and hardware description are as follows.

### 2.1 Parking Entry Unit

The parking entry unit is the initial portion of our system. The interface acts as an access point for the available parking spots and necessary guidelines with directions. The communication through the user identification unit and the parking entry unit is established through the LoRa module. The device is well equipped with a backup battery source for maintaining uninterrupted communication and proper parking area distribution for employees and other staff (Fig. 1).

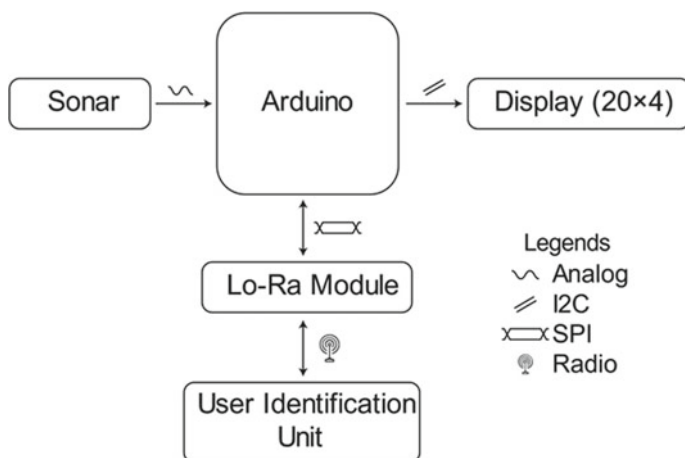


**Fig. 1** Parking entry unit

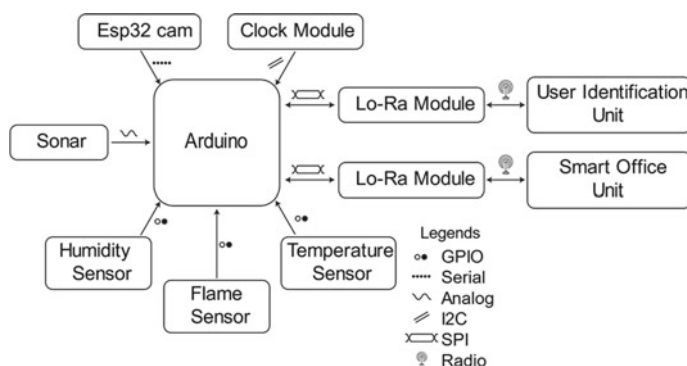


**Fig. 2** User identification unit

## 2.2 User Identification Unit

The user identification unit is the primary identification unit for our system to identify parking users. For the identification purpose, the device uses an onboard ESP-32 cam which only activates when an obstacle is sensed less than a threshold value by the sonar. This approach saves power consumption, which prolongs the duration of power supply by the onboard battery during a power outage. The system collects the number plate data and matches it to memory by identifying parking users. The device is also equipped with a temperature and flame sensor so that a sudden anomaly is avoided, and the safety of the parking area is ensured. The device also uses a clock module to calculate the parking time to calculate parking expenses correctly. The module also covers the security aspect of office employees. Here, Fig. 2 shows a functional diagram of the user identification system.

## 2.3 Smart Office Unit

The smart office is the main access point for the system to identify office employees before entering the premises while maintaining the flow of data accumulated by the previous subsections. The device has an RFID reader, a fingerprint sensor, and a keypad for letting the employees choose their option for identification. After successful entry, the device notes the entering and exiting timing using the onboard RTC module. It helps the device to calculate total office hours while also maintaining office security. The device also contains a temperature and an infrared sensor. Any sudden change in the environment can trigger the module, which conveys to the authorized authority and the local fire department using the onboard GSM module. The same path is also used when an anomaly is detected by the previous sub-devices received by the LoRa-based communication. Although the device is attached to the main power line, it is also backed up by a battery which the use of LoRa prolongs

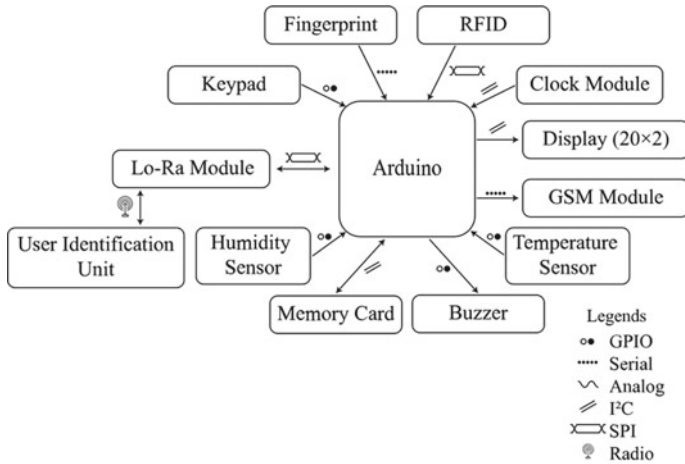


**Fig. 3** Smart office unit

power delivery as it is the primary source of power consumption. Here, Fig. 3 shows an operational flow of our smart office system.

### 3 Parameters for LoRa Transmissions

The development of our LoRa-based system must be in line with our functional and operational needs, which means considering factors like real time, security, traceability, and integration with GPS and other smart sensor nodes. This includes considering technical parameters such as operating frequencies, bandwidth, as well as environmental factors such as large buildings or other obstructions that impede the line of sight. We know LoRa modulation is directly linked to data rate (DR), spreading factor (SF), and bandwidth (BW) parameters. As a result, these characteristics must be chosen in accordance with the functional and technological needs of the IoT systems. The relationship between the spreading factor and bandwidth can be represented as follows:

$$DF = SF \times \frac{1}{\frac{2SF}{BW}} \text{ bits s}^{-1} \quad (1)$$

The transmission time, also known as the “Time on air” is the summation of transmission time set for preamble length with message transmission time for the physical layer’s which can also be determined by the symbol time. These parameters are also directly related to the spread factor and bandwidth. The equations for the total transmission time can be expressed as follows:

$$T_{tx} = T_{preamble} + T_{message} \quad (2)$$

$$T_{preamble} = T_{sp} \times (N_{sym} + 4.25) \quad (3)$$

Here,  $N_{sym}$  represents the total number of symbols used by the transceiver section. Again, the symbol period shown by  $T_{sp}$  can be derived as following:

$$T_{sp} = \frac{2^{SF}}{BW} \quad (4)$$

$$T_{message} = T_{sp} \times N_p \quad (5)$$

Here,  $N_p$  is the total symbols transmitted which can also be determined as,

$$N_p = 8 + \max \left[ \text{ceil} \left[ \frac{28 + 8PL + 16 \times CRC - 4SF}{4 \times (SF - 2DE)} \right] \times (CR + 4), 0 \right] \quad (6)$$

where CR represents the coding rate. It normally ranges in between 1 and 4. The PL stands for the payload rate, which means the physical payload lengths in bytes. The CRC of the cyclic redundancy check represents the error detection rate, and DE represents the low data rate which is turned on for calculations.

As the receiver's section receives an incoming signal, it must be between the sensitivity levels of the transceivers. Otherwise, there is a possibility of misdirection or total signal loss. So, the system must consider the parameters for losses and gains from the transmitter side to the receiver end,

$$P_{rx}(\text{dBm}) = P_{tx}(\text{dBm}) + G_{sys}(\text{dB}) - L_{sys}(\text{dB}) - L_{ch}(\text{dB}) - M(\text{dB}) \quad (7)$$

$G_{sys}$  and  $L_{sys}$  are the antenna and cable gains and losses, respectively.  $L_{ch}$  is related to transmission medium losses, which are also affected by environmental factors or reliefs in the geographic area. The fading margin is connected to  $M$ . As our system is implemented indoors, we also must consider the noise floor, which implies,

$$NF(\text{Noise Floor}) = 10 \times \log_{10}(k \times BW \times 1000) \text{ dBm} \quad (8)$$

Here,  $K$  is the Boltzmann's constant, and  $T_{293}$  is 293 K also considered as the room temperature.

## 4 System Description

This chapter will be focusing on the structural demonstration of our three staged devices. The detailed view is given below:

Initially, we used Kicad to design the primary circuitries for this project and then printed the design on fiberglass. Although the three sections of the project have a different purpose, the basic structure for each part is kept as identical as possible with minimal dissimilarities in a few sensors. Each circuit board has an onboard power distribution circuit that switches between battery and main-line connection to ensure minimal power usage while continuing an uninterrupted communication system. The following figures show an array of the modules used in real life with all the components on board in front and rear-view perspectives. Figure 4a and b represents the parking entry unit. The module contains a  $(20 \times 4)$  liquid crystal display and a sonar module at the front end for detecting and communication purpose.

Figure 4c and d shows the module for the user identification unit. The frontend side houses the sonar, Esp-32 cam, and an inferred sensor to detect the car and scan for hazardous situations. All the other sensors are kept at the rear for safety reasons. For the same reason, the components for the smart office unit are also kept at the rear except for the  $(20 \times 2)$  display, Rfid module, keypad, and fingerprint sensor. This can be seen in Fig. 4e and f. As these sensors are used for identification, they are housed on the front side. All boards communicate and share information through the LoRa communication system in harmony and uninterruptedly.

## 5 Results

This section focuses on the experimental results and performance aspects of our system. Here, we have tried to collect data from different aspects and scenarios considering spreading factors, the coding rate, bandwidths, and other critical factors for an adequate understanding of our system performance. The properties of the three LoRa modes employed with the SX1278 transceiver are summarized in Table 2.

### 5.1 Transmitter Performance Analysis

As per Eqs. (2)–(6), the time on air shows similar results as a function of the payload size at different spreading factors and coding rates. For the test, we have set the channel as 500 kHz. Figure 5a shows a linear increment in time on air value with each spreading factor variation. This also shows as the spreading factor increases, the amount of power consumption also increases. Again Fig. 5b shows a relation between the coding factors with the time on air. An increased amount of encoding bits increases the packet transmission, which corresponds to the transmitter module's increased power consumption.

Figure 6 shows a relation between the amount of power consumption and the size of the actual payload sent by the transceiver module. As our system is subjected to sending a variable amount of data over the lifetime of the battery cell, the power consumption rate plays a vital role in performance judgment. As the amount of dataset

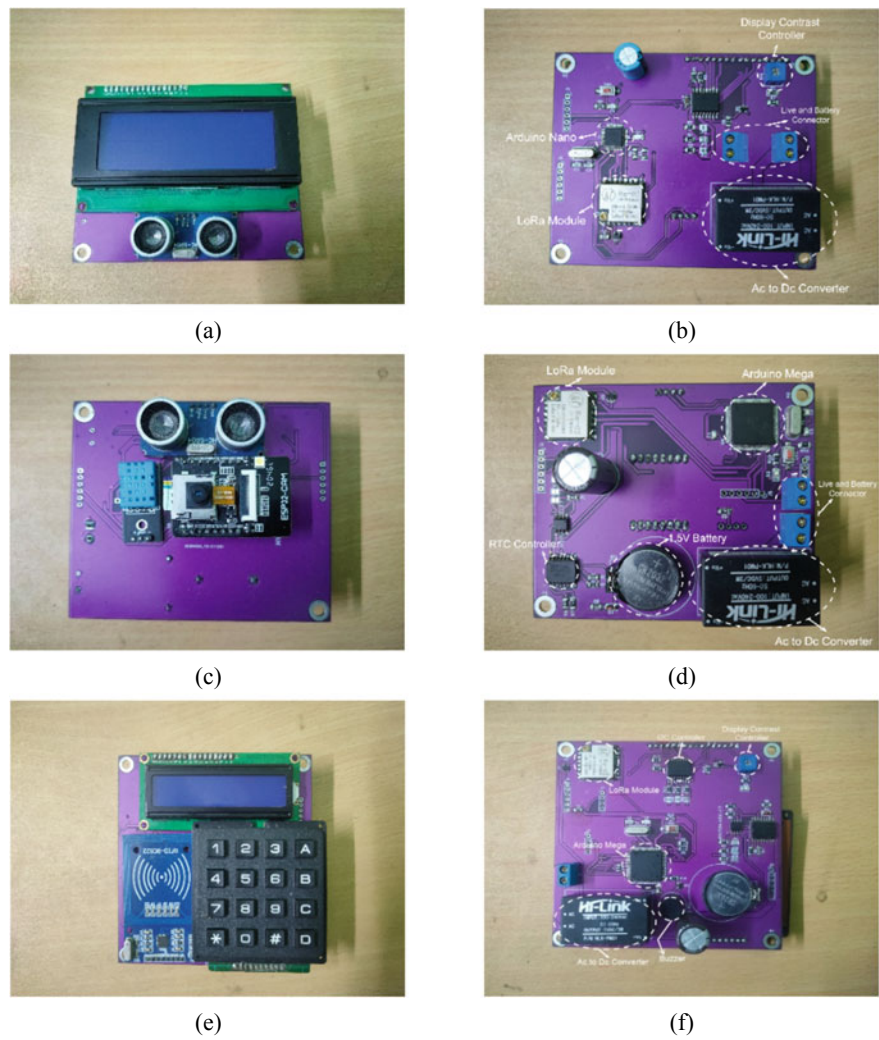


**Fig. 4** **a** Front side of the parking entry unit **b** back side of the parking entry unit **c** front view of the user identification unit **d** back view of the user identification unit **e** front side of the smart office unit **f** back side of the smart office unit

**Table 2** Different LoRa modes

| Modes  | Characteristics                                         | Explanations                                              |
|--------|---------------------------------------------------------|-----------------------------------------------------------|
| Mode 1 | $BW = 125 \text{ kHz}$ , $SF = 12$ , $CR = \frac{4}{5}$ | Largest distance mode (max range and slow data rate)      |
| Mode 2 | $BW = 250 \text{ kHz}$ , $SF = 10$ , $CR = \frac{4}{5}$ | Intermediate mode                                         |
| Mode 3 | $BW = 500 \text{ kHz}$ , $SF = 7$ , $CR = \frac{4}{5}$  | Minimum range, high data rate, and minimum battery impact |

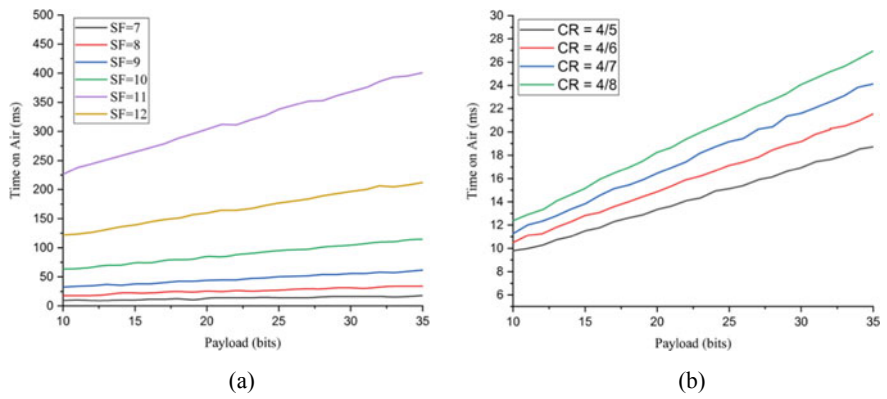


**Fig. 5** **a** Time on air versus payload at different SF; and **b** time on air versus payload at different CR

increases, the time on air also increases, which means an increment in effort for the transceiver module to send data, which naturally increases the amount of power consumption and decreases the battery life. However, as LoRa devices consume power at the microampere range, even the worst-case with a large dataset can draw out around 11 years of battery life from an average capacity battery.

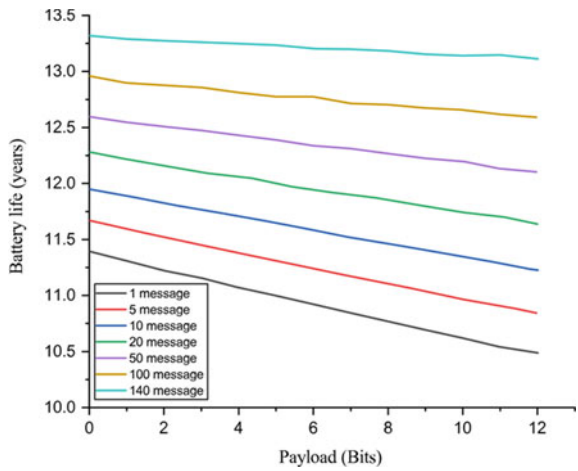
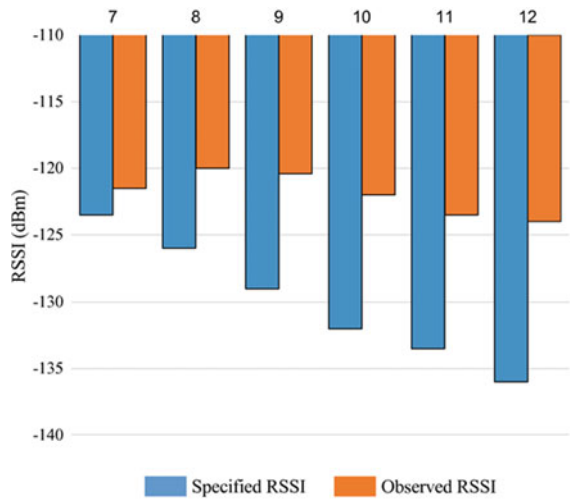


**Fig. 6** Battery life in transceiver with various payload sizes, SF = 10, coding rate 4/6, and CRC enabled. Other parameters are 8 symbols preamble and bandwidth 125 kHz

**Fig. 7** Observed RSSI pattern with various spreading factors



**5.2 Receiver Sensitivity Analysis**

This subsection focuses on the receiver sensitivity of the LoRa module. For this experiment, we have chosen a bandwidth of 125 kHz and a code rate of 4/5. Around 10,000 packets of data were channeled through the system to measure the received signal strength indicators (RSSI) accurately. Here, we set the transmission power for the transmitter module around 2 dBm with a 3 dBi antenna attached to the receiver and transmitter end. As the distance between the two nodes increased, packets began to be dropped off the order of 100 m. Figure 7 shows the RSSIs with the lowest values recorded.

These measured findings are somewhat higher than the specified values, and there is no evidence of the predicted decline as the spreading factor increases. The packets with the lowest RSSIs, on the other hand, were also received with a high SINR, close to 20 dB. This is most likely because the entrance is indoors, resulting in different shadows.

**5.3 Other Sensors Performance**

One of the various sensors used in the user identification unit is the temperature and flame sensors. Both sensors are well tuned and work with correlation with each other to successfully identify and warn necessary personnel about a possible hazard. The combination of two sensors is implemented to ensure the existence of potential fire hazards in the parking area as often the fire sensor can detect inferred light emitted from a possible car of sunlight exposure. As the system detects value over the threshold limit, the device then conveys the information to the office unit, sending



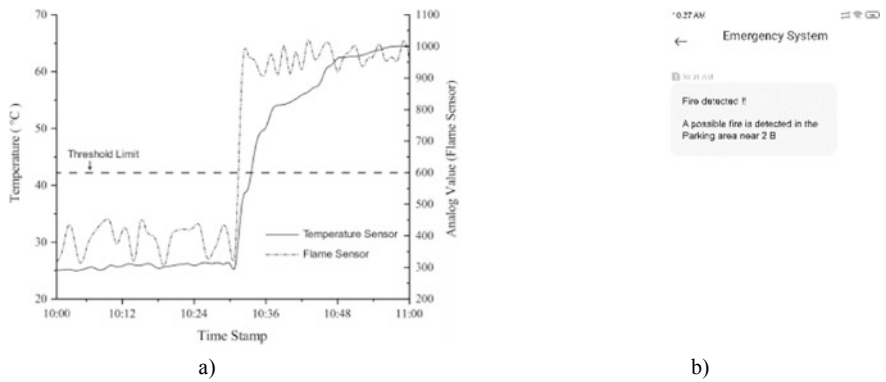


**Fig. 8** **a** Real-time data for flame and temperature sensor. **b** Emergency message sent by the system

an SOS message to the assigned persons about the situation. Figure 8a shows a graph showing a real-time dataset with a flame and temperature sensor. Here, we used a gas litter for testing purposes to simulate fire. As soon as the flame and temperature sensors detect the anomaly, the device sends a message with an accurate position using the GSM system. Figure 8b shows the notification message sent by the GSM module with position data.

Table 3 shows a list of employees represented by the Id user matching their fingerprint samples after the registration process. There were 400 samples taken from the sensor to calculate the sensor performance. There were 20 ID-holding employees, each giving 20 samples for matching. Although in maximum cases, the result came affirmative in 18 cases; it failed to recognize the fingerprint from the owner. This can be caused due to a variety of issues, including a misplaced fingerprint, a damp or greasy finger, or even a narrow finger surface area.

From the data, we have can calculate the percentage of accuracy from the following equation:

$$\text{Percentage of Accuracy} = \frac{\text{number of successful fingerprint matching}}{\text{total number of fingerprint test data}} \times 100\%$$
$$\text{Percentage of Accuracy} = \frac{382}{400} \times 100\%$$
$$= 95.5\%$$

**Table 3** Fingerprint sensor matching test

|               |    |    |    |    |    |    |    |    |    |    |
|---------------|----|----|----|----|----|----|----|----|----|----|
| ID user       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| Match testing | 20 | 19 | 18 | 20 | 20 | 18 | 17 | 20 | 19 | 20 |
| ID user       | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Match testing | 20 | 20 | 20 | 19 | 20 | 17 | 20 | 18 | 18 | 19 |

## 6 Conclusions

In this investigation, we have designed, developed, and analyzed the performance aspect of our IoT-based smart office and parking system. The system was designed to maintain an uninterrupted communication stream between modules while consuming competitively lower power consumption and communicating from a distance up to 1.5 km in indoor situations. All sensors in various modules are fine-tuned to ensure accurate data for monitoring office employees' activities while also maintaining the parking and office security systems. In simulated scenarios, the system monitored environmental temperature and humidity and detected fire with a 92% accuracy. The smart office has an accuracy of 95.5% for the fingerprint sensor. However, this percentage goes even higher when we put the Rfid and keypad system, where both have an accuracy of near 100%. Again, the ability to connect with necessary personnel in hazardous situations also makes the device a valuable asset for maintaining office and parking area management and security coverage.

## References

1. MacGillivray C, Torchia M, Bisht A, Kalal M, Leung J, Membrila R, Siviero A, Wallis N, Torisu Y (2020) Worldwide Internet of Things forecast update. IDC Research, pp 2019–2023
2. Tankovska H. Internet of Things (IoT) spending worldwide 2021. Accessed 09 Feb 2022
3. Paul R, Ahmed F, Ahmad MM, Rahman A, Hosen N (2021) A smart approach towards home automation and security system based on IoT platform. *Int J Innov Sci Res Technol*
4. Jadhav AR, Rajalakshmi P (2017) IoT enabled smart and secure power monitor. In: *IEEE region 10 symposium (TENSYP)*, pp 1–4
5. Paul R, Ahmed F, Shahriar TR, Ahmad M, Ahammad A (2021) Centralized power monitoring & management system of an institution based on Android app. In: *International conference on automation, control and mechatronics for Industry 4.0 (ACMI)*, pp 1–5
6. Zhou Q, Zheng K, Hou L, Xing J, Xu R (2019) Design and implementation of open LoRa for IoT. *IEEE Access* 7:100649–100657
7. Usmonov M, Gregoretti F (2017) Design and implementation of a LoRa based wireless control for drip irrigation systems. In: *2017 2nd international conference on robotics and automation engineering (ICRAE)*, pp 248–253
8. Lora Alliance. Available at: <https://www.lora-alliance.org/>. Accessed 25 Feb 2022
9. Shahjalal M, Hasan MK, Islam MM, Alam MM, Ahmed MF, Jang YM (2020) An overview of AI-enabled remote smart-home monitoring system using LoRa. In: *International conference on artificial intelligence in information and communication (ICAIIIC)*, pp 510–513
10. Tosi J, Taffoni F, Santacatterina M, Sannino R, Formica D (2017) Performance evaluation of Bluetooth low energy: a systematic review. *Sensors* 17:2898
11. Vejlgård B, Lauridsen M, Nguyen H, Kovács IZ, Mogensen P, Sørensen M (2017) Coverage and capacity analysis of Sigfox, LoRa, GPRS, and NB-IoT. In: *85th vehicular technology conference (VTC Spring)*, pp 1–5
12. Lauridsen M, Nguyen H, Vejlgård B, Kovács IZ, Mogensen P, Sørensen M (2017) Coverage comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km<sup>2</sup> area. In: *85th vehicular technology conference (VTC Spring)*, pp 1–5
13. Sinha RS, Wei Y, Hwang SH (2017) A survey on LPWA technology: LoRa and NB-IoT. *Ict Express* 3(1):14–21

14. Shahidul Islam M, Islam MT, Almutairi AF, Beng GK, Misran N, Amin N (2019) Monitoring of the human body signal through the Internet of Things (IoT) based LoRa wireless network system. *Appl Sci* 9(9):1884
15. Ma YW, Chen JL (2018) Toward intelligent agriculture service platform with LoRa-based wireless sensor network. In: *International conference on applied system invention (ICASI)*, pp 204–207

# Chapter 26

## Bandwidth Borrowing Technique for Improving QoS of Cluster-Based PON System



Mehedi Hasan, Sujit Basu, and Monir Hossen 

**Abstract** The usage of online polling-based dynamic bandwidth allocation (DBA) algorithms in a passive optical network (PON) substantially decreases the granting, back-off delays and end-to-end latency when compared to offline polling-based DBA algorithms. One of the major drawbacks of these DBA scheme is lack of fairness in bandwidth allocation for the ONUs connected to a PON system under different load conditions. To improve the quality of services (QoS) of the PON system cluster-based polling algorithm was introduced. Where the optical line terminal (OLT) separates all active ONUs into several clusters then performs DBA algorithm cluster by cluster in each time cycle and excess bandwidth from one cluster is added as surplus bandwidth to the next cluster once available. But the problem occurs when the initial clusters are exhausted. In this situation, apart from sharing the surplus bandwidth with the next cluster, the initial clusters themselves would need some extra bandwidth to be served. In this paper, cluster-based PON with bandwidth borrowing scheme is introduced to improve the QoS of the PON system when the initial clusters are highly loaded. Where the OLT separates all active ONUs into several clusters throughout each time cycle, but here apart from sharing the surplus bandwidth, the highly loaded initial clusters will borrow bandwidth from the next cluster, and continue it up to the last cluster in each time cycle to fit the boundary of a time cycle. For this scheme, performances are evaluated through simulation for three different performance parameters and the computer simulated results are also compared with the existing cluster-based PON system. The simulation results reveal that the proposed algorithm outperforms the existing one.

**Keywords** Bandwidth borrowing · Cluster-based PON · Online and offline polling

---

M. Hasan · M. Hossen (✉)

Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [mhossen@ece.kuet.ac.bd](mailto:mhossen@ece.kuet.ac.bd)

S. Basu

Institute of Information and Communication Technology, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

## 1 Introduction

The bandwidth demands of Internet users and their numbers are increasing significantly day by day [1]. For this reason, the usage of the passive optical network (PON) system is getting more and more accepted and preferred among the end-users. It is also due to the fact that less fiber is required, only passive devices are used, and it is simple to maintain [2]. The physical connection of these popular PONs is maintained from an optical line terminal (OLT) to many optical network units (ONUs) by a passive optical splitter in a tree topology, i.e., a one-point-to-multi-point network. Due to such kind of structure, two types of data transmission directions are present and for two different directions, the system uses two different wavelengths [3]. The direction from OLT to ONUs is called downstream whose traffic nature is broadcasting and from ONUs to OLT is called upstream whose traffic nature is multipoint-to-point. For this specific nature of upstream traffic, ONUs would share a single upstream channel through time division multiple access (TDMA) technique, while for the downstream direction it is not required. So, different bandwidth allocation techniques are put in by OLT in case of utilizing the upstream channel bandwidth efficiently.

In recent years, the dynamic bandwidth allocation (DBA) algorithm is the most favored bandwidth allocation technique that is carried out by the OLT based on the Report messages obtained from the ONUs in each time cycle. PON is split into two basic kinds depending on DBA algorithm processing, namely offline and online polling-based PON. In an offline polling system, the DBA process is completed in a time cycle after receiving all Report messages from ONUs [4]. That implies ONUs must wait for grant messages until the OLT gets all of the Request messages sent by the ONUs in a time cycle. This introduces an increased value of granting delay, as well as an increase in the idle time between two consecutive time cycles. The online polling system was implemented primarily to address these issues [5]. In the online polling scheme, grant messages are delivered by the OLT as soon as it gets the Report message from a specific ONU, without taking into account the Report messages from other ONUs. As a result, the problem of having a huge granting delay in the case of an offline polling scheme disappears in an online polling system. But it becomes challenging to guarantee fairness in bandwidth allocation when the OLT provides a granted window to one ONU without taking into account the Report messages from other ONUs. One of the solutions of distributing the excess bandwidth in an online polling-based approach is demonstrated in Ref. [6]. In this multi-thread polling scheme, the surplus bandwidth is estimated from a lightly loaded ONU and distributed across the following sequences of ONUs in a time cycle. Although this provided a solution to how the excess bandwidth is going to be distributed, the condition is limited up to some specific load conditions. For example, if the beginning ONUs are highly loaded relative to the last sequenced ONUs in any time cycle, desirable excess bandwidth distribution is not achievable. Later the two-phase polling algorithm was developed to address this issue of inefficient cycle time usage [7]. The ONUs of a system is separated into two groups in this algorithm, and the DBA algorithm is conducted as soon as the Report messages of one sub-group

are received by the OLT. However, this method does not include the utilization of unused time windows from the first sub-group of ONUs to the second sub-group of ONUs, as well as a priority scheduling mechanism to service the busiest ONUs.

The authors of Ref. [8] presented a dynamic upstream data transmission sequence (DUDTS) algorithm for cluster-based PON (CP) that incorporates the benefits of both existing offline and online polling-based systems. In a CP, in each time cycle, an OLT separates all active ONUs into a number of clusters, and unused bandwidth from one cluster is added as surplus bandwidth to the next cluster. This idea of clustering the ONUs, successfully alleviated the difficulties caused by the two-phase polling technique. Because the unused bandwidth of an initial cluster in the present CP system is utilized by the next cluster, and there is a mechanism for priority scheduling with the help of Report messages and traffic incoming status. The condition is defined in this current CP system when the requested time window for a cluster is shorter than the time window that was pre-specified for a cluster. However, the other condition is not established when the requested window for cluster is greater than the time frame that was pre-specified for a cluster.

In practical life in a CP system, the initial clusters can be highly loaded. Enthusiasm is grown to solve this particular problem in the existing CP system that motivated us enough to introduce CP with bandwidth (BW) borrowing (CPBB) technique. By the proposed scheme, we have solved the problem of inefficient BW allocation by borrowing BW from next successive cluster for exhausted ONUs under a cluster. As a result of introducing the proposed scheme, BW allocation will be more efficient as well as the buffering delay will be decreased with higher fairness.

The main concept of our proposed CPBB technique for a CP system is briefly discussed in Sect. 2. We have also depicted and discussed the computer simulated results and compared them with the existing CP system for three different performance parameters: grant-to-request-ratio (GTRR), buffering delay and fairness in Sect. 3. Finally, the paper is concluded in Sect. 4.

## 2 Proposed BW Borrowing Technique for CP System

It is an improved algorithm of CP [9] when the prior sequence clusters are heavily loaded. If the requested time window size of an initial cluster is greater than the predefined window size of that cluster in a time cycle, then the OLT will borrow the required extra window from the predefined window of the next cluster to ensure better bandwidth allocation for the ONUs of the exhausted cluster. Thus, next cluster window will be started from the point exactly where the window for the updated prior cluster is finished with a window size equals to the predefined window size for each cluster. After that OLT will check for the successive clusters prior to the last cluster if the requested window is greater than the predefined window size of each cluster or not. If it is again true, then OLT will again borrow bandwidth from the next cluster and update the cluster size accordingly. The process will continue and OLT will adjust the time cycle boundary in the last cluster window. In Fig. 1, the

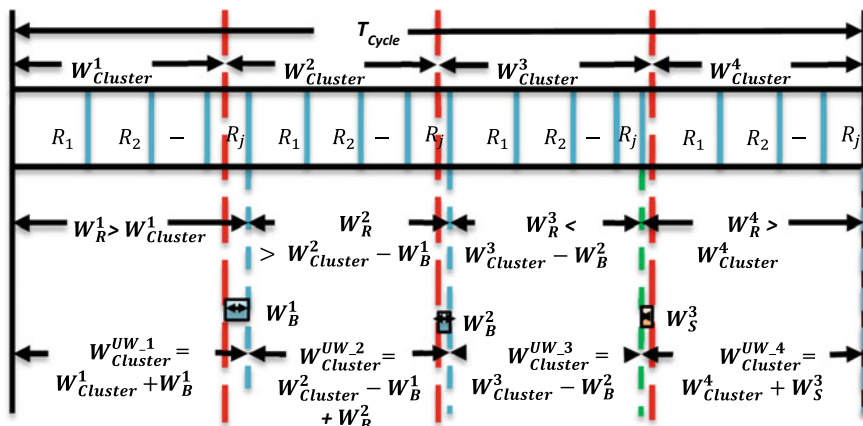


**Fig. 1** Window size distribution of the cluster-based PON using proposed bandwidth borrowing technique

proposed CPBB scheme is depicted in details. At first, OLT separates the total ‘ $N$ ’ number ONUs in ‘ $C$ ’ number of clusters or groups. After clustering there are now ‘ $n = N/C$ ’ number of ONUs per cluster where  $n = 1, 2, \dots, j$ . OLT in the meantime defines the predefined window size for each of the clusters.

The following equation is used to calculate the predefined window size of cluster  $i$ ,  $W^i_{Cluster}$  which will be the ratio of total time window ( $T$ ) to cluster number.

$$W^i_{Cluster} = T/C, \quad (1)$$

where  $i = 1, 2, \dots, C$ . It is worthwhile to mention that OLT will need this  $W^i_{Cluster}$ , so that it can compare with the requested window size of a cluster based on the Report messages and take decisions either to borrow any time window for a cluster or does it has any surplus window for next cluster. According to Fig. 1, the OLT classify the  $N$  number of ONUs into 4 clusters and their corresponding predetermined cluster windows are  $W^1_{Cluster}$ ,  $W^2_{Cluster}$ ,  $W^3_{Cluster}$  and  $W^4_{Cluster}$ . After determining the predetermined cluster window size, the OLT is going to find the sum of the requested window of each cluster.

From the figure, for cluster<sup>1</sup>, OLT will be adding the window sizes of ONU<sub>1-j</sub> and it is seen that the sum of the requested window ( $W^1_R$ ) of the ONUs for cluster<sup>1</sup> is bigger than the  $W^1_{Cluster}$  size, i.e.,  $W^1_R > W^1_{Cluster}$ . In this case, there is no surplus window or idle window based on the Report messages, but it has some extra window demand in comparison with the predetermined window size. That means the ONUs of cluster<sup>1</sup> is highly loaded and the  $W^1_{Cluster}$  is not enough to meet its bandwidth demand. To ensure efficient bandwidth distribution, OLT will borrow the needed extra BW ( $W^1_B$ ) from the cluster<sup>2</sup>, i.e.,  $W^2_{Cluster}$ , and can be calculated as below:

$$W^1_B = W^1_R - W^1_{Cluster} \quad (2)$$

Then the window size of cluster<sup>1</sup> is updated as Eq. (3) and the maximum allocated window size for each ONU of cluster  $W_{\text{Max}}^1$  will be determined by Eq. (4).

$$W_{\text{Cluster}}^{\text{UW}_1} = W_{\text{Cluster}}^1 + W_B^1, \quad (3)$$

$$W_{\text{Max}}^1 = W_{\text{Cluster}}^{\text{UW}_1} / n, \quad (4)$$

where  $W_{\text{Cluster}}^{\text{UW}_1}$  is the updated window size for cluster<sup>1</sup> and  $n$  is the number of ONUs in cluster<sup>1</sup>.

So, maximum window size in CPBB scheme of a CP system is greater than that of the CP system of without CPBB scheme, i.e.,  $W_{\text{Cluster}}^{\text{UW}_1} / n > W_{\text{Cluster}}^1 / n$ . As a result, OLT will comply more bandwidth demand of the exhausted ONUs of cluster<sup>i</sup>. Finally, the OLT allocates the granted window for an ONU  $j$  of cluster<sup>i</sup>.

$$W_G^{1,j} = \begin{cases} W_R^{1,j} & \text{if } W_R^{1,j} \leq W_{\text{Max}}^1 \\ W_{\text{Max}}^1 & \text{if } W_R^{1,j} > W_{\text{Max}}^1 \end{cases}, \quad (5)$$

where  $W_R^{1,j}$  is the requested window size of an ONU  $j$  of cluster<sup>1</sup> and  $W_G^{1,j}$  is the granted window size for an ONU  $j$  of cluster<sup>1</sup>.

At the point where the cluster<sup>1</sup> finished its updated window, the OLT starts the calculation for remaining window size of cluster<sup>2</sup> which is equal to  $W_{\text{Cluster}}^2 - W_B^1$ . After that the OLT will check the sum of the requested window sizes of all the ONUs of cluster<sup>2</sup>. Whether the sum of the requested window sizes of the ONUs of cluster<sup>2</sup> is greater than the  $W_{\text{Cluster}}^2 - W_B^1$  or not. In Fig. 1, for cluster<sup>2</sup>, the sum of the requested window is again greater than the  $W_{\text{Cluster}}^2 - W_B^1$ , i.e.,  $W_R^2 > (W_{\text{Cluster}}^2 - W_B^1)$ . As a result, the OLT needs to borrow BW  $W_B^2$  from  $W_{\text{Cluster}}^3$ . Now, the updated window size of the cluster<sup>2</sup>, i.e.,  $W_{\text{Cluster}}^{\text{UW}_2}$  can be calculated as below:

$$W_{\text{Cluster}}^{\text{UW}_2} = (W_{\text{Cluster}}^2 - W_B^1) + W_B^2 \quad (6)$$

Similar to Eqs. (4) and (5), the OLT again calculates the maximum window size, i.e.,  $W_{\text{Max}}^2$ , and the granted window sizes, i.e.,  $W_G^{2,j}$ , for the ONUs of cluster<sup>2</sup>. Then the window size for the cluster<sup>3</sup> is updated as Eq. (7),

$$W_{\text{Cluster}}^{\text{UW}_3} = (W_{\text{Cluster}}^3 - W_B^2) \quad (7)$$

In the case of cluster 3, the sum of the requested windows of the ONUs of cluster<sup>3</sup> is less than the  $W_R^3 < W_{\text{Cluster}}^3 - W_B^2$ . As a result, cluster<sup>3</sup> does not need any BW borrowing from the cluster<sup>4</sup>. Moreover, there is some surplus window ( $W_S^3$ ) which can be utilized in cluster<sup>4</sup>. Thus, the updated window size for cluster<sup>4</sup> will be increased by the surplus window from cluster<sup>3</sup> and can be calculated as Eq. (8).

$$W_{\text{Cluster}}^{\text{UW}_4} = W_{\text{Cluster}}^4 + W_S^3 \quad (8)$$



For the last cluster<sup>4</sup>, the OLT accordingly calculates the maximum and granted window sizes for the ONUs of cluster<sup>4</sup>. So, there will be no unused time window in a time cycle and the utilization of the time cycle will be more efficient. In such types of specific case where the initials clusters are heavily loaded compared to the last cluster then the proposed CPBB scheme will ensure better QoSs rather existing DBA scheme of the CP system.

### 3 Simulation Results

In this section, computer simulation results were generated to compare the proposed CPBB scheme to the existing DBA algorithm of CP system. For both the proposed and existing schemes, simulation results for the GTRR, buffering delay, and fairness parameters are obtained. In the simulation setup, the OLT connected with 16 and 32 ONUs in a tree topology and ONUs were grouped in 4 clusters. The OLT is linked to the ONUs through optical fiber via a splitter, with an average distance of 10–20 km between them. The data transfer rates between the ONUs and the OLT was assumed to be 1 Gbps, while the connection speeds between the ONU and the users were assumed to be 100 Mbps. Each data packet was considered to be 1500 bytes long. For each ONU, a buffer size of 10 MB was considered. The other simulation parameters that were considered during simulation are shown in Table 1.

To obtain the performance parameters the initial clusters was considered highly loaded. When any cluster prior to last cluster are heavily loaded, simulation results show that our CPBB system outperforms the existing DBA scheme of CP system.

**Table 1** Simulation parameters [8]

| Symbols     | Description                       | Values       |
|-------------|-----------------------------------|--------------|
| Eo          | Length of Ethernet overhead       | 576 bits     |
| R           | Length of the report message      | 304 bits     |
| $G_t$       | Guard time interval               | 5 $\mu$ s    |
| D           | Packet length                     | 12 $\mu$ s   |
| $T_{cycle}$ | Length of cycle time              | 2000 $\mu$ s |
| N           | Number of ONUs                    | 16 and 32    |
| C           | Number of clusters                | 4            |
| $t_{win}$   | Predetermined cluster window size | 500 $\mu$ s  |

### 3.1 GTRR Analysis

The GTRR is characterized as the ratio of sum of all granted windows to the sum of all requested windows in a time cycle as presented in Eq. (9) [9]. It defines the closeness between the granted bandwidth in comparison to the requested bandwidth. Larger value of GTRR means more requested windows are served that ensures the lower buffering delay and greater fairness.

$$\text{GTRR} = \frac{\sum_{i=1}^c \sum_{j=1}^n W_G^{i,j}}{\sum_{i=1}^c \sum_{j=1}^n W_R^{i,j}} \quad (9)$$

The GTRR comparison among the proposed CPBB scheme and the existing DBA scheme of CP system for 16 and 32 ONUs for 4 clusters is shown in Fig. 2. For the existing 16 ONU condition, for the normalized load up to 0.5, the GTRR value remains 1 whereas in the case of our proposed method it continues up to normalized load up to 0.7. In our proposed scheme, for the normalized load from 0.8 to 0.9 the GTRR value increases on average 7.5% and for the maximum offered load (the normalized load is 1) the GTRR value increases nearly 12% compared to existing scheme. So, from these little observations we can say that the scheme that we have proposed is more efficient in case of granting requested windows at high load condition.

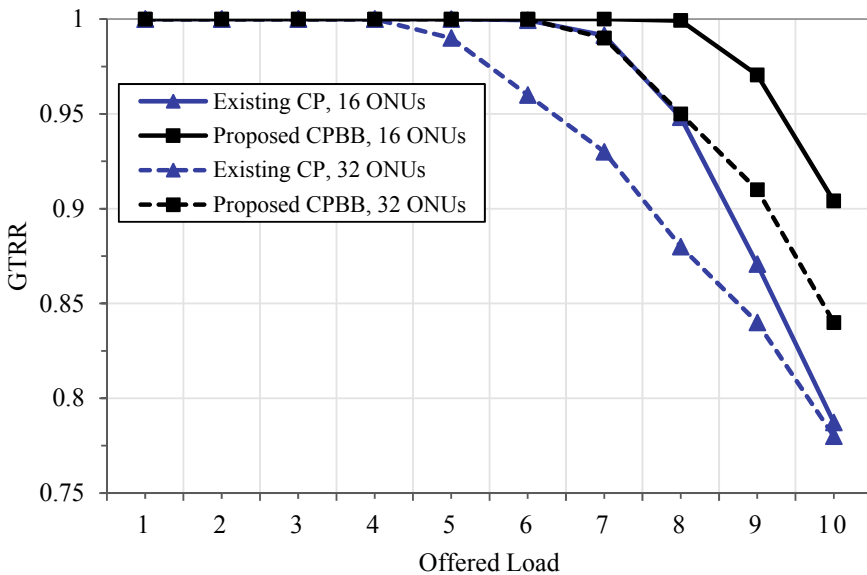


**Fig. 2** Comparison of GTRR between the proposed and existing CP with 16 and 32 ONUs

In case of 32 ONU system, it is noticeable that for the normalized offered load of 0.1–0.6 the GTRR value for 32 ONUs is 1 for our proposed CPBB scheme while, in the existing scheme, the value of GTRR is 1 for normalized offered load from 0.1 to 0.4. 99% requests are granted in the case of the existing method for the 0.5 normalized load condition. While, the proposed method, in the case of more normalized load 0.7 the same condition occurs. Even, 91% request are granted when the normalized load is 0.9 and 95% requests are granted when the normalized load it 0.8. That means CPBB scheme is performing definitely better in this case of high load and even in case of low load also.

It can be concluded like, the CPBB technique for 16 ONUs is doing better in case of the performance parameter GTRR than the 32 ONUs system. So, if the number of ONUs are increasing in a PON system, for our proposed method, the system will be gradually degrading in the case of performance parameter GTRR.

### 3.2 Buffering Delay Analysis

The buffer delay is the time it takes between the time when a packet reaches in the buffer and when it is transmitted. It is good for a system when the buffering delay is low, apparently this condition occurs when the GTRR value is high.

Figure 3 depicts a buffering delay comparison between the proposed CPBB scheme and the existing DBA scheme of CP system for 16 and 32 ONUs. For the 16 ONU system, the buffering delay is remaining close in both the proposed and existing schemes in light load conditions (up to 50–60% load). In the case of a normalized offered load of 0.7–0.9, in the CPBB scheme, buffering delay is reduced by around 30–35% and in the worst case (for 100% load), the buffering delay is reduced to nearly 28% compared to the existing scheme.

In the light load conditions (up to 50% load), for the 32 ONU system, the buffering delay remains closed in both the proposed and existing schemes. In the case of a normalized offered load of 0.6 to 1, we can observe that CPBB scheme provides about 20–30% lower buffering delay compared to the existing scheme.

If we now compare our result for the 32 ONU system with the 16 ONU system, the buffering delay is high in PON with larger number of ONUs in the case of our proposed method still it is lower than the existing scheme.

### 3.3 Fairness Analysis

Fairness is another performance parameter by which we can evaluate the PON systems. The higher the value of fairness the better it is in the sense of application of the PON systems. The fairness of a PON system can be calculated using Eq. (10) [9].

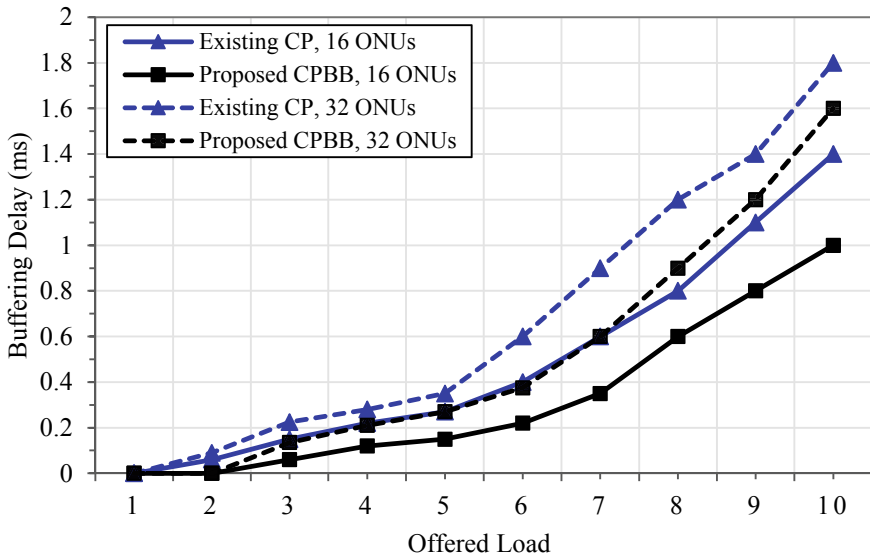


**Fig. 3** Comparison of buffering delay between the proposed and existing 4-cluster-based PON with 16 and 32 ONUs

$$F = \frac{\left\{ \sum_{i=1}^c \sum_{j=1}^n W_G^{i,j} \right\}^2}{N \left\{ \sum_{i=1}^c \sum_{j=1}^n \left( W_G^{i,j} \right)^2 \right\}} \quad (10)$$

Figure 4 shows a fairness comparison of the proposed CPBB scheme with the existing DBA scheme of CP system for 16 and 32 ONUs. For the 16 ONU PON system, fairness decreases as the load increases for both the existing and proposed schemes. In the case of a 10% load, the fairness for both the existing and proposed scheme is 1. When the load is increased from 20 to 50% the fairness is on average 95.5% for our proposed scheme but for the existing scheme, the average fairness is reduced to 94.5%. For 100% load, the fairness is 93% in the proposed scheme but for the existing scheme, the fairness was 92%. So, fairness is slightly better than the existing scheme.

For the PON system with 32 ONUs, when the load is low, as in the case of 10% normalized load, the fairness for both the existing and proposed schemes is 100%. When the load is increased from 20 to 50%, the fairness of the proposed scheme drops from 92 to 88%, while the fairness of the existing scheme drops from nearly 91 to 87%. Fairness ranges from approximately 87% for other high loads between 60 and 90% for the proposed CPBB scheme, while it ranges from 86 to 86.5% for the existing CP scheme. In the case of a 100% normalized offered load, the proposed scheme with 32 ONU has fairness of 88%, whereas the existing scheme has fairness of 86.5%.

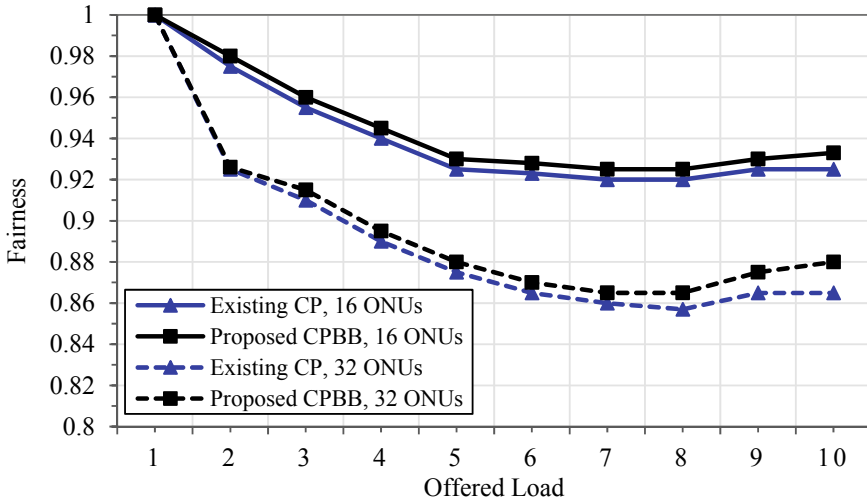


**Fig. 4** Comparison of fairness among the proposed and existing 4-cluster-based PON with 16 and 32 ONU's

## 4 Conclusion

In this paper, we have proposed a modified DBA algorithm for CP architecture using BW borrowing technique. Since the CP system holds both the benefits of offline and online polling strategies, in the proposed CPBB scheme we have just eliminated one of its limitations, which is when the initial clusters are highly loaded the existing method had no solution to it. The solution was based on the BW borrowing technique and the idea of this paper was mainly generated from the cellular system's BW borrowing schemes. We have just modified this idea and implemented it on CP system as per need. In the modified scheme, clustering is present and after clustering we are intended to detect a situation where the requested time window of a cluster is bigger than the predefined time window of a cluster. When such situation arises, OLT updates the cluster size by borrowing the required bandwidth from the consecutive next cluster. This is the core basic of our modified scheme that was proposed in this paper. Performance parameters are obtained using computer simulation which are GTRR, buffering delay and fairness. For all the cases of the performance parameters or the QoS's the proposed CPBB scheme provides better performances compared to the existing DBA algorithm of CP system.

## References

1. World Internet Users Statistics and 2021 World Population Stats. [internetworldstats.com](http://internetworldstats.com)
2. Morshed MS, Hossen M (2019) Rahman MM, Dynamic hybrid slot size bandwidth allocation algorithm for reducing packet delay and jitter variation of real time traffic in EPON. *Optik Int J Light Electron Opt* 183:523–533
3. Kramer G, Mukherjee B, Dixit S, Ye Y (2002) Supporting differentiated classes of services in Ethernet passive optical networks. *J Opt Netw* 1(8/9):280–298
4. Saha S, Hossen M, Hanawa M (2018) A new DBA algorithm for reducing delay and solving the over-granting problem of long reach PON. *Opt Switch Netw* 31:62–71
5. Helmy A, Fathallah H, Mouftah H (2012) Interleaved polling versus multi-thread polling for bandwidth allocation in long-reach PONs. *J Opt Commun Netw* 4(3)
6. Mercian A, McGarry MP, Reisslein M (2013) Offline and online multi-thread polling in long-reach PONs: a critical evaluation. *J Lightw Technol* 31(12)
7. Hossen M, Kim KD, Park Y (2013) A PON-based large sensor network and its performance analysis with Sync-LS MAC protocol. *Arab J Sci Eng* 38(8):2115–2123
8. Basu S, Hossen M, Hanawa M (2020) A new polling algorithm for dynamic data transmission sequence of cluster-based PON. In: 2020 IEEE region 10 symposium (TENSYP), Dhaka, Bangladesh
9. Basu S, Hossen M, Hanawa M (2021) Cluster-based PON with dynamic upstream data transmission sequence algorithm for improving QoS. *Opt Fiber Technol* 64

## Chapter 27

# Ensemble of Boosting Algorithms for Parkinson Disease Diagnosis



Maksuda Rahman, Md. Kamrul Hasan , Masshura Mayashir Madhurja, and Mohiuddin Ahmad

**Abstract** Parkinson's disease (PD) is a common dynamic neurodegenerative disorder due to the lack of the brain's chemical dopamine, impairing motor and nonmotor symptoms. The PD patients undergo vocal cord dysfunctions, producing speech impairment, an early and essential PD indicator. The researchers are contributing to building generic data-driven decision-making systems due to the non-availability of the medical test(s) for the early PD diagnosis. This article has provided an automatic decision-making framework for PD detection by proposing a weighted ensemble of machine learning (ML) boosting classifiers: random forest (RF), AdaBoost (AdB), and XGBoost (XGB). The introduced framework has incorporated outlier rejection (OR) and attribute selection (AS) as the recommended preprocessing. The experimental results reveal that the one-class support vector machine-based OR followed by information gain-based AS performs the best preprocessing in the aimed task. Additionally, one of the proposed ensemble models has outputted an average area under the ROC curve (AUC) of 0.972, outperforming the individual RF, AdB, and XGB classifiers with the margins of 0.5 %, 3.7 %, and 1.4 %, respectively, while the advised preprocessing is incorporated. Since the suggested system provides better PD diagnosis results, it can be a practical decision-making tool for clinicians in PD diagnosis.

**Keywords** Parkinson disease · Outlier rejection · Attribute selection · Machine learning models · Ensemble classifiers

---

<https://scholar.google.com/citations?user=36WXELIAAAJ&hl=en>.

---

M. Rahman · M. M. Madhurja

Department of Electronics and Communication Engineering (ECE), Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh

Md. K. Hasan (✉) · M. Ahmad

Department of Electrical and Electronic Engineering (EEE), Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh

e-mail: [m.k.hasan@eee.kuet.ac.bd](mailto:m.k.hasan@eee.kuet.ac.bd)

## 1 Introduction

Parkinson's disease (PD) is a continuous neuropathological degeneration of the central nervous system in which the motor functions of the human body are impaired. PD is the second most frequently found disease among neurological disorders after Alzheimer's, which has crossed more than one million people in North America alone [1]. It is delineated that 20 out of 100,000 new cases yearly are diagnosed with the incidence rates rising steeply over 50 years [2]. PD symptoms are illustrated as motor and nonmotor, where the motor symptoms include exercise disorders, bouncing, stepping problems, immobility, and postural fluctuation. In contrast, nonmotor manifestations possess mental dysfunction, perspective disease, unhappiness, anxiety, and speech and gait changes. The patients with PD worsen over time with normal bodily functions, including breathing, balance, movement, and heart function [3]. Although no medical examination is obtainable to interpret it in the early phases conclusively, diagnosing PD in an earlier stage is crucial to overcome its difficulties. However, it is contesting to distinguish PD from other neurological ailments, relying on the radiologist's experience. Therefore, a computer-aided diagnosis (CAD) system assists the radiologist in interpreting the PD in the early stage.

Many studies have been conducted, employing artificial intelligence (AI) methods to mitigate and prevent PD's vast severity [3]. In its continuity, Revett et al. [4] focused on the characteristics of PD patients, ending that voice analysis is more useful in discerning PD patients and healthy people. An audio analysis of PD patients with AVEC features had experimented with in [5]. The authors applied minimum redundancy, and maximum relevance to AVECs as a preprocessing and support vector machine (SVM) model, exhibiting that SVM with AVEC exceeded the SVM without preprocessing. Olanrewaju et al. [6] used multilayer feed-forward neural network (MLFNN) for the early PD diagnosis. Singh et al. [7] applied principle component analysis and fisher discriminant ratio for attribute selection and SVM, utilizing different features from brain MRI scan images. Das [8] compared the performance of various machine learning (ML) methods and claimed that a neural network is best for the aimed task. Lastly, feature priority, recursive feature elimination, and the SVM model were employed in [9] to detect PD patients with thirteen features. Although many articles have already been published, there is still room for performance improvement, making it more suitable for real-time appliances by reducing the attribute's dimension. However, the article's core contributions are as follows.

- Introducing a PD detection framework for employing proposed ML-based weighted ensemble models
- Integrating suitable preprocessing for outlier rejection and attribute selection
- Optimizing the hyperparameters of various ML models and suggesting a weighted ensemble of ML models that are optimized
- Conducting complete ablation studies for the preprocessing and PD classifier determination to recommend the best possible framework.



This paper is designed as follows: Sect. 2 covers the materials and methodology. Section 3 provides the experimental results with proper ablation analyses. Finally, Sect. 4 concludes the article with the future working directions.

## 2 Materials and Methodologies

This section is divided into two parts: Firstly, we explain the utilized dataset in Sect. 2.1, and secondly, the proposed approach is described in Sect. 2.2, exhibiting its all-composite algorithms.

### 2.1 Dataset

The utilized PD dataset (PDD) [10] is summarized in Table 1, defining a total of twenty-two features as  $F1 \sim F22$ . PDD consists of 195 provided vowel phonations from 31 men and women with 23 interpreted PD patients. The time since diagnosis ranged from 0 to 28 years, and the subjects' ages were 46 to 85 years, with a mean age value of 65.8 and a standard deviation of 9.8. The voice signals were recorded directly to a computer using CSL 4300B hardware, sampled at 44.1 kHz, with 16-bit resolution [10]. The attributes' mean values in the fourth column of Table 1 demonstrate that they extensively deviate in magnitudes ranging from  $-5.684$  to  $197.11$ , demanding attention like feature normalization during the structure of the PD diagnosis pipeline. Figure 1 displays projected 2D attributes employing the t-SNE method in (a) and the class distribution after projecting the high-dimensional attributes to a single dimension in (b). Both the figures visualize overlapping between the classes of the utilized PD dataset, exposing the difficulties in the PD diagnosis task with massive overlapped class boundaries. Again, Fig. 2 dispenses the attributes' outliers, showing a considerable outlier presence in  $F2$  and  $F4-F15$  attributes and mandates eliminating them to avoid unwanted learning bias toward the outlier attribute value [11]. However, the following section describes the proposed approach for the robust PD diagnosis even though various challenges exist in the chosen PDD dataset.

### 2.2 Proposed Framework

Figure 3 demonstrates the overall framework of the recommended PD diagnosis scheme.

The essential preprocessing, which consists of outlier rejection (OR) and attribute selection (AS), is followed by  $K (= 5)$  fold cross-validation (KCV). The training samples of KCV are utilized for the ML models' hyperparameter optimization and

**Table 1** Brief description utilized PD dataset with attribute's notations and mean  $\pm$  standard deviation (std). Details of this dataset can be found in [10]

| S. No. | Attributes                  | Short description                                              | Mean $\pm$ std      |
|--------|-----------------------------|----------------------------------------------------------------|---------------------|
| 1      | MDVP:Fo (Hz) ( $F1$ )       | Average vocal fundamental frequency                            | $154.23 \pm 41.4$   |
| 2      | MDVP:Fhi (Hz) ( $F2$ )      | Maximum vocal fundamental frequency                            | $197.11 \pm 91.5$   |
| 3      | MDVP:Flo (Hz) ( $F3$ )      | Minimum vocal fundamental frequency                            | $116.33 \pm 43.5$   |
| 4      | MDVP:Jitter (%) ( $F4$ )    | Inconsistency of period-to-period                              | $0.006 \pm 0.005$   |
| 5      | MDVP:Jitter (Abs) ( $F5$ )  | Fundamental frequency variation in cycle-to-cycle              | $4.4e-5 \pm 3.5e-5$ |
| 6      | MDVP:RAP ( $F6$ )           | Comparative perturbation                                       | $0.003 \pm 0.003$   |
| 7      | MDVP:PPQ ( $F7$ )           | Perturbation quotient of period in 5-point                     | $0.004 \pm 0.003$   |
| 8      | Jitter:DDP ( $F8$ )         | Average absolute differences between cycle                     | $0.010 \pm 0.009$   |
| 9      | MDVP:Shimmer ( $F9$ )       | Amplitude of peak-to-peak (local shimmer)                      | $0.030 \pm 0.019$   |
| 10     | MDVP:Shimmer (dB) ( $F10$ ) | Amplitude of peak-to-peak in terms of decibels                 | $0.282 \pm 0.195$   |
| 11     | Shimmer:APQ3 ( $F11$ )      | Quotient of amplitude perturbation in 3-point                  | $0.016 \pm 0.010$   |
| 12     | Shimmer:APQ5 ( $F12$ )      | Quotient of amplitude perturbation in 5-point                  | $0.018 \pm 0.012$   |
| 13     | MDVP:APQ ( $F13$ )          | Quotient of amplitude perturbation in 11-point                 | $0.025 \pm 0.017$   |
| 14     | Shimmer:DDA ( $F14$ )       | Absolute differences among the amplitudes of following periods | $0.047 \pm 0.031$   |
| 15     | NHR ( $F15$ )               | Ratio of noise-to-harmonic                                     | $0.025 \pm 0.041$   |
| 16     | HNR ( $F16$ )               | Ratio of harmonic-to-noise                                     | $21.89 \pm 4.43$    |
| 17     | RPDE ( $F17$ )              | Measure of nonlinear dynamical complexity                      | $0.499 \pm 0.104$   |
| 18     | DFA ( $F18$ )               | Detrended fluctuation analysis based on casual walk            | $0.718 \pm 0.055$   |
| 19     | Spread1 ( $F19$ )           | Nonlinear measure of fundamental frequency variation           | $-5.684 \pm 1.09$   |
| 20     | Spread2 ( $F20$ )           | Nonlinear measure of fundamental frequency variation           | $0.227 \pm 0.083$   |
| 21     | D2 ( $F21$ )                | Measure of nonlinear dynamical complexity                      | $2.382 \pm 0.383$   |
| 22     | PPE ( $F22$ )               | Nonlinear measure of fundamental frequency variation           | $0.207 \pm 0.090$   |

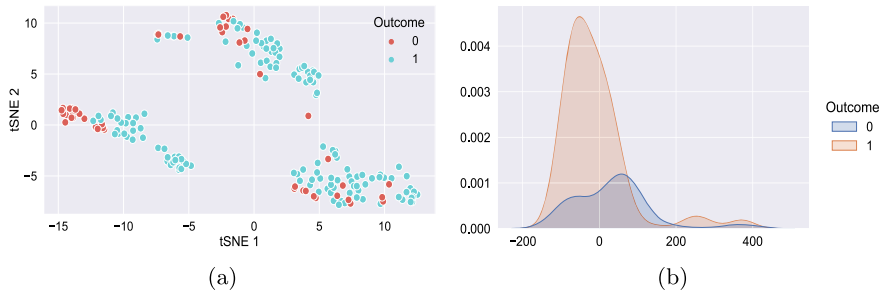


**Fig. 1** Two different visualizations of the utilized PD dataset for demonstrating the overlapping between the classes (**0**: PD negative and **1**: PD positive), where **a** exhibits projected 2D features using the t-SNE method and **b** displays the class distribution after projecting the high-dimensional features to a single dimension

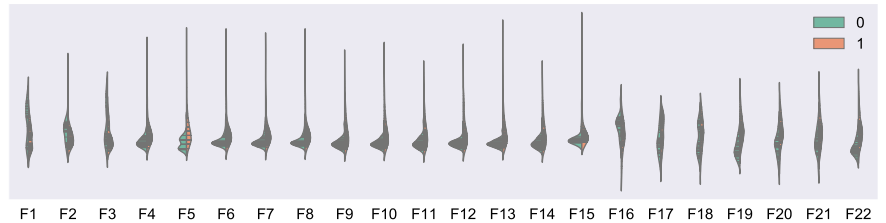


**Fig. 2** The violin plots of all the attributes in the used PD dataset, presenting their entire class (**0**: PD negative (green) and **1**: PD positive (orange)) distribution. The breadth of each curve resembles the relative frequency of data points in every region

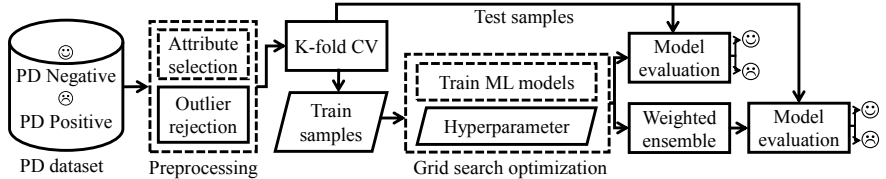


**Fig. 3** The recommended PD diagnosis framework integrating OR, AS, grid-search parameter optimization, and the proposed weighted ensemble model

their training. Individually trained models are then evaluated using the test set of the KCV and are ensemble according to the proposed weighed schemes to enhance the PD detection results.

**Proposed Preprocessing.** The suggested preprocessing is the essential step in the proposed PD detection framework, which subsequently includes OR and AS steps—multiple OR and AS methods are incorporated into the framework to determine the best-performing approaches.

The outlier in any attribute is a considerably deflected sample from other observations of that attribute (see Fig. 2), leading to a misled training process resulting in longer training times, less accurate models, and ultimately poorer results. It must be rejected from data distribution due to the sensitivity of ML models to the data range and its population distribution [12]. However, in this article, the employed ORs are based on different algorithms like Z-score (ZS-OR), percentile (P-OR), standard deviation (S-OR), interquartile range (IQR-OR), isolation forest (IF-OR), and one-class SVM (OSVM-OR) to conduct the complete ablation studies on the aimed task and utilized PDD dataset. Again, AS is essential in ML primarily because it can determine the most efficient and effective features for a given ML model system. Reducing the number of input variables is desirable to reduce the computational cost and, in some cases, improve the model's performance [12]. The recommended framework has comprised two commonly applied AS algorithms for the ablation studies, such as information gain (IG) and fisher score (FS), for executing the complete ablation studies on the aspired task and used dataset.

**Proposed Classifier.** Different ML models like Gaussian Naive Bayes (GNB), random forest (RF), AdaBoost (AdB), and XGBoost (XGB) are firstly trained and evaluated in our experimental protocol (see Sect. 3). Various crucial hyperparameters of those ML models are extensively optimized using a grid-search algorithm [13]. The ensemble of various ML models is a well-known strategy to extend the performance of any automated decision-making system [12]. In the proposed ensemble classifier, the outputs from  $N$  different models ( $Y_j (j \forall N) \in \mathcal{R}^C$ ) are aggregated, where  $C = 2$  indicates the class numbers, either having PD or not, with confidence values of  $P_i \in \mathcal{R}$ ,  $i \forall C$ , to the unseen test data ( $P_i \in [0, 1]$ ). We have recommended a weighted aggregation in the proposed ensemble model employing the Eq. (1).

$$P_i^{\text{ens}} = \frac{\sum_{j=1}^N (W_j \times P_{ij})}{\sum_{i=1}^C \sum_{j=1}^N (W_j \times P_{ij})}, \quad (1)$$

where the weights  $W_j$  are the corresponding AUCs of  $j$ th ML model. AUC is chosen as poundage for the suggested ensemble model as it is an unbiased metric and independent of class imbalance, as in the utilized PDD dataset. However, the output of the ensemble model  $Y \in \mathcal{R}^C$  has the confidence values  $P_i^{\text{ens}} \in [0, 1]$ . Finally, the ultimate class label of the unseen data ( $X \in \mathcal{R}^n$ ) will be  $C_i$  if  $P_i^{\text{ens}} = \max(Y(X))$ .

### 3 Results and Discussion

The experiments are carried out in a *Windows-10* machine, operating various Python APIs. The utilized computer has the following hardware configuration: Intel® Core™ i7-7700 HQ CPU @ 2.80 GHz processor with Install memory (RAM): 16.0 GB and GeForce GTX 1060 GPU with 6 GB GDDR5 memory. The models are evaluated in

terms of sensitivity (Sn), specificity (Sp), accuracy (Acc), and area under the ROC curve (AUC) [14]. The test’s sensitivity contemplates the probability that the given test will be positive among diseased people. In contrast, the test’s specificity mirrors the probability that the screening test will be negative among those who, in fact, do not have the disease. Accuracy ensures the ratio of the correctly identified samples to all the samples. Lastly, AUC illustrates the degree of separability measurement, demonstrating how much the designed model can distinguish between target classes.

3.1 Outlier Rejection

This section manifests the ablation studies for OR scheme selection via indirect assessment of all the employed methods. Figure 4 explicates the average recalls produced by all the OR methods, proving that the presence of OR generates poor PD detection outcomes.

The presence of the outlier introduces the skewness and kurtosis in the attribute’s distribution, which tends to underestimate and overestimate the expected outcome, respectively, as reflected in the first bar in Fig. 4. Therefore, the rejection of outliers successfully improved the PD detection results, as displayed in Fig. 4. It is noteworthy that the experimental results in Fig. 4 proclaim that the OSVM-OR method outperforms all other OR methods concerning the recall value. The best-performing OSVM-OR technique captures characteristics of training normal instances to distinguish between them and outliers, which leads to OSVM-OR being superior when comparing the utilization of Z-score, percentile, standard deviation, and interquartile range-based approaches.

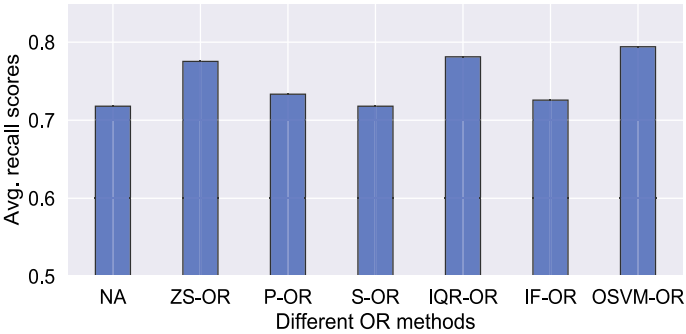


Fig. 4 Average precisions for indirect judgment of different OR methods (see details OR techniques’ acronyms in Sect. 2.2)

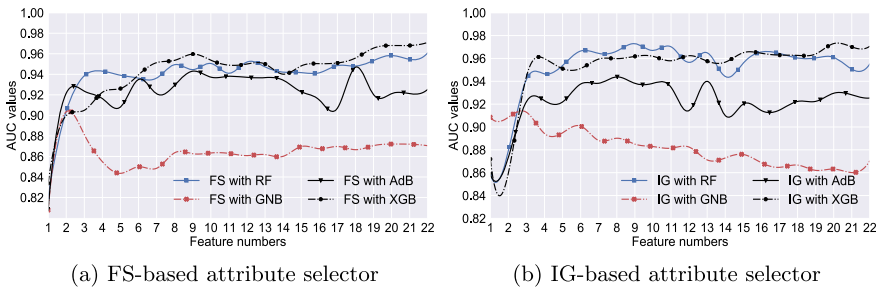


**Fig. 5** AUC versus features numbers, examining two distinct attribute selection methods and four separate ML-based classifiers

### 3.2 Attribute Selection

This section demonstrates the ablation studies for AS schemes in terms of indirect evaluation of all the applied processes, as illustrated in Fig. 5.

The AUC values are conferred for different feature numbers for two AS methods in each subfigure, employing four ML-based diagnosis models for indirect assessment. It is observed from Fig. 5a that the FS-based AS with XGB has a higher AUC value of approximately 0.96 while top-9 attributes are selected. Although XGB can output greater than 0.96 AUC value for FS-based AS, it applies more than 20 features. Additionally, it is also noticeable from Fig. 5a that other classifiers expose a similar pattern as XGB when top-9 attributes are selected while displaying slightly better AUC with the increased attributes (see Fig. 5a). On the other hand, Fig. 5b indicates that the highest value of AUC (approximately 0.97) from all the experiments is attained at top-9 attributes while the IG-based AS approach is applied. Therefore, the IG-based AS method with nine attributes is applied in the following experiments to develop an automated PD detection system.

### 3.3 PD Diagnosis

The comprehensive experimental results to design an automated PD detection system are discussed in this section. Table 2 exposes the results obtained from different single ML classifiers and four variants of our proposed ensemble classifiers, incorporating OR, AS, and hyperparameter tuning.

The first four rows of Table 2 demonstrate the results for four individual classifiers, conferring the highest possible accuracy of 0.903 from RF and XGB models. Remarkably, the GNB model produces PD detection results with a higher false-negative rate (type-II error) of 26.7 %, leading to an abysmal performance in medical diagnostic paradigms. In contrast, the other three methods, such as RF, AdB, and XGB, have fewer type-II errors with better AUC and Acc values, especially the RF model per-

**Table 2** PD diagnosis results were obtained from four individual ML models and proposed weighted ensemble models, including OR, AS, and hyperparameter tuning. The best results are denoted by the bold fonts, and the italic fonts indicate the second-best outcomes

| Classifiers      | Sn $\uparrow$                       | Sp $\uparrow$                       | Acc $\uparrow$                      | AUC $\uparrow$                      |
|------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| GNB              | 0.733 $\pm$ 0.040                   | <b>0.933 <math>\pm</math> 0.133</b> | 0.783 $\pm$ 0.014                   | 0.894 $\pm$ 0.050                   |
| RF               | 0.962 $\pm$ 0.042                   | 0.725 $\pm$ 0.170                   | 0.903 $\pm$ 0.039                   | 0.967 $\pm$ 0.034                   |
| AdB              | 0.947 $\pm$ 0.018                   | 0.706 $\pm$ 0.192                   | 0.886 $\pm$ 0.044                   | 0.935 $\pm$ 0.035                   |
| XGB              | <i>0.969 <math>\pm</math> 0.045</i> | 0.700 $\pm$ 0.247                   | 0.903 $\pm$ 0.046                   | 0.958 $\pm$ 0.045                   |
| ENS-1 (Proposed) | 0.954 $\pm$ 0.038                   | 0.728 $\pm$ 0.149                   | 0.897 $\pm$ 0.039                   | 0.970 $\pm$ 0.032                   |
| ENS-2 (Proposed) | <i>0.969 <math>\pm</math> 0.045</i> | <i>0.747 <math>\pm</math> 0.241</i> | <b>0.914 <math>\pm</math> 0.048</b> | <i>0.971 <math>\pm</math> 0.032</i> |
| ENS-3 (Proposed) | 0.962 $\pm$ 0.042                   | <i>0.747 <math>\pm</math> 0.241</i> | 0.909 $\pm$ 0.049                   | 0.962 $\pm$ 0.048                   |
| ENS-4 (Proposed) | <b>0.970 <math>\pm</math> 0.045</b> | <i>0.747 <math>\pm</math> 0.241</i> | <b>0.914 <math>\pm</math> 0.054</b> | <b>0.972 <math>\pm</math> 0.037</b> |

**ENS-1:**  $W_{\text{RF}} \times \text{RF} + W_{\text{AdB}} \times \text{AdB}$ ; **ENS-2:**  $W_{\text{RF}} \times \text{RF} + W_{\text{XGB}} \times \text{XGB}$ ; **ENS-3:**  $W_{\text{XGB}} \times \text{XGB} + W_{\text{AdB}} \times \text{AdB}$ ; **ENS-4:**  $W_{\text{RF}} \times \text{RF} + W_{\text{AdB}} \times \text{AdB} + W_{\text{XGB}} \times \text{XGB}$ .

forms best as a single classifier that provides an AUC of 0.967. Hence, RF, AdB, and XGB models are considered for the ensemble in the proposed scheme. We develop four variants of the ensemble model for ablation studies: ENS-1 (RF+AdB), ENS-2 (AdB+XGB), ENS-3 (RF+XGB), and ENS-4 (RF+AdB+XGB).

The last four rows of Table 2 demonstrate the results for four proposed ensemble classifiers (ENS-1 to ENS-4) that used the AUC of each ensemble candidate classifier as a weight in the ensemble models. A close observation in Table 2 reveals that the proposed ENS-4 outperforms all the remaining classifiers in terms of three metrics like Sn, Acc, and AUC while performing a second-best classifier in terms of Sp. Remarkably, the ENS-4 model produces PD detection results with a less false-negative rate (type-II error) of 3.0 %. Again, it also points to be noted that the ENS-4 has also exceeded the baseline single best RF model for all the metrics with the margins of 0.8, 2.2, 1.1, and 0.5 % concerning the Sn, Sp, Acc, and AUC, respectively.

This paragraph depicts the statistical ANOVA test to validate the experimentations, exhibiting the fivefold CV results concerning the PD diagnosis accuracy in the Box and Whisker plot in Fig. 6.

We apply  $\alpha = 0.05$  as a threshold to reject the void supposition if the  $p$ -value  $\leq 0.05$ , yielding effective outcomes. The ANOVA test confirms a  $p$ -value of 0.00214 ( $\leq 0.05$ ), announcing that an alternative hypothesis is obtained, vigorously pointing out that none of the means are equivalent (also portrayed in Fig. 6). Additionally, a post-hoc T-test is included with the ANOVA test to determine the better PD detection model from all eight different models, confirming the dominance of the proposed weighted ensemble model that is ENS-4 ( $W_{\text{RF}} \times \text{RF} + W_{\text{AdB}} \times \text{AdB} + W_{\text{XGB}} \times \text{XGB}$ ). Although the median accuracies of RF, ENS-1, ENS-2, and ENS-3 are slightly higher than the median accuracy of ENS-4, it has a better mean of accuracy with a significantly lower interquartile range, meaning ENS-4's inter-fold

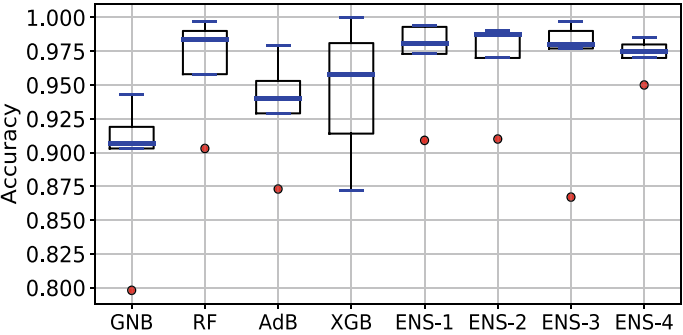


**Fig. 6** Box and whisker plots of AUC results acquired from fivefold cross-validation on different ML-based and ensemble classifiers, where M-1 to M-8 represent GNB, RF, AdB, XGB, ENS-1, ENS-2, ENS-3, and ENS-4, respectively

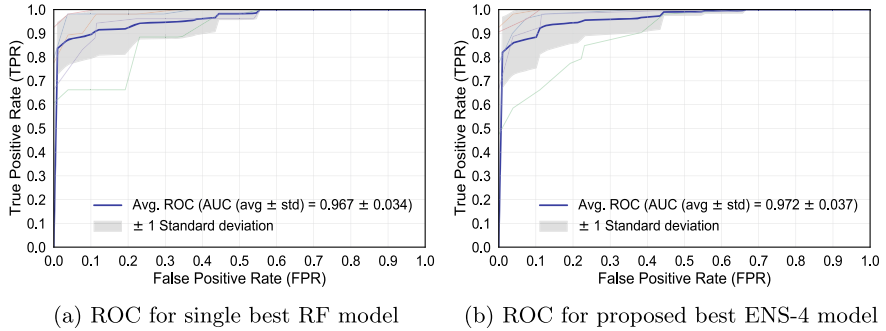


**Fig. 7** The ROC curves for the single best ML model and proposed best ensemble model for fivefold cross-validation, conferring the average and each fold ROC with their corresponding AUC values

deviation is less than those of three models. Those discussions substantially prove the superiority of the proposed ENS-4 for the PD diagnosis.

Furthermore, single best RF and proposed ensemble ENS-4 models are again compared and contrasted in the ROC curves in Fig. 7.

The obtained ROC curves indicate that the suggested pipeline commits a mean AUC value of 0.972 with a deviation of 0.037, while the average AUC for the RF model is 0.967. Again, this result reveals our model’s robustness with remarkably rarer inter-fold deviations.

### 3.4 State-of-the-Art Comparison

This section compares and contrasts the experimental results in Table 3 on the same PDD dataset (see Sect. 2.1).



**Table 3** State-of-the-art comparison with recent works for the same PDD datasets, where NSF means the number of selected features

| Articles                     | OR      | AS    | NSF | Classifier | AUC   |
|------------------------------|---------|-------|-----|------------|-------|
| Wroge et al. [5]             | N/A     | AVECs | 11  | SVM        | 0.924 |
| Bhattacharya and Bhatia [15] | N/A     | N/A   | 22  | SVM        | 0.632 |
| Sriram et al. [16]           | N/A     | N/A   | 22  | SVM        | 0.889 |
| Olanrewaju et al. [6]        | N/A     | N/A   | 22  | MLFNN      | 0.950 |
| Proposed                     | OSVM-OR | IG-AS | 9   | ENS-4      | 0.972 |

**ENS-4:** Weighted ensemble of RF, AdB, and XGB

Many articles [5, 6, 15, 16] have already been published for the same task on the PDD dataset. Table 3 reports the value of an AUC of the corresponding article for quantitative comparison along with their OR and AS methods and the number of selected features (NSF). Comparing them in terms of AUC, our suggested framework has outputted better results than the works of Wroge et al. [5], Bhattacharya and Bhatia [15], Sriram et al. [16], and Olanrewaju et al. [6] with the margins of 4.8 %, 34.0 %, 8.3 %, and 2.2 %, respectively. Those significant rises in the AUC value are probably due to our pipeline’s appropriate OR and AS techniques appliance. It is also substantial that our pipeline has utilized only nine attributes, whereas [6, 15, 16] applied all the twenty-two attributes. The utilization of many features brings the curse of dimensionality, and attribute-space becomes sparser, pushing the ML classifiers to be overfitted by lowering generalizing [12]. Besides, assembling classifiers from datasets with numerous characteristics is more computationally mandating [13]. Although [5] applied our nearest eleven features, they have outputted 4.8 % less value of AUC, which is unexpected. In contrast, our proposed system has employed a minimum number of attributes with the improved PD detection results, making it more applicable in real-time applications.

## 4 Conclusion and Future Works

This article has automated the PD detection task for the publicly available PDD dataset by proposing a robust framework consisting of preprocessing and the proposed weighted ensemble model. The proper preprocessing, especially OR and AS methods, substantially enhances the PD detection results. The ML models have exposed their maximum performances when their hyperparameters are optimized. Additionally, the testing consequences have highlighted the capability of the suggested weighted ensemble model over the single ML models like GNB, RF, AdB, and XGB as it weighted totalities of the output likelihoods of the ensemble contenders ML models. Again, corresponding to other examinations, our research delivers a better authentic model employing nine attributes, i.e.,  $F_{22}$ ,  $F_{19}$ ,  $F_1$ ,  $F_3$ ,  $F_{13}$ ,  $F_{20}$ ,

*F12*, *F11*, and *F5* (high to low attribute significance), which are easily attainable. Since the recommended system delivers better PD detection outcomes, it can be a helpful decision-making instrument for clinicians in PD diagnosis. In the future, the presented framework will also be engaged in the other detection issues to evaluate its effectiveness and versatility.

## References

1. Olanow CW, Stern MB, Sethi K (2009) The scientific and clinical basis for the treatment of Parkinson disease (2009). *Neurology* 72:S1–S136
2. Tsanas A, Little MA, McSharry PE, Ramig LO (2010) Enhanced classical dysphonia measures and sparse regression for telemonitoring of Parkinson's disease progression. In: 2010 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 594–597
3. Alzubaidi MS, Shah U, Dhia Zubaydi H, Dolaat K, Abd-Alrazaq AA, Ahmed A, Househ M (2021) The role of neural network for the detection of Parkinson's disease: a scoping review. In: *Healthcare*, vol 9. Multidisciplinary Digital Publishing Institute, p 740
4. Revett K, Gorunescu F, Salem ABM (2009) Feature selection in Parkinson's disease: a rough sets approach. In: 2009 International multiconference on computer science and information technology. IEEE, pp 425–428
5. Wroge TJ, Özkanca Y, Demiroglu C, Si D, Atkins DC, Ghomi RH (2018) Parkinson's disease diagnosis using machine learning and voice. In: 2018 IEEE signal processing in medicine and biology symposium (SPMB). IEEE, pp 1–7
6. Olanrewaju RF, Sahari NS, Musa AA, Hakiem N (2014) Application of neural networks in early detection and diagnosis of Parkinson's disease. In: 2014 International conference on cyber and IT service management (CITSIM). IEEE, pp 78–82
7. Singh G, Vadera M, Samavedham L, Lim EC-H (2016) Machine learning-based framework for multi-class diagnosis of neurodegenerative diseases: a study on Parkinson's disease. *IFAC-PapersOnLine* 49:990–995
8. Das R (2010) A comparison of multiple classification methods for diagnosis of Parkinson disease. *Expert Syst Appl* 37:1568–1572
9. Senturk ZK (2020) Early diagnosis of Parkinson's disease using machine learning algorithms. *Med Hypotheses* 138:109603
10. Little M, McSharry P, Hunter E, Spielman J, Ramig L (2008) Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *Nature Precedings* 1–1
11. Hasan MK, Alam MA, Roy S, Dutta A, Jawad MT, Das S (2021) Missing value imputation affects the performance of machine learning: a review and analysis of the literature (2010–2021). *Inf Med Unlocked* 100799
12. Hasan MK, Alam MA, Das D, Hossain E, Hasan M (2020) Diabetes prediction using ensemble of different machine learning classifiers. *IEEE Access* 8:76516–76531
13. Hasan MK, Jawad MT, Dutta A, Awal MA, Islam MA, Masud M, Al-Amri JF (2021) Associating measles vaccine uptake classification and its underlying factors using an ensemble of machine learning models. *IEEE Access* 9:119613–119628
14. Hasan MK, Wahid SR, Rahman F, Maliha SK, Rahman SB et al (2022) Grasp-and-lift detection from EEG signal using convolutional neural network. [arXiv:2202.06128](https://arxiv.org/abs/2202.06128)
15. Bhattacharya I, Bhatia MPS (2010) SVM classification to distinguish Parkinson disease patients. In: *Proceedings of the 1st Amrita ACM-W celebration on women in computing in India*, pp 1–6
16. Sriram TV, Rao MV, Narayana G, Kaladhar D, Vital TPR (2013) Intelligent Parkinson disease prediction using machine learning algorithms. *Int Eng Innov Technol* 3:212–215

# Chapter 28

## GLCM and HOG Feature-Based Skin Disease Detection Using Artificial Neural Network



Nymphia Nourin , Paromita Kundu , Sk. Saima ,  
and Md. Asadur Rahman

**Abstract** Skin, the vital body part, can be affected by various diseases and inflammations because of numerous known and unknown bacteria, fungi, and other microorganisms. Automated skin disease detection is a potential method to reduce the cost and improve the effectiveness of skin disease detection in the initial stage. Our proposed work presents a computer-aided skin disease detection approach using an Artificial Neural Network. This work uses an image analysis technique to classify four classes of images, i.e., eczema, hemangioma, malignant melanoma, and stasis dermatitis. It takes the digital images of the disease and does the necessary preprocessing steps. Then the GLCM and HOG features are extracted from the image. An ANN model is trained with these features. The overall accuracy of our work is 93.5% and 78.1% for GLCM and HOG features, respectively. The test result of the process is shown to the user through GUI which also recommend the treatment for the initial stage.

**Keywords** GLCM features · HOG features · Artificial neural network · Graphical user interface

## 1 Introduction

Skin is the outermost largest organ of the body that consists of the dermis, epidermis, and subcutaneous tissues. These three layers work as a protective layer to the body's internal organs. Though it works as a protective layer, it is not indestructible. Skin is constantly influenced by external and genetic factors, i.e., viruses, bacteria, allergies, fungi, etc. Exposure to Ultraviolet Light (UV) also damages the skin [1]. The symptoms of skin disease vary according to the severity of the disease. Skin diseases are mild and, in some cases, neglected. In severe cases, i.e., skin cancer, it can

---

N. Nourin · P. Kundu · Sk. Saima · Md. A. Rahman (✉)

Department of Biomedical Engineering, Military Institute of Science and Technology, Mirpur Cantonment, Dhaka 1216, Bangladesh

e-mail: [asadur@bme.mist.ac.bd](mailto:asadur@bme.mist.ac.bd)

even cause death. Skin diseases can be minor to severe, i.e., acne, eczema, psoriasis, hemangioma, vitiligo, scabies, dandruff, leprosy, mastitis, seborrheic keratosis, melanoma, basal cell carcinoma, keratinocyte Skin Cancer. Among all skin diseases, melanoma is the deadliest cancer, which occurs due to high exposure to the sun. Melanoma occurs in only 4% of skin cancers, but it is the most lethal. The deadly skin cancer melanoma has treatment if diagnosed in the early stage. But if the cancer is detected late, it grows and spreads to the others body parts [2]. Diagnosis of skin diseases is usually made via visual inspection by physicians. Some pathological tests, i.e., blood and hormone tests, identify skin diseases. It is challenging for physicians to differentiate between healthy and lesion skin through the visual inspection process. Sometimes the result can be inaccurate, which can lead to unfair and delayed treatment. Sometimes it becomes difficult to identify diseases that have similarities in appearance. Computer-Aided Diagnosis (CAD) can overcome these limitations. In computer-aided diagnosis, image processing is necessary to distinguish diseases. Image processing removes the artifacts from skin lesion images and does further modifications. Then according to the features, different classifiers identify the disorders based on these features.

Several strategies and procedures have been used to explore automatic skin disease diagnosis in the past few decades. In [3], the proposed system utilized image processing and machine learning techniques for over 1800 images to classify three types of skin diseases. In this method, a Quadratic Support Vector Machine was used as a classifier which acquired 94.74% accuracy. Using the HAM10000 dataset, seven types of skin lesions are detected in [4]. The classification model used in the proposed method was a Fully Convolutional Network (FNC) which achieved an accuracy of 98%. In [5], CNN is applied as a classification tool to detect four common types of skin disorders. The test dataset contains 1067 images. The algorithm has an accuracy of  $87.25 \pm 2.24\%$ . In our proposed method, we have used the GLCM and HOG features of the image to classify four classes of image, i.e., eczema, hemangioma, melanoma, and stasis dermatitis using an ANN model.

The remainder of the paper is written as follows. Section 2 presents the Methodology of the work. In Sect. 3, the results and discussion are presented. Finally, Sect. 4 conclusions.

## 2 Materials and Method

The automated identification approach to skin disease has been extensively employed and has made substantial progress. The application of advanced diagnostic techniques for skin disease screening has improved diagnostic reliability and reduced diagnostic procedure time [6]. We looked into the techniques to identify skin disease over the years. The skin disease research we looked at revealed a viable method for detecting skin disease. Figure 1 depicts the fundamental processes of employing a computer-aided technique to detect skin diseases in our work.

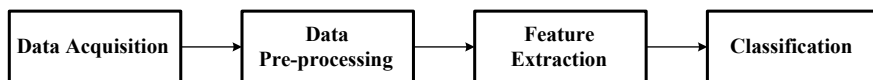


**Fig. 1** Basic steps of computer-aided skin disease detection



**Fig. 2** Skin lesion images of eczema, hemangioma, melanoma, and stasis dermatitis

## 2.1 Data Acquisition

Image acquisition is a significant step for computer-aided image classification methods because it provides the input data for the whole process. The skin images can be clinical or dermoscopic. Clinical images contain more noise than dermoscopic images because of the tools used. In this work, skin diseases in different parts of the world have been studied. From that, a database is formed with disease pictures from two separate sites: (a) Dermnet NZ [7] and (b) Science photo library [8] and all the pictures were with actual segments. These two websites are free online resources so there are no ethical issues related to data collection. We selected four different classes of skin diseases which were eczema, hemangioma, malignant melanoma, and stasis dermatitis. A total of 643 images of these four classes were collected from the sites. All of the skin disease images were in .jpg format. Some of the images contained watermarks of the respected website. This database contains both dermoscopic and clinical images (Fig. 2).

## 2.2 Data Preprocessing

Image preprocessing aims to improve image data by removing undesired flaws or enhancing specific visual features that are relevant for further processing and analysis. Image resizing, either upscaling or downscaling depending on the need for increased or decreased resolution, is an important signal processing strategy due to the variety of data sources and formats. The database that we formed contained images that were not uniform in size. In our work, we have resized all the images to  $100 \times 100$  size using the bicubic interpolation method [9]. We resized the images to get an equal number of features for all images. The process of data classification for diseases was

not significantly affected by the information lost as a result of image resizing. Then we converted the RGB image into grayscale using the weighted average method. Also, for HOG feature extraction the image is converted to binary or logical form.

2.3 Feature Extraction

Feature extraction describes to dimensionality reduction procedure of converting raw data into numerical features or variables, which are processed while conserving necessary information of the original data or observations. In this work, we have employed Gray Level Co-occurrence Matrix (GLCM) and Histogram of Oriented Gradient (HOG) methods to determine the texture features from images.

The GLCM approach of obtaining textural information from images has proven to be a popular statistical tool. The steps involved in GLCM-based feature extraction are presented by the flow diagram in Fig. 3. In our work, we have used twenty-two haralick’s features for offset [0 2] and [−2 2] calculated from the probability matrix to extract the features of texture data of images using the co-occurrence matrix [10] (Table 1).

The histogram of oriented gradients (HOG) is a feature extraction approach that clusters an image or object while deleting extraneous information to extract important characteristics. It is a structural or shape-based feature descriptor that’s commonly used in computer vision, object identification, digit classification, and image processing. The method estimates gradient orientation in localized areas of a picture and generates a histogram for each one [9]. The steps that we followed in this work to calculate HOG features are shown in Fig. 4.

Table 1 GLCM features that are extracted from images

|                                                 |                                           |
|-------------------------------------------------|-------------------------------------------|
| (i) Energy                                      | (xii) Cluster shade                       |
| (ii) Dissimilarity                              | (xiii) Cluster prominence                 |
| (iii) Inverse difference                        | (xiv) Maximum probability                 |
| (iv) Entropy                                    | (xv) Sum of squares                       |
| (v) Contrast                                    | (xvi) Sum average                         |
| (vi) Correlation                                | (xvii) Sum variance                       |
| (vii) Homogeneity                               | (xviii) Sum entropy                       |
| (viii) Autocorrelation                          | (xix) Difference variance                 |
| (ix) Information measures of correlation (1)    | (xx) Difference entropy                   |
| (x) Information measures of correlation (2)     | (xxi) Normalized inverse difference (INN) |
| (xi) Inverse difference moment normalized (IDN) | (xxii) Maximal correlation coefficient    |

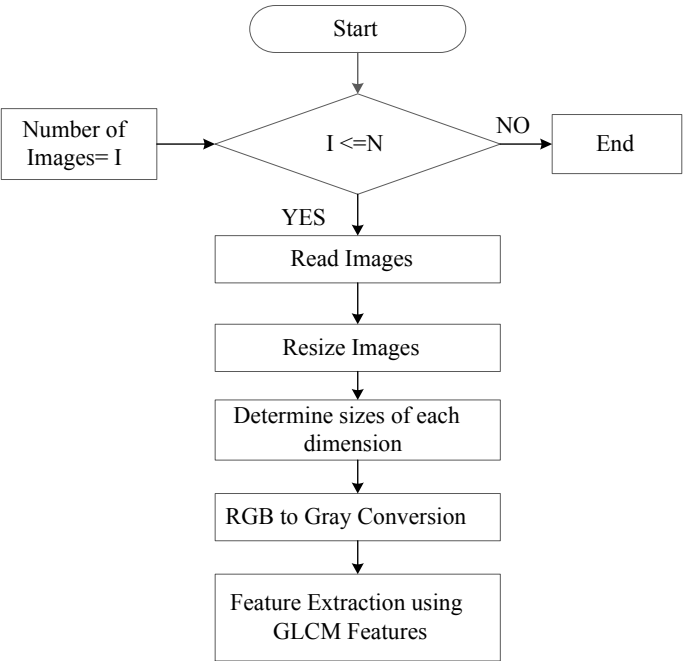


Fig. 3 Flow diagram of data processing and feature extraction using GLCM feature

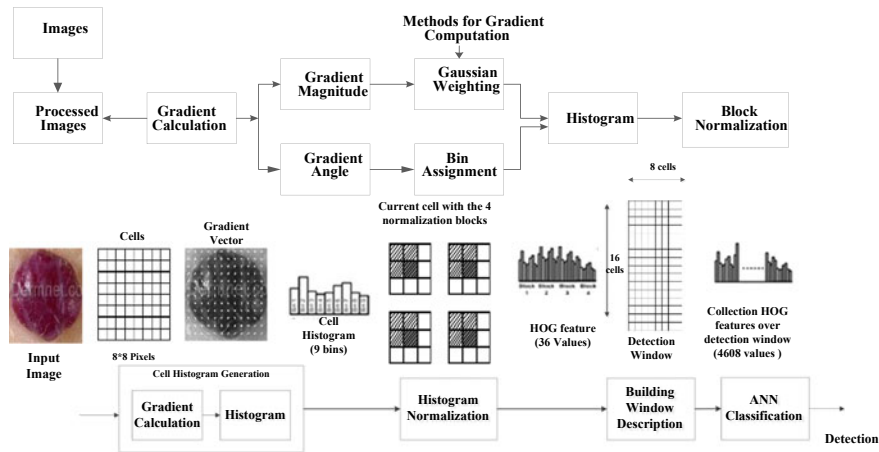


Fig. 4 Flow diagram of HOG feature calculation

2.4 Classification

Following the extraction of GLCM and HOG features from images, a classifier can be trained to categorize a test sample into one of the known classes. The ANN model can be trained with the extracted features from the training set.

3 Results and Discussion

The process is implemented in MATLAB 2018a in a computer having Intel core i5 processor (2.50 GHz) with 8 GB RAM.

In this work, we used our own created dataset that contains 643 images in total. Here, 224 images are of eczema, 60 hemangioma, 188 malignant melanoma, and 171 stasis dermatitis. Initially, all the images are resized in uniform size so that we get a uniform number of features from all the images. The images are resized to  $100 \times 100$  size. Then we have converted the RGB images to grayscale. The preprocessing steps for this work are shown in Fig. 5.

Following the preprocessing steps, we have calculated the GLCM and HOG features from the images. Then we trained the ANN model with these extracted features. The test results for each class are shown in Figs. 6 and 7 through the confusion matrix. Here, Class 1 is Malignant Melanoma, Class 2 is Stasis Dermatitis, Class 3 is Hemangioma, and Class 4 is Eczema.

Figure 6 shows the individual rate of accuracy for four skin disease classes for GLCM features. The figure shows the number of correct predictions as well as incorrect predictions for each class of diseases. From Fig. 6, the accuracy of the first column is 92.0%. This means 173 images of melanoma were correctly identified as

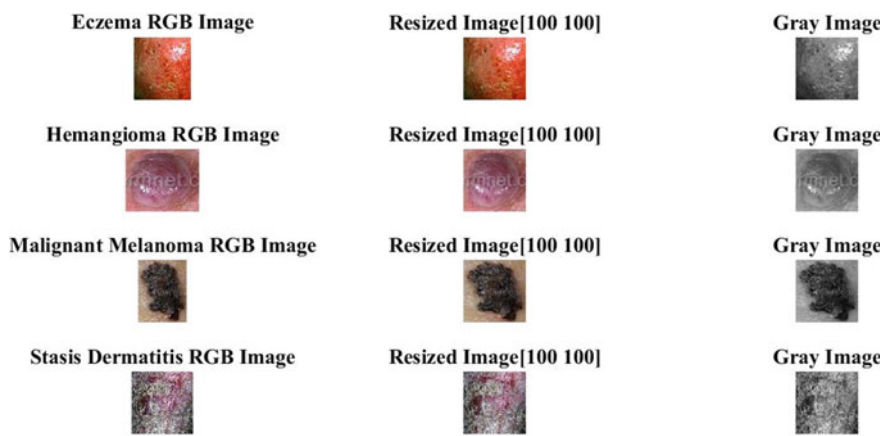


Fig. 5 Preprocessing of the images



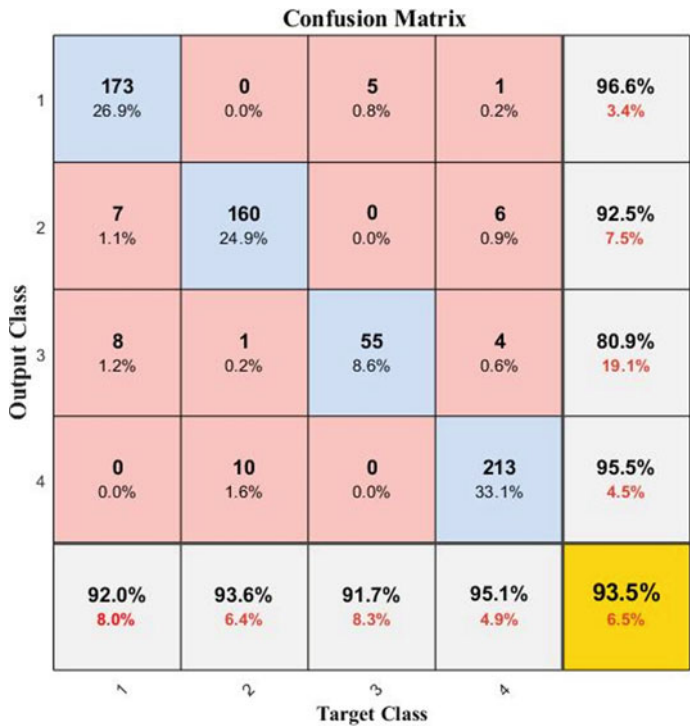


**Fig. 6** Percentage accuracy by confusion matrix for GLCM features

melanoma and 7 images of melanoma are misclassified as stasis dermatitis, and 8 as hemangioma. Similarly, the accuracy of the first row is 96.6% that is 173 images are classified as melanoma and 5 images are misclassified as hemangioma, and 1 is eczema. The diagonal of the table shows the correctly identified diseases. The total accuracy for GLCM features of the images using the ANN model is 93.5%. That means GLCM features give promising results in the image identification process. Similarly, in Fig. 7 the accuracy of the individual class using HOG features is shown through the confusion matrix. The total accuracy for HOG features of the images using the ANN model is 78.1%.

Comparing the results of GLCM and HOG features it can be seen that GLCM features give more promising results than HOG features.

Finally, we have developed a Graphical user interface shown in Fig. 8 through which an average user who does not have professional skills can initially determine his disease. The GUI also shows a recommendation Section where the user can get the idea of treatment and consult the doctor for a serious condition.

GUI takes an input image from the dataset and then pre-processes the image, i.e., Image resizing grayscale conversion, and binary conversion (for HOG feature extraction). Then it extracts the GLCM or HOG features from the image. Then it compares the feature from the trained ANN model and gives a predictive result. The

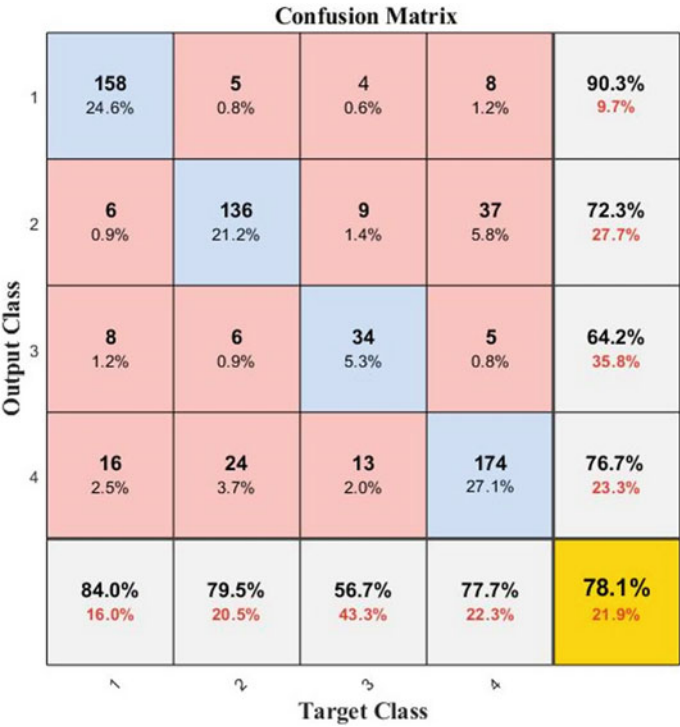


Fig. 7 Percentage accuracy by confusion matrix for HOG features

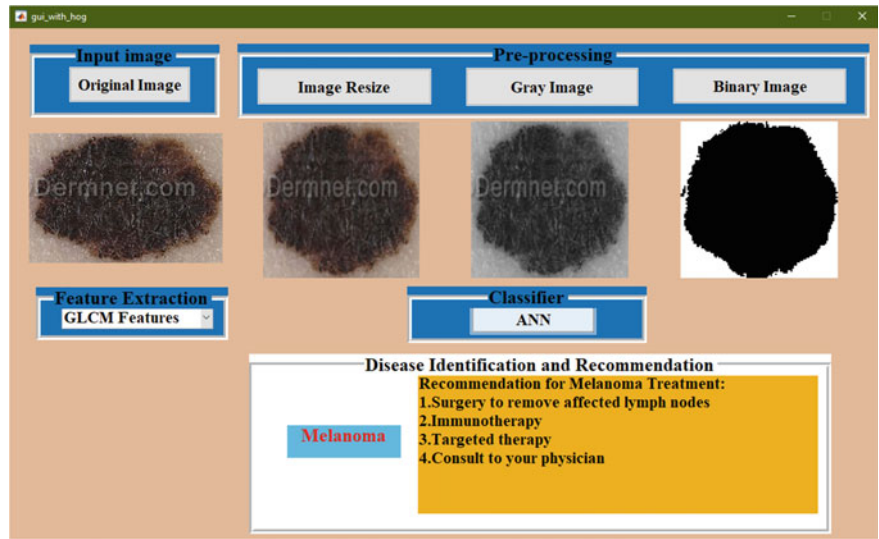


Fig. 8 The result is shown in graphical user interface (GUI) with a recommendation

classification result is shown to the user with a recommendation section from which the user gets an initial idea about the disease. Physicians can also use this system as a cross-validation process.

## 4 Conclusion

Prevention of transmission and providing the proper treatment are possible if the skin diseases are detected early and accurately. Automated computer-aided methods are effective in this case. Traditional methods of skin disease detection, i.e., lasers, pathological and hormone tests, and photonic medical technologies are expensive and time-consuming. To reduce the cost an initial assessment can be done using a computer-aided method. This process only needs a camera and a computer. Our proposed method shows a promising result to detect skin diseases. The experiment is conducted on our own created dataset of 643 total images of four classes of images. In our dataset, we have cropped the surroundings of the image and only kept the region of interest. Following the preprocessing steps, we extracted the GLCM and HOG features of the images. We have trained our ANN model with those features and finally tested the images. The overall accuracy of our classifier model is 93.5% and 78.1% for GLCM and HOG features, respectively. The results show that classification via GLCM features gives better accuracy than the HOG features. Then we developed a GUI to make the system user-friendly. For future work, we want to compare the result of our model with the prediction of the physician as a cross-validation system.

## References

1. Wei F (2018) Skin disease recognition method based on image color and texture features. *Comput Math Methods Med* 2018(5):99–110
2. Sheha MA (2012) Computational and mathematical methods in medicine automatic detection of melanoma skin cancer using texture analysis. *Int J Comput Appl* 42(20):22–26
3. Hameed N (2019) An intelligent computer-aided scheme for classifying multiple skin lesions. *Computers* 8(3):62
4. Adegun AA (2020) FCN-based DenseNet framework for automated detection and classification of skin lesions in dermoscopy images. *IEEE Access* 8:150377–150396
5. Zhang X (2018) Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge. *BMC Med Inform Decis Mak* 18(2):69–76
6. Stanganelli I (2005) Computer-aided diagnosis of melanocytic lesions. *Anticancer Res* 25(6C):4577–4582
7. DermNet NZ. <https://dermnetnz.org/>. Accessed 21 Jul 2020
8. Science Photo Library, <https://www.sciencephoto.com/>. Accessed 05 Sept 2020
9. Danahy EE (2007) Algorithms for the resizing of binary and grayscale images using a logical transform. *Int Soc Opt Photonics* 6497:64970Z

10. Kolkur S (2016) Survey of texture based feature extraction for skin disease detection. In: 2016 international conference on ICT in business industry & government (ICTBIG). IEEE, India, pp 1–6

## Chapter 29

# Subject Dependent Cognitive Load Level Classification from fNIRS Signal Using Support Vector Machine



Syeda Umme Ayman , Al Arrafuzzaman , and Md. Asadur Rahman

**Abstract** Cognitive workload, the amount of mental effort perceived by an individual while doing a task, is reflected in the metabolic activity of different parts of the brain. Being an important factor of metabolism, oxygen concentration at different portions of tissue can define the amount of undergoing metabolic activity. This work aims to study and classify the cognitive workload level of different subjects from brain hemodynamics. As functional Near-Infrared Spectroscopy (fNIRS) can analyze brain activity by observing the oxygen concentration, an fNIRS dataset from a renowned source has been considered for this work. Among the various types of data recorded from various cognitive tests, the data used for this work is based on an n-back test performed on 26 healthy individuals, containing three types of data (0-back, 2-back, 3-back). The relative concentration change of the oxygenated hemoglobin ( $\text{HbO}_2$ ) is calculated per channel corresponding to each trial and task. Using the analysis of variance (ANOVA), the change is tested to find significant inference among the three levels and the outcome shows the inconsistency of significant channels between subjects. The data were then classified into two classes, low-level workload (0-back) and high-level workload (2-back and 3-back) and using the Support Vector Machine (SVM), a model is developed to classify the 2 classes of cognitive workload with the highest accuracy of 94.44% and an average accuracy of 75.85% for 50:50 train test ratio. This work also studies the brain functionality of individuals under varying cognitive loads by generating topographic images via MATLAB.

**Keywords** Cognitive workload · Hemodynamics · fNIRS · n-back test · ANOVA · SVM

---

S. U. Ayman · A. Arrafuzzaman · Md. A. Rahman (✉)

Department of Biomedical Engineering, Military Institute of Science and Technology (MIST),  
Mirpur Cantonment, Dhaka 1216, Bangladesh

e-mail: [asadur@bme.mist.ac.bd](mailto:asadur@bme.mist.ac.bd)

## 1 Introduction

The human brain, the central processing unit, controls all human activities such as movements, mental workload, visualization, emotion, perception, senses, attention, cognitive skill, etc. Passive Brain-Computer Interfaces are systems that provide a means of communication between the user and the external devices by reading the brain signals and exploiting the information to adapt the human-machine interfaces in a useful way. For example, passive BCI can help in the improvement of road safety by measuring the mental or cognitive workload of the driver. According to the concept of cognitive workload, humans have limitations in information processing at any given time. The cognitive workload comes from the term 'Cognition', which means the psychological procedures of acquiring knowledge, understanding utilizing thought and senses; also respond perceiving all actions [1]. Cognitive workload refers to the quantity of mental effort required to complete any task [2]. For example, arithmetic operations, reading, writing, driving, problem-solving, and learning that require efficient use of working memory. Operations such as driving under stressful conditions, monitoring air traffic control, and piloting an aircraft are impactful applications of passive BCI in monitoring mental workload [3]. The optimal level of cognitive workload is very crucial in high-risk scenarios like the defense domain where important decisions are required to be made in real-time. In the field of medicine, early detection of Autism Spectrum Disorder (ASD) and Alzheimer's disease dementia and cognitive impairment also requires taking a cognitive test. In the field of research, there are several applications like the assessment of surgeons' intraoperative cognitive workload and analyzing children's cognitive workload due to the excessive use of smartphones. So the measurement of cognitive activity has become an essential part to avoid physical and mental risks and maintain efficiency and productivity in work performance.

The cognitive workload is related to brain functionality that has a close relationship with brain hemodynamics [4]. This brain function can be evaluated via changes in the concentration of oxygenated hemoglobin (HbO<sub>2</sub>) and the deoxygenated hemoglobin (dHb). There are many non-invasive functional neuroimaging techniques to determine the workings and connectivity of human brain systems like EEG, MEG, Fmri, and fNIRS based on the gathering of hemodynamic or electromagnetic signals relate to the human brain activity. From the recent literature review, it has been found that the mostly proposed BCI system to measure the cognitive workload is based on the EEG modality that provides lower accuracy in the classification [5] because of its poor spatial resolution (EEG: 5–9 cm) [6]. At the same time, the traditional methods of the evaluation of cognitive workload included subjective measures such as interviews or questionnaire-based approaches where the participants reported the amount of workload induced during the task which does not allow for real-time evaluation of cognitive workload. Even recent research does not provide any means to visualize brain functionality graphically. Therefore, passive BCI system implementation requires optimal imaging tools for the robust estimation of a person's cognitive state by enhancing classification accuracy [7].

To overcome the challenges, we propose to use the fNIRS signal to find out the neuronal activation pattern of the human brain based on mental arithmetic tasks measured by the optical method (fNIRS). The fNIRS, a hemodynamic neuroimaging modality, discovered in 1977 by Jobsis [8], uses near-infrared light (650–1000 nm) to measure the concentration changes of the HbO<sub>2</sub> and dHb [9]. It enables real-time evaluation of hemoglobin oxygenation providing a very good spatial resolution (~1–1.5 cm), moderate temporal resolution (up to 100 Hz), portability to use, cost-effectiveness, a high value of the signal-to-noise ratio (SNR), less motion artifact compared to fMRI, MEG, EEG, and PET. Using this modality proper information about human cognitive activity will be extracted, modeled, and classified for the successful passive BCI application like neuro-rehabilitation, evaluation of the effectiveness of the drug, and also in mind control devices. Eventually, the objective of this research proposal is:

- To model a pattern of the neuro-activations of cognitive tasks based on activations from prefrontal fNIRS signals.
- To develop a machine learning model for the classification of the cognitive task measured through fNIRS.
- To generate topographic images that will help to visualize the functionality of the brain of the individuals.

The remaining paper is organized as follows: Sect. 2 describes the materials and methodology. In Sect. 3, experimental results and respective discussions are provided. Finally, the work is concluded in Sect. 4.

## 2 Materials and Methods

### 2.1 Materials

For this research work, we have chosen the visual n-back task which is a classic memory test, frequently used in neuroimaging research and the most common brain stimulator which was first introduced by Kirchner in 1958 [10]. It parametrically varies working memory load and thus task difficulty, while keeping overall task procedures constant across conditions. The n-back data has been collected from an open-access dataset that provides both EEG and fNIRS data collected simultaneously during a set of cognitive tasks [11]. The data has been recorded using the modality NIRS 36 Channel and conducted by the researcher of the Department of Biomedical Engineering, Berlin Institute of Technology and this article was published in a well-known journal called SCIENTIFIC DATA.

The collected n-back dataset consists of three sessions where each session is comprised of three individual series for individual n-back tasks, i.e., 0-back, 2-back, and 3-back tasks in the following order of  $\{(3 \rightarrow 2 \rightarrow 0) \rightarrow (2 \rightarrow 0 \rightarrow 3) \rightarrow (0 \rightarrow 3 \rightarrow 2)\} \rightarrow \{(2 \rightarrow 0 \rightarrow 3) \rightarrow (0 \rightarrow 3 \rightarrow 2) \rightarrow (3 \rightarrow 2 \rightarrow 0)\} \rightarrow \{(0 \rightarrow 3 \rightarrow 2) \rightarrow (3 \rightarrow 2 \rightarrow 0) \rightarrow (2 \rightarrow 0 \rightarrow 3)\}$ .

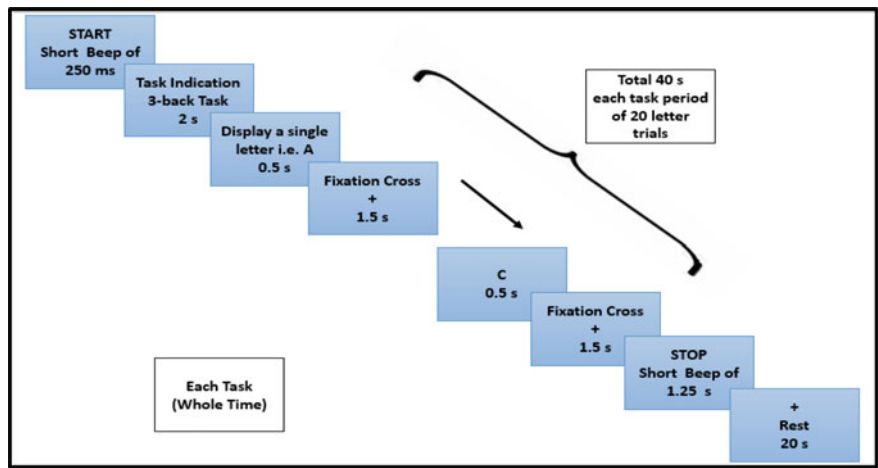


**Fig. 1** Time sequence of n-back task

→ 2 → 0) → (2 → 0 → 3)}. So, for each subject total of 9 n-back series (3 sessions × 3 series) were performed where a single series recording time was 62 s. In a single series, n-back task instructions were played for the first 2 s, followed by 20 trials of the n-back task for the next 40 s, and then the last 20 s were reserved for taking rest. Initially, a 250 ms beep was provided at the start and end of the task. Additionally, the word ‘STOP’ was also displayed at the end of the task followed by a cross fixation during the rest period to keep the subject focused and avoid unnecessary head and eye movements, allowing the brain to relax to the standard baseline state. The time sequence for a single task has been depicted in Fig. 1.

**2.2 Experimental Paradigm**

For the data acquisition, 26 healthy individuals (7 males and 19 females, average age  $26.1 \pm 3.5$  years (mean ± standard deviation) ranging from 17 to 33 years) participated. The participants sat on an armchair in front of a 24" LCD monitor. The distance between the participant’s eyes and the monitor was approximately 1.2 m. They were instructed to put their fingers on some particular buttons and keep their eyes on the monitor and refrain as much as possible from moving their body throughout the data recording. For the n-back task, a series of numbers were displayed to the participants sequentially back to back and were asked to match the numbers displayed on the screen by pressing specific keys on the keyboard, i.e., 7 for 0-back, 8 for 2-back and 9 for 3-back to check whether the number was similar to the number that had appeared n times back.



### 2.3 Data Acquisition

fNIR data has been obtained using NIRScout (NIRx Medizintechnik GmbH, Berlin, Germany) at a sampling rate of 10.4 Hz placing 16 sources and 16 detectors at frontal (16 channels around AFz, AF3, AF4, AF7, and AF8), motor (4 channels each around C3 and C4), parietal (4 channels each around P3 and P4), and occipital (4 channels around POz) areas of the brain. An adjacent source-detector pair configures a NIR channel and that how is 36 channels were configured around AFpz, AFp3, AFp4, AFp7, AFp8, AF1, AF2, AF5h, AF6h, AF7, AF8, AFFz, AFF3h, AFF4h, AFF5h, AFF6, FCC3, FCC4, C3h, C4h, C5h, C6h, CCP3, CCP4, CPP3, CPP4, P3h, P4h, P5h, P6h, PPOz, PPO3, PPO4, PO1, PO2, and POOz according to the international 10–5 system. The fNIRS data represents the relative changes in the concentration of both HbO<sub>2</sub> and dHb. For this experimental study, the oxygenated data have been chosen.

### 2.4 Methods

The fNIRS detects the relative changes in the concentration of HbO<sub>2</sub> and dHb by measuring the light attenuation between the source and detector, placed at 30 mm, following modified Beer-Lambert Law [12], formulated as,

$$I_{\text{out}} = I_{\text{in}} 10^{-\text{OD}_{\lambda}} \quad (1)$$

where,  $I_{\text{out}}$  is the detected light,  $I_{\text{in}}$  is the incident light and  $\text{OD}_{\lambda}$  is the optical density for  $\lambda$  wavelength. The modified Beer-Lambert law allows us to measure the hemodynamics in the form of the optical signal and gives the value of the concentration changes of HbO<sub>2</sub> and dHb. The position of NIR optodes in the human scalp has been demonstrated in Fig. 2. It has been seen that because of the activation of a specific area of the brain due to cognitive workload there will be an increased amount of HbO<sub>2</sub> and decreased amount of dHb generating a negative correlation between the concentration level of HbO<sub>2</sub> and dHb.

For the experimental work, first, the 36-channel fNIRS dataset has been loaded for the n-back task. The data was separated based on those types of tasks like 0-back, 2-back, and 3-back shown in Fig. 3 based on the provided markers and each task consists of 9 trials. Therefore, the data dimension for each type of n-back task has been structured as time-duration  $\times 36 \times 9 \times 26$ , where time-duration represents the cognitive task duration through samples, 36 is the channel number, 9 denotes the number of trials performed by the participants and the number of participants is 26.

The signal has been preprocessed through filtering and baseline correction. The raw fNIRS signal generally contains many artifacts like technical, and biological including cardiovascular effects like fast heart-beat, respiration, and slow waves such as Mayer waves, etc. that may affect the acquired data [7]. The movement of

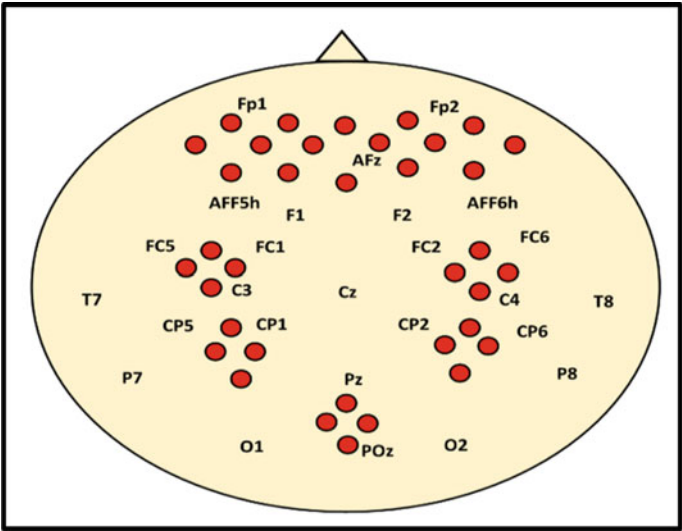


Fig. 2 Position of NIR sources and detectors on the human scalp

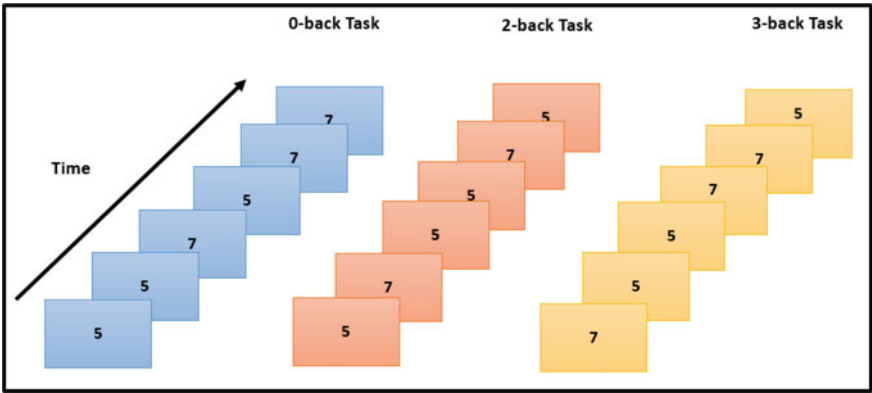


Fig. 3 Data separation based on tasks (0-back, 2-back, and 3-back)

the subject can also affect the recorded signal by generating spikes. Figure 4 shows the raw fNIRS signal of subject 1 for the 0-back task with the spikes for random 3 trials.

To get rid of the artifacts, the whole data have been filtered with a Savitzky–Golay (SG) filter of 3rd order and 121 frame length to remove all the signal components greater than 0.01 Hz. Savitzky–Golay filter is a widely used fNIRS data smoothening filter designed based on local least-squares polynomial approximation [13, 14]. The filtered data has gone through the process called baseline correction to reduce the

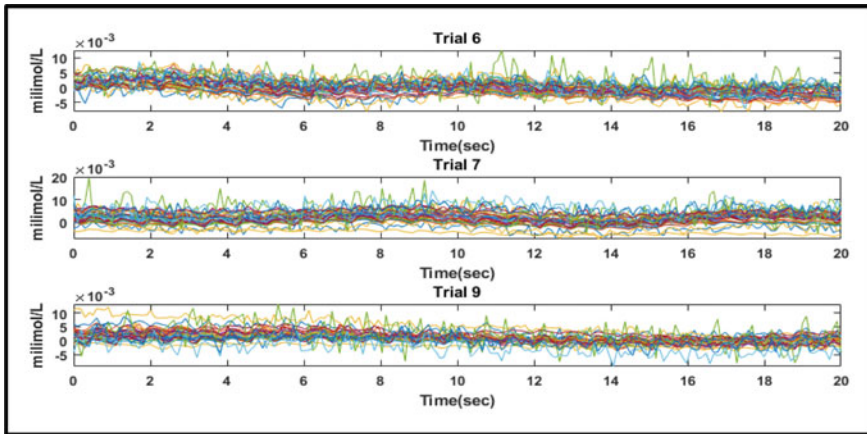


**Fig. 4** The HbO<sub>2</sub> concentration change of raw fNIRS data for 3 trials of subject 1 in the context of the 0-back task

effect of instrumental noises that may cause temporal drifts and may change over time and also across optodes.

Then one-way ANOVA test has been performed with a  $p$ -value  $< 0.001$  based on the mean features of data (concentration change of HbO<sub>2</sub>) for each type of task (0-back, 2-back, 3-back), taking 15 s intervals to detect the significant channels. To classify the workload level, we have considered the 0-back task as a low-level workload and combined the 2-back and 3-back task as a high-level workload. Then 5 sets of statistical features were extracted from each trial for all 36 channels of 26 subjects. The features are as follows: mean value, minimum, maximum, standard deviation, and skewness of HbO<sub>2</sub> concentration.

For the classification, we have applied the Support Vector Machine (SVM) due to its high prediction accuracy. The main mechanism followed by SVM is to generate the hyper-planes that maximize the margin among the clusters of data. The nearest points of the hyper-planes are called support vectors. For this work, MATLAB toolbox in place of the one versus all approach of SVM [7] has been used. In the case of HbO<sub>2</sub>, the classification accuracy was calculated with a fivefold cross-validation technique for all twenty-six participants. The average classification accuracy was also calculated for HbO<sub>2</sub> to justify the effectiveness of fNIRS data.

### 3 Results and Discussion

To ensure the preprocessing assessment, we have depicted the five frontal area channels signal: AF7, AFF5, AFp7, AF5h, and AFp3, and compared them with the filtered and baseline corrected signals with raw data. Figure 5a represents the raw oxygenated data of 1st 5 frontal channels participating 0-back task by subject 1. Figure 5b shows

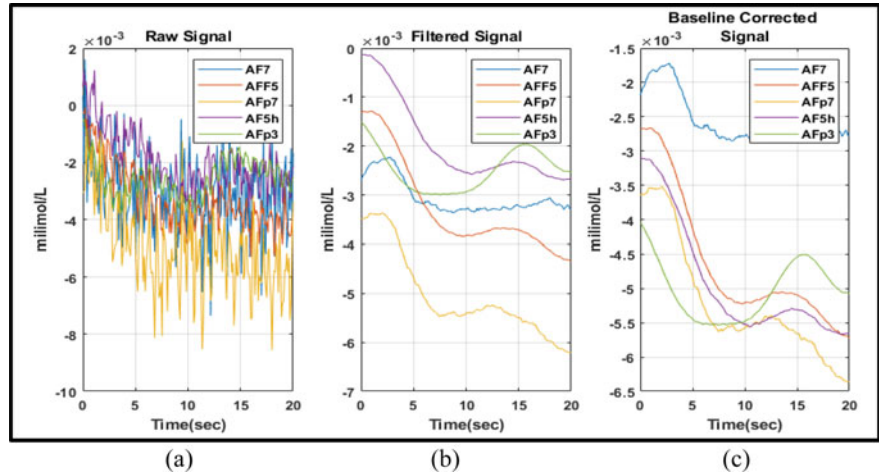


**Fig. 5** The preprocessing stage of fNIRS data for 1st 5 channels of subject 1 in the context of the 0-back task. **a** 0-back raw; **b** 0-back filtered; **c** 0-back baseline corrected

the SG-filtered signals free from artifacts and finally, Fig. 5c represents the baseline corrected signals free from interference-effect.

At the same time, we have also demonstrated the changes in the concentration level of the oxygenated data for 3 different tasks at different time intervals in Fig. 6. From the figure it has been clear that the relative  $\Delta[\text{HbO}_2]$  for the 3-back task is much more than the 0-back and 2-back tasks (3-back > 2-back > 0-back).

To find out the specific regions of the brain that becomes active during the different level of cognitive task one-way ANOVA test was performed. From the statistical

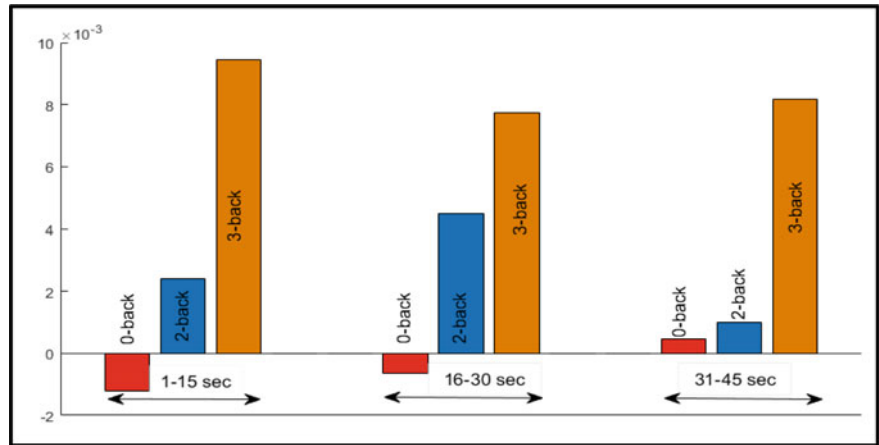


**Fig. 6**  $\Delta[\text{HbO}_2]$  at different time intervals for 0-back, 2-back, and 3-back tasks for a single subject

**Table 1** Subject list with and without significant channels

| Subject IDs with significant channels                    | Subject IDs without significant channels |
|----------------------------------------------------------|------------------------------------------|
| 1, 2, 3, 4, 7, 9, 10, 15, 16, 18, 19, 20, 21, 22, 23, 24 | 5, 6, 8, 11, 12, 13, 14, 17, 25, 26      |

significance test, we have found that some of the subjects show some significant channels and some participants do not show any and that has been listed in Table 1.

The significant test result gives us the idea that the statistical approach is not the appropriate way to figure out the active regions of the brain for all subjects. Consequently, we required an advanced type of learner that will identify and classify the different types of load levels. To achieve this goal, i.e., cognitive load level identification, a soft machine learning approach, SVM is used as the cognitive load level predictor based on supervised learning of subject-based fNIRS data. The general investigation is found from the functionality of the brain that easy task (0-back) and hard task (2-back and 3-back) are significantly separable. On the other hand, finding the difference between two hard tasks like 2-back and 3-back tests is quite complex and subtle. That’s why we have considered 0-back as a low-level workload and 2-back and 3-back as a high-level workload. So, the classification problem is considered as a binary problem. This assumption proceeds to predict two-level cognitive loads (easy and hard) from the fNIRS signal. In this work, SVM is modeled for a binary classifier with the 3rd order polynomial kernel function. The classification accuracy is estimated by a fivefold cross-validation technique. The resulting classification accuracies are given in Table 2 using the feature of average relative change in [HbO<sub>2</sub>].

The final step of our research work was to generate the topo-plots from the fNIRS data to visualize the functionality of the brain at different load levels. Topo-plot is a plot of a topographic map of the scalp data field in a 2D circular view (looking at the top of the head) using the co-interpolation of a fine Cartesian grid. These plots have been generated by the researchers of the Berlin Institute of Technology using their customized software. We have tried to map the images by creating a user defined function in MATLAB that has minimized the cost. The topo-plots generated by the MATLAB showing the functionality of the brain at different load level and different time intervals are depicted in Table 3.

4 Conclusions

This research work has widely examined the fNIRS dataset of cognitive load which was based on 3 different cognitive activities (0-back, 2-back, and 3-back task). Through this work, a positive correlation between cognitive load and oxygen concentration in the prefrontal cortex has been derived and validated in the concept of neurovascular coupling. The use of EEG to classify the cognitive workload affected by poor spatial resolution, provides lower accuracy. As a result, this work proposed

**Table 2** Workload classification accuracy with different train-test ratios

| Subject ID | Classification accuracy (%) |                       |                       |
|------------|-----------------------------|-----------------------|-----------------------|
|            | Train-40%<br>Test-60%       | Train-50%<br>Test-50% | Train-60%<br>Test-40% |
| 1          | 81.82                       | 86.11                 | 82.14                 |
| 2          | 72.73                       | 86.11                 | 82.14                 |
| 3          | 75                          | 83.33                 | 78.57                 |
| 4          | 75                          | 80.56                 | 71.43                 |
| 5          | 61.36                       | 55.55                 | 64.29                 |
| 6          | 59.09                       | 66.67                 | 67.86                 |
| 7          | 81.82                       | 80.56                 | 78.57                 |
| 8          | 61.36                       | 69.44                 | 85.71                 |
| 9          | 63.64                       | 52.78                 | 53.57                 |
| 10         | 56.82                       | 63.89                 | 68.29                 |
| 11         | 86.36                       | 83.33                 | 71.43                 |
| 12         | 75                          | 86.11                 | 85.71                 |
| 13         | 61.36                       | 91.67                 | 82.14                 |
| 14         | 54.55                       | 63.89                 | 67.86                 |
| 15         | 45.45                       | 61.11                 | 71.42                 |
| 16         | 47.73                       | 77.78                 | 67.86                 |
| 17         | 63.64                       | 58.33                 | 57.14                 |
| 18         | 93.18                       | 86.11                 | 82.14                 |
| 19         | 63.64                       | 69.11                 | 75                    |
| 20         | 90.91                       | 88.89                 | 85.71                 |
| 21         | 77.27                       | 69.44                 | 82.14                 |
| 22         | 86.36                       | 86.11                 | 60.71                 |
| 23         | 88.63                       | 86.11                 | 85.71                 |
| 24         | 79.54                       | 80.56                 | 82.14                 |
| 25         | 70.45                       | 63.89                 | 60.71                 |
| 26         | 90.91                       | 94.44                 | 92.86                 |
| Average    | 71.48                       | 75.85                 | 74.86                 |

fNIRS signal to classify the mental workload levels. We also tried to establish a statistical relationship between cognitive activity and the regions of the prefrontal cortex but for some subjects, the relationship cannot be derived. That's why we switched to the supervised machine learning model to classify the cognitive workload obtaining 94.44% highest accuracy and an average accuracy of 75.85%. The final achievement we got was by generating the topo-plots from the fNIRS signal.

However, some faulty data were found in almost every trial and among 36 channels randomly some channels provided faulty data that varies from subject to subject.

**Table 3** Topo-plots of 2 classes of workload at different time intervals

| Load level                          | Time intervals |           |           |           |
|-------------------------------------|----------------|-----------|-----------|-----------|
|                                     | (1–10) s       | (11–20) s | (21–30) s | (31–40) s |
| Low-level (0-back task)             |                |           |           |           |
| High-level (2-back and 3-back task) |                |           |           |           |

During the preprocessing stage, these faulty channels were not avoided due to the variation in data dimensions for all subjects. The reason behind the lower accuracy of some subjects can be explained by the term ‘BCI Illiteracy’ which can be overcome by adding EEG. From the current research, it has been found that generating an EEG-fNIRS hybrid model can enhance the level of accuracy [13].

For the proper assessment of brain functionality, we cannot rely only on this dataset. Shortly, we have the intention to pursue our research on other datasets to compare our findings. We also have the target to continue our survey with different mental tasks to extract more information about brain connectivity and functional patterns that might help in regulating mind-controlling devices. Finally, our next aim is to combine EEG and fNIRS to generate a hybrid model to differentiate the cognitive load level more precisely.

## References

1. Eysenck MW, Brysbaert M (2018) Fundamentals of cognition, 3rd edn. London, Routledge. <https://doi.org/10.4324/9781315617633>
2. Paas F, Renk A, Sweller J (2010) Cognitive load theory and instructional design: recent developments. *Educ Psychol* 38(1):1–4. [https://doi.org/10.1207/S15326985EP3801\\_1](https://doi.org/10.1207/S15326985EP3801_1)
3. Gupta A, Siddhad G, Pandey V, Roy PP, Kim B-G (2021) Subject-specific cognitive workload classification using EEG-based functional connectivity and deep learning. *Sensors* 21(20). <https://doi.org/10.3390/s21206710>
4. Rahman MA, Ahmad M, Uddin MS (2019) Modeling and classification of voluntary and imagery movements for brain–computer interface from fNIR and EEG signals through convolutional neural network. *Health Inf Sci Syst* 7:22. <https://doi.org/10.1007/s13755-019-0081-5>
5. Bagheri M, Power SD (2022) Simultaneous classification of both mental workload and stress level suitable for an online passive brain–computer interface. *Sensors* 22(2). <https://doi.org/10.3390/s22020535>
6. Burle B, Spieser L, Roger C, Casini L, Hasbroucq T, Vidal F (2015) Spatial and temporal resolutions of EEG: is it really black and white? A scalp current density view. *Int J Psychophysiol* 97(3):210–220. <https://doi.org/10.1016/j.ijpsycho.2015.05.004>
7. Khanam F, Aowlad Hossain ABM, Ahmad M (2022) Statistical valuation of cognitive load level hemodynamics from functional near-infrared spectroscopy signals. *Neurosci Inform*. <https://doi.org/10.1016/j.neuri.2022.100042>
8. Jöbsis FF (1977) Noninvasive, infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters. *Science* 198(4323):1264–1267. <https://doi.org/10.1126/science.929199>
9. Naseer N, Hong K-S (2015) fNIRS-based brain-computer interfaces: a review. *Front Hum Neurosci* 9(3). <https://doi.org/10.3389/fnhum.2015.00003>
10. Kirchner WK (1958) Age differences in short-term retention of rapidly changing information. *Exp Psychol* 55(4):352–358. <https://doi.org/10.1037/h0043688>
11. Shin J, Von Lüthmann A, Kim DW, Mehnert J, Hwang HJ, Müller KR (2018) Simultaneous acquisition of EEG and NIRS during cognitive tasks for an open access dataset. *Sci Data* 5:180003. <https://doi.org/10.1038/sdata.2018.3>
12. Bozkurt A, Rosen A, Rosen H, Onaral B (2005) A portable near infrared spectroscopy system for bedside monitoring of newborn brain. *BioMed Eng OnLine* 4:29. <https://doi.org/10.1186/1475-925X-4-29>



13. Rahman MA, Ahmad M (2016) A straight forward signal processing scheme to improve effect size of fNIR signals. IEEE, pp 439–444. <https://doi.org/10.1109/ICIEV.2016.7760042>
14. Schafer RW (2011) What is a Savitzky-Golay filter? IEEE Signal Process Mag 28(4):111–117. <https://doi.org/10.1109/MSP.2011.941097>

# Chapter 30

## Smoke Recognition in Smart Environment Through IoT Air Quality Sensor Data and Multivariate Logistic Regression



S. M. Mohidul Islam and Kamrul Hasan Talukder

**Abstract** Automatic classification and monitoring the human activity using sensors is a decisive technology for Ambient Assisted Living (AAL). Early recognition approaches involved in manually outlining expert guidelines for the sensor values. Recently, machine learning-based human activity recognition methodologies have fascinated a lot of attention. Some human activities produce smoke in the environment which is danger in most situations. The objective of the proposed work is to detect the smoke making activities that are taken out inside any smart environment based on the data assimilated by a set of environmental sensors inspecting the constituents of the air. We have used lower and upper bound based capping technique to handle the outliers and then we have standardized the contents of the features using standard scaling. These preprocessing makes the sensor data more apposite for selected logistic regression algorithm. The outcome of the proposed method shows better result than many state-of-the-art methods.

**Keywords** Smoke activity · Smart environment · IOT air quality sensors · Outlier · Data standardization · Logistic regression

## 1 Introduction

In the last few years, the interest on Ambient Assisted Living (AAL) is increasing because of wide attainability of low-cost nonetheless accurate sensors as well as for high demand of its real-time applications. AAL helps people to stay lively for a longer period, without losing communal associates and freedom [1, 2]. Human activity modeling and recognition is a prime point for AAL and a potential area for many real-world uses such as for smart environments, health care, eldercare,

---

S. M. M. Islam (✉) · K. H. Talukder  
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh  
e-mail: [mohid@cse.ku.ac.bd](mailto:mohid@cse.ku.ac.bd)

K. H. Talukder  
e-mail: [khtalukder@ku.ac.bd](mailto:khtalukder@ku.ac.bd)

psychology and surveillance applications that are mobile and context-aware [3, 4]. The purpose of Human Activity Recognition (HAR) is to identify the kind of activity accomplished by any individual or a group of people, by examining data from sensors or observations, as a role of the context in which they are collected [5]. One important concept in HAR is the monitoring of activities in smart environment.

Activity recognition in smart environment supports various applications such as baby care, eldercare, healthcare, human computer interaction, sociology and security; therefore, this research field has captured the attention since the 1980s [4]. Smart environments can be defined in different applications and settings. One such setting is a smart home, which refers to any living space equipped with a set of sensors, actuators, automated devices to provide ambient intelligence for daily living task support especially to baby, elderly people, patients, and the disabled, remote monitoring, promotion of safety and well-being; compliance with legal regulations, such as the General Data Protection Regulation (GDPR), and privacy preservation [4, 6].

The former assessment of activity is commonly done manually through questionnaires and interviews. The manual assessment of activity of a person is not feasible in a real-life situation because this is a time consuming and error prone process [7, 8]. Various sensor-based (wearable, in-home, intrusive) systems can be found in literature widely to successfully recognize and quantify HAR without relying on self-reporting [9]. In order to detect user's activities, two methods of collecting data from miniature sensors have been extensively researched: an array of environmental or body sensors. In first method, sensors are typically placed in many places in the environment to detect activity by tracking features such as motion, location, and user's interactions with objects [10]. Although this method demonstrated impressive result for complex actions, it has many limitations because of its fixed nature; in the cases such as if the user leaves the place, hides from fixed sensors, or does any activities which are not detectable by these sensors [11]. Alternatively, the second method uses an array of sensors attached to the human body parts like wrists, legs, waist, and chest instead of fixed environmental ones to track the features such as acceleration of particular limbs as well as the body as a whole [10]. Although this method can efficiently detect human activity, it is itchy to carry or wear the body sensors everywhere specially while sleeping or showering [11]. The goal of the research presented in this paper is to improve the recognition result of first approach.

Regardless of the sensor-type, all systems necessitate enormous amount of collected data processing and classification to originate information about the activities. Leading to activity recognition, former methodologies involved in manually defining expert rules and thresholds for sensor stream. This recognition task can nowadays be performed by machine learning models which are able to automatically mine and learn pertinent features [12]. Various machine learning algorithms have been used to detect and classify human activity from sensor data, such as Gaussian Mixture Models [13], Hidden Markov Models (HMM) [10, 14], K-means clustering [15], rule-based [15], probabilistic-based [16], Support Vector Machine (SVM) [17], the Naïve Bayes (NB) classifier [14, 18], and  $k$ -nearest neighbor ( $k$ -NN) [19]. NB and  $k$ -NN often provide decent outcomes despite being relatively simple

and also provide better perception into the “complexity” of the underlying patterns than HMM or SVM, making them the noble state-of-the-art algorithms [9, 19].

This paper focuses on recognition of a specific class of human activity: smoke making activity. Recognition of smoke making activity is an important task which can highly help in the early detection to avoid possible upcoming critical situations and to react rapidly to occurrences or emergency state of affairs. We have used the IOT air quality sensor dataset [19] where wireless sensor data is collected from the home of five different rooms. The Authors in [19] derive information about Activities of Daily Living (ADL) that are carried out inside the room by an individual living solo at home. They did it without any standardization, only by gathering environmental data, acquired by a set of cheap and “easy to use” electrochemical gas sensors. They use MQ2, MQ9, MQ135, MQ137, MQ138, and MG-811 sensors. Sensors were placed randomly where the activities were accomplished inside the rooms. To monitor most of the home ADL, they focus on 4 target situations: normal situation, cleaning activity, meals preparation, and presence of smoke. A microcontroller unit (MCU) gathered six sensors values using general-purpose input/output (GPIO) interfaces, pre-processes them, and sends information to IoT platform with present activity’s label. The goal of the proposed research is to improve the smoke recognition results by applying some useful data processing methods and tuning properly the proposed multivariate logistic regression model.

To describe this, the rest of the paper is organized as follows: after this introductory section, Sect. 2 describes the materials and methods which includes data acquisition and transformation, preprocessing methods, and recognition approach. Section 3 outlines the details of the experimental setup, evaluation of proposed method on recognition results and the comparative results with other state-of-the-art. Finally, concluding remarks on the proposed study and scope of future research is drawn in Sect. 4.

## 2 Materials and Methods

The aim of the present work is to recognize smoke making activity in any smart environment such as in smart home based on chemical composition of the air due to smoke making activities, which can help a care giver for quick response in critical situation. The proposed work involves several main steps: data acquisition and transformation, outlier detection and handling, data standardization, optimized smoke presence recognition model creation and testing. The entire workflow is illustrated in Fig. 1 and elaborated richly in the subsections below.

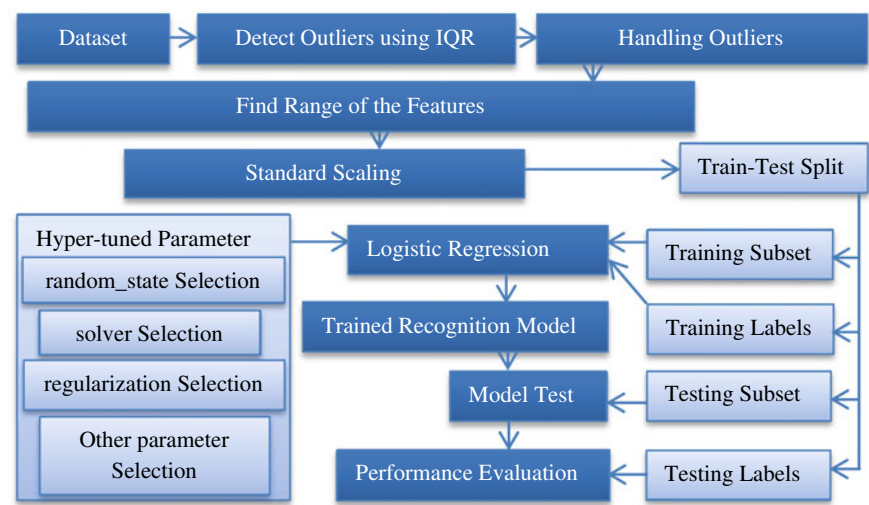


**Fig. 1** The workflow of the proposed smoke presence recognition model

### 2.1 Data Acquisition and Transformation

We have used the Air Quality dataset for ADL classification, prepared by Gambi et al. [19]. We used publicly available version of this dataset and is collected from Kaggle [20]. This dataset contains 1845 samples. Each sample comprises of seven integer values where first six denote gas immersion detected by the sensors and the last one denotes the current action's label. We named the six descriptive features as 'MQ2', 'MQ9', 'MQ135', 'MQ137', 'MQ138', and 'MG811' and one target feature as 'Activity'. As seen before, the original dataset contains 4 target classes. We transformed these 4 target classes into 2 classes: 1 for Smoke presence or Smoke making activity and 0 for all other activities (non-smoke activity). This is because this study focuses on research question: Is the person in smart environment doing smoke making activity or not? That means, our research objective is to recognize smoke presence in smart environment from the data taken form six different environmental sensors. The dataset contains no missing values. But the dataset may contain outlier which is need to handle and need to standardize the dataset also.

### 2.2 Outlier Detection and Handling

Before taking step to forward, we need to identify whether the candidate dataset contains outlier data or not? If yes, they are required to process because outlier data has an extreme impact on the performance of algorithm that we have used as

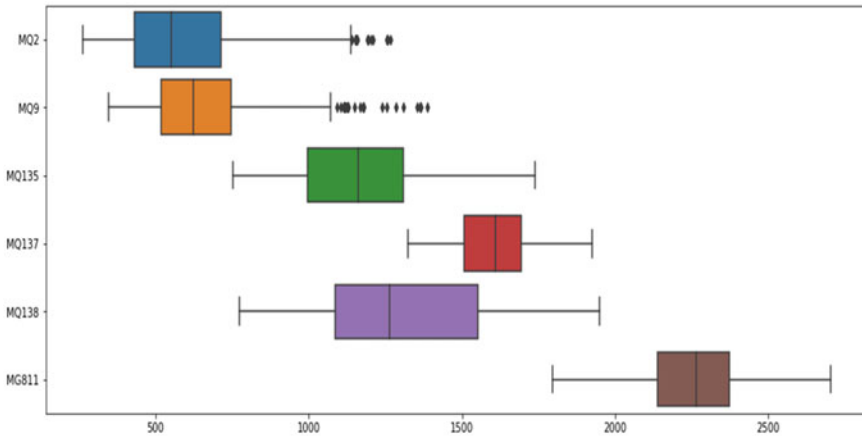


**Fig. 2** Boxplots of descriptive features of air quality dataset

recognition model here. How we detect the outliers and handle them, is described details in the below.

**Outlier Detection.** To know whether any descriptive feature of the dataset contains outlier data, we analyze their boxplots because boxplot visualize data distribution with separation of outliers [21]. The boxplots are illustrated in Fig. 2. If exist, the outliers are plotted individually in two ends of the boxplot.

From boxplots, we see that the features corresponding to MQ2 and MQ9 sensors contain such individual plots in right ends, which represent outlier data. Boxplot only shows whether it contains outlier or not, but it does not specify what are the outlier values. To find the outliers, we have used one mathematical method: Inter Quartile Range (IQR) method. The steps of the method are shown in the following [22]:

1. Sort the data of the feature (that contains outlier) in ascending order
2. calculate the 1st and 3rd quartiles (Q1, Q3) from sorted data
3. compute Inter Quartile Range,  $IQR = Q3 - Q1$
4. compute lower bound =  $(Q1 - 1.5 * IQR)$  and upper bound =  $(Q3 + 1.5 * IQR)$
5. loop through the values of the feature and check for those who fall below the lower bound and above the upper bound and mark them as outliers.

After detecting the outliers, we need to process/handle them as described below.

**Handling Outliers.** There are various methods of handling outliers [22]. But, here we have used a new method named ‘lower and upper bound-based capping’. In this technique, the outliers who exist above the upper bound value, are capped at upper bound  $(Q3 + 1.5 * IQR)$  value and the outliers who exist below the lower bound value, are floored at lower bound  $(Q1 - 1.5 * IQR)$  value. As shown before, in this dataset the outliers only exist above the upper bound value of MQ2 and MQ9 attributes. So, those (outlier) data points are replaced with upper bound value of the

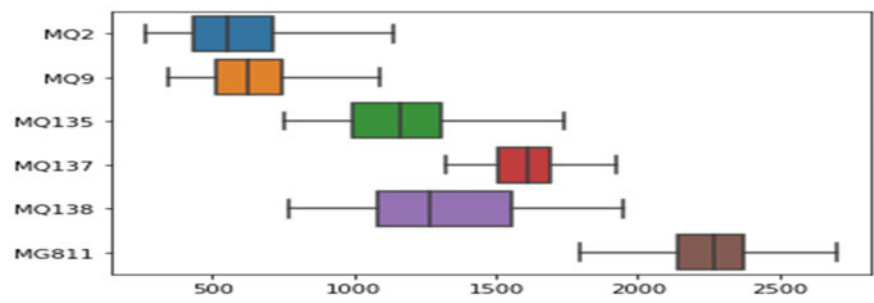


**Fig. 3** Boxplots of descriptive features of air quality dataset after treating the outliers

corresponding features. Figure 3 visualize the data after treating the outliers. From this figure, we see that any of the features contain no outlier now.

**2.3 Data Standardization**

From Figs. 3 and 4a, we can notice that there are large variations in the ranges of the features. Range is the difference between maximum and minimum values, which figure out the dispersion in data [21].

The ranges of descriptive features (after handling outliers) are shown in Table 1.

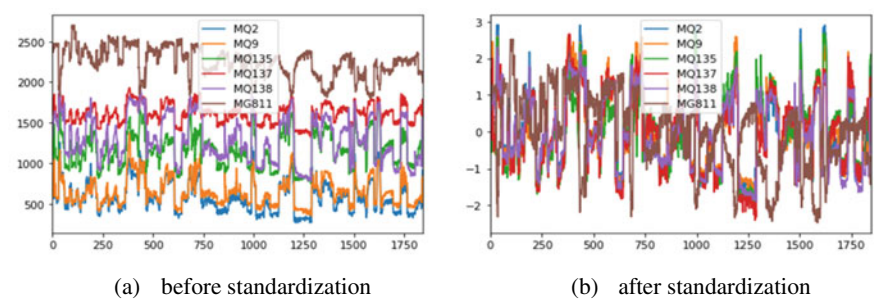


**Fig. 4** Line plot of descriptive features of the dataset

**Table 1** Ranges of the descriptive features

| Feature |       |       |       |        |       |
|---------|-------|-------|-------|--------|-------|
| MQ2     | MQ9   | MQ135 | MQ137 | MQ138  | MG811 |
| 874.5   | 743.5 | 985.0 | 603.0 | 1175.0 | 906.0 |
| Range   |       |       |       |        |       |

Large variation in the data range can bias the outcome of machine learning model. Scaling is a solution of this problem which is a method of equalizing the ranges of features and which relatively unbiased the dataset. Standard scaler is one of the methods of scaling which we have used in this study. Standard scaler standardize the data. It works as Eq. (1) and transforms the data into standard form [21].

$$x_{\text{new}} = \frac{x_i - x_{\text{mean}}}{x_{\text{sd}}}$$

(1)

Which means, standardization is performed by dividing the standard deviation value ( $x_{\text{sd}}$ ) of any feature ( $x$ ) with subtraction of mean value ( $x_{\text{mean}}$ ) from each data ( $x_i$ ) of that feature. After standardization, the mean ( $\mu$ ) of each feature will be 0 and standard deviation ( $\sigma$ ) will be 1. Thus the features value become standardized. Table 2 shows the ranges of the descriptive features of the dataset after standardization. The range values in this table represent that there is almost equal dispersion in data. These can also be justified by line plot of Fig. 4b.

Figure 5 shows the boxplots of the descriptive features of the dataset after treating the outliers and data standardization and this figure justify the equal data dispersion claim of Table 2.

But an issue is observed here. We saw in Fig. 3 that no feature of the dataset contains outliers after handling them. But after data standardization, we see outlier again in MQ9 features. This can be seen from boxplot of Fig. 5. So we need to detect and handle this outlier and we did that in same way as described in Sect. 2.2. Finally, the dataset contains standardized data with no outlier and this well prepared dataset is used for training and testing the logistic regression model.

Table 2 Ranges of the descriptive features after treating the outliers and data standardization

| Feature  |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
| MQ2      | MQ9      | MQ135    | MQ137    | MQ138    | MG811    |
| 4.621539 | 4.396547 | 4.718912 | 5.076260 | 4.205734 | 5.006852 |
| Range    |          |          |          |          |          |

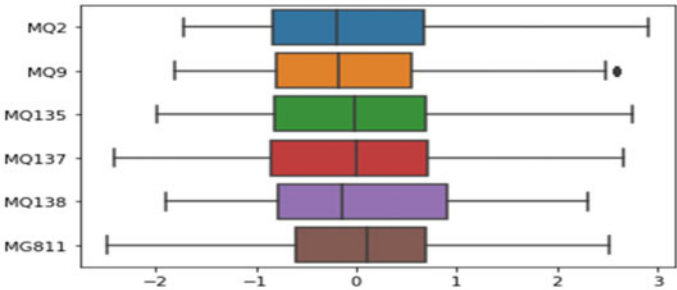


Fig. 5 Boxplots of descriptive features after treating the outliers and data standardization



## 2.4 Learning and Recognition Using Logistic Regression

Learning and classifying the “smoke situation” from sampled data is performed using multivariate logistic regression, a parametric, probabilistic algorithm. Logistic regression provides good performance with relatively less computational cost and small dataset dimension. Also, overfitting problem is very less in logistic regression [23].

Here, the value of a multivariate linear regression ( $y = b_0 + b_1x + \dots + b_nx_n$ ) is modeled into a probability value using sigmoid/logistic function [24]. This probability value is then compared with a threshold value, to predict (binary) class. The multivariate logistic regression is expressed as follows [23]:

$$p = \frac{1}{1 + e^{-(b_0 + \sum_{i=1}^n b_i x_i)}} \quad (2)$$

If this  $p$  value is between 0.5 and 1 then the final output of the logistic regression will be (class) 1, otherwise the output will be (class) 0 [23]. Figure 6 shows the flowchart of logistic regression.

By aligning to this study, this multivariate logistic regression can be expressed as:

$$p = \frac{1}{1 + e^{-(b_0 + b_1(MQ2) + b_2(MQ9) + b_3(MQ135) + b_4(MQ137) + b_5(MQ138) + b_6(MG811))}} \quad (3)$$

In the above equation,  $p$  is the calculated probability value,  $MQ2$ ,  $MQ9$ ,  $MQ135$ ,  $MQ137$ ,  $MQ138$ , and  $MG811$  are the features of the dataset,  $b_0$  is the y-axis intercept,  $b_1, \dots, b_6$  are the coefficients/weights which are unknown and the model will estimate them when trained on the available values of the dataset.

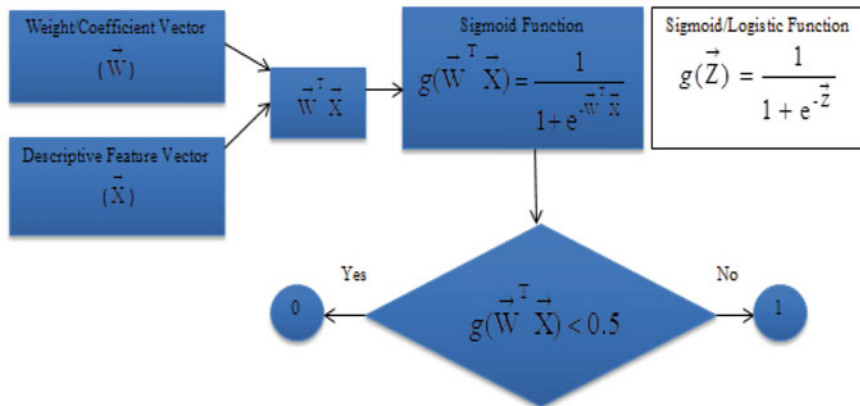


**Fig. 6** Flowchart of logistic regression

3 Results and Discussion

Randomly selected 80% data of the dataset is used for training and the rest 20% is used for testing or recognition. That means, among 1845 samples, 369 samples are used for testing. For evaluating classifier performance, we have used accuracy as performance metric. Accuracy is the number of instances appropriately classified over the total number of instances in the test set [21].

3.1 Experiment Setup

A Python program was developed to recognize each  $x_i$  and compare the recognized class  $y'_i$  with the actual class  $y_i$ . The parameters were selected with the aim of getting the best classification of dataset. In Table 3 we have listed all the selected values assigned to the parameters for our experiments.

From Table 3, it is seen that we have selected random state parameter,  $random\_state = 43$ . Using a fixed integer value for random state will produce the same results across different calls. This value is selected by experiments. In our work random state was selected that maximizes validation accuracy. Many validation tests were executed by changing random state value. The following procedure has therefore been implemented:

- a. The process is reiterated for random state values between 1 and 100, while keeping other parameters fixed as in Table 3
- b. At the end, we selected random state which maximizes the validation accuracy.

Figure 7 shows the accuracy for various random state. From this figure, we see that the method shows best accuracy for  $random\_state = 43$ . That's why, this value of  $random\_state$  is selected.

**Table 3** The values assigned to the parameters for logistic regression

| Parameter         | Value assigned |
|-------------------|----------------|
| class_weight      | None           |
| fit_intercept     | True           |
| intercept_scaling | 1              |
| max_iter          | 100            |
| penalty           | l2             |
| solver            | lbfgs          |
| random_state      | 43             |

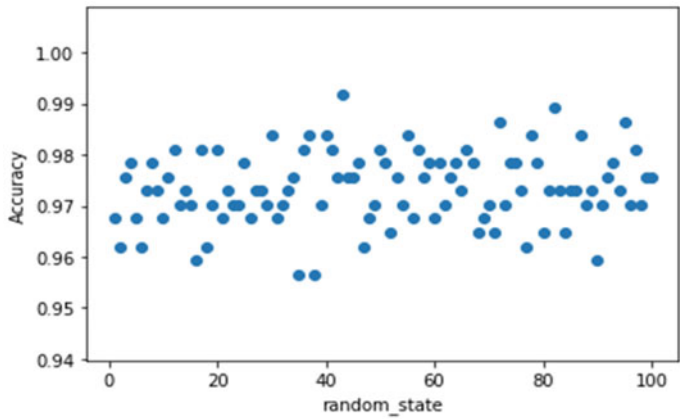


**Fig. 7** Random state versus accuracy

**3.2 Recognition Outcomes**

After hyper-tuning the logistic regression algorithm for the prepared dataset, we move on to the recognition operation. Table 4 presents the tuned logistic regression model’s outcomes regarding classification accuracy value for various preprocessing methods of proposed study. Results of all cases provided in the table are achieved from same train and test sets.

Though our method is to use data standardization and upper bound-based capping, but for clearness, we experimented in six different approaches and have presented the results to illustrate more accuracy while data standardization and handling outliers by upper bound is provided. From above table, we see comparatively less accuracy when neither data is standardized nor outliers are handled. Again, we see that removing outliers is not a good solution in handling outliers’ case. In this case, replacing (capping) the outliers by upper bound value, is a good solution. That means, using both data standardization and handling outliers by upper bound provides best accuracy, which is 99.19%.

**Table 4** Recognition results

| Experiment No. | Data standardization | Handling outliers               | Accuracy (%) |
|----------------|----------------------|---------------------------------|--------------|
| 1              | No                   | No                              | 98.92        |
| 2              | No                   | Removing outliers               | 98.62        |
| 3              | No                   | Capping outliers by upper bound | 98.65        |
| 4              | Yes                  | No                              | 98.19        |
| 5              | Yes                  | Removing outliers               | 98.90        |
| 6              | Yes                  | Capping outliers by upper bound | 99.19        |

**Table 5** Results comparison

| Work              | Preprocessing used                             | Classifier used     | Accuracy (%) |
|-------------------|------------------------------------------------|---------------------|--------------|
| Proposed work     | Upper bound-based capping and standard scaling | Logistic regression | 99.19        |
| Gambi et al. [19] | No preprocessing                               | <i>k</i> -NN        | 98.85        |

**3.3 Performance Comparison**

We have compared our best result with Gambi et al. [19], who prepared the dataset used in this study and also experimented on it. The comparative result is shown in Table 5.

The reported results of Table 5 confirm that properly tuned logistic regression model using preprocessing methods applied in this study shows more accurate result for same dataset and for same research question. Besides this, our other two experiments also show better accuracy (experiment number 1 and 5 in Table 4) than the method in [19]. So the scenario examined in this paper conclude that logistic regression model with properly processed data is likely to be better suited than other common recognition algorithms, such as *k*-NN for distinguishing and recognizing smoke in smart environment through air quality sensor data.

**4 Conclusion**

This article reveals the viability of smoke making activities recognition using data assimilated by a set of cost-effective gas sensors. The sensor data contains noises or outliers which are removed by proper outlier handling method. Standard scaler is used to standardize the range of various sensors data. A logistic regression algorithm is used to anticipate recent smoke situation based on known data. The accuracy of the proposed system is more than 99%, hence detecting with high accuracy the considered smoke presence, a possible dangerous situation. After properly training the classifier and tuning accordingly, the proposed system can be applied to any smart environment. Further study can be carried out to predict multi-action activities that are performed in a room at a time or in different time.

**Acknowledgements** This work was supported by Information and Communication Technology Division (ICTD), Ministry of Posts, Telecommunications and Information Technology, Government of People’s Republic of Bangladesh, through ICT fellowship.

## References

1. Monekosso D, Florez-Revuelta F, Remagnino P (2015) Ambient assisted living [guest editors' introduction]. *IEEE Intell Syst* 30(4):2–6
2. Gambi E, Ricciuti M, Ciattaglia G, Rossi L, Olivetti P, Stara V, Galassi R (2018) A technological approach to support the care process of older in residential facilities. In: Leone A, Caroppo A, Rescio G, Diraco G, Siciliano P (eds) *Italian forum of ambient assisted living*. Springer, Cham, pp 71–79
3. Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity recognition from accelerometer data. *AAAI* 5(2005):1541–1546
4. Sahaf Y (2011) Comparing sensor modalities for activity recognition. Doctoral dissertation, Washington State University
5. Ranasinghe S, Al Machot F, Mayr HC (2016) A review on applications of activity recognition systems with regard to performance and evaluation. *Int J Distrib Sens Netw* 12(8):1550147716665520
6. Sehili MA, Lecouteux B, Vacher M, Portet F, Istrate D, Dorizzi B, Boudy J (2012) Sound environment analysis in smart home. In: Patern F, de Ruyter B, Markopoulos P, Santoro C, van Loenen E, Luyten K (eds) *International joint conference on ambient intelligence*, vol 7683. Springer, Berlin, Heidelberg, pp 208–223
7. Wilson D, Consolvo S, Fishkin K, Philipose M (2005) In-home assessment of the activities of daily living of the elderly. In: *Extended abstracts of CHI: workshops—HCI challenges in health assessment*, p 2130
8. Debes C, Merentitis A, Sukhanov S, Niessen M, Frangiadakis N, Bauer A (2016) Monitoring activities of daily living in smart homes: understanding human behavior. *IEEE Signal Process Mag* 33(2):81–94
9. Urwyler P, Rampa L, Stucki R, Büchler M, Müri R, Mosimann UP, Nef T (2015) Recognition of activities of daily living in healthy subjects using two ad-hoc classifiers. *Biomed Eng Online* 14(1):1–15
10. Stikic M, Huynh T, Van Laerhoven K, Schiele B (2008) ADL recognition based on the combination of RFID and accelerometer sensing. In: *2nd international conference on pervasive computing technologies for healthcare*. IEEE, pp 258–263
11. Gomaa W, Elbasiony R, Ashry S (2017) ADL classification based on autocorrelation function of inertial signals. In: *16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, pp 833–837
12. Compagnon P, Lefebvre G, Duffner S, Garcia C (2020) Learning personalized ADL recognition models from few raw data. *Artif Intell Med* 107:101916
13. Almudevar A, Leibovici A, Tentler A (2008) Home monitoring using wearable radio frequency transmitters. *Artif Intell Med* 42(2):109–120
14. Van Kasteren TLM, Englebiene G, Kröse BJ (2010) An activity monitoring system for elderly care using generative and discriminative models. *Pers Ubiquit Comput* 14(6):489–498
15. Berenguer M, Giordani M, Giraud-By F, Noury N (2008) Automatic detection of activities of daily living from detecting and classifying electrical events on the residential power line. In: *10th IEEE international conference on e-health networking, applications and services (HealthCom 2008)*. IEEE, Singapore, pp 29–32
16. Bang S, Kim M, Song SK, Park SJ (2008) Toward real time detection of the basic living activity in home using a wearable sensor and smart home sensors. In: *30th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, pp 5200–5203
17. Fleury A, Vacher M, Noury N (2009) SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. *IEEE Trans Inf Technol Biomed* 14(2):274–283
18. Tapia EM, Intille SS, Haskell W, Larson K, Wright J, King A, Friedman R (2007) Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In: *11th IEEE international symposium on wearable computers*. IEEE, pp 37–40

19. Gambi E, Temperini G, Galassi R, Senigagliesi L, De Santis A (2020) ADL recognition through machine learning algorithms on IoT air quality sensor dataset. *IEEE Sens J* 20(22):13562–13570
20. Air Quality Dataset. <https://www.kaggle.com/datasets/saurabhshahane/adl-classification>. Accessed 21 Jan 2022
21. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier
22. Detecting and Treating Outliers. <https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/>. Accessed 28 Jan 2022
23. Nehal N (2018) Machine learning algorithm, 1st edn. Dimik Publication, Dhaka
24. Logistic regression. [https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression). Accessed 29 Jan 2022

# Chapter 31

## Learning from Learned Network: An Introspective Model for Arthroscopic Scene Segmentation



Shahnewaz Ali, Feras Dayoub, and Ajay K. Pandey

**Abstract** The knee joint, due to its complex, confined, and dynamic nature, poses several challenges in visualizing anatomical details for diagnostic and minimally invasive surgical procedures. Furthermore, presence of texture- and feature-less tissues introduce image artifacts, and large intra-class dissimilarity makes automatic image segmentation a complex problem to solve. Segmentation models cannot handle situations when image frame distribution differs from the training data set (out-of-distribution). In this study, we implement a secondary ‘learning network’ with the ability to learn as the primary segmentation network fails in handling segment with inconsistent data distributions. The combined architecture significantly increases the overall accuracy and fidelity of the segmentation process. We have achieved this by evaluating internal feature maps and confidence maps of the segmentation model in a separate training within the fully convolutional network. The proposed two-stage segmentation network flags failures in segmentation and re-labels them as it can learn both in- and out- data distributions. This provides state-of-the-art segmentation accuracy across different tissue types with dice similarity 0.9112, 0.963, 0.862, for ACL, femur, tibia, respectively.

**Keywords** Knee arthroscopy · MIS · Robotic-Assisted surgery · Segmentation · Surgical scene segmentation · Deep learning · Failure prediction · Failure correction · Uncertainty · Performance monitoring

---

S. Ali (✉) · F. Dayoub · A. K. Pandey

School of Electrical Engineering and Robotics, Faculty of Engineering, Queensland University of Technology, Gardens Point, Brisbane QLD 4001, Australia

e-mail: [shahnewaz.ali@hdr.qut.edu.au](mailto:shahnewaz.ali@hdr.qut.edu.au)

# 1 Introduction

Knee arthroscopy is a complex surgical procedure commonly used to diagnose and treat joint injury [1]. Poor visualization of the surgical site further makes it a challenging task for surgeons, who struggle to maneuver tools inside the confined knee cavity, therefore, some unintentional tissue damage occurs [1–6]. Tissue awareness of the knee anatomy and depth sensing techniques in an automatic and intra-operative manner has significant prospects in improving surgical outcomes [7–10]. Such awareness can be achieved through surgical scene segmentation (Fig. 1).

Multi-class tissue and tool segmentation in arthroscopy is still an open challenge [1, 2, 8, 11]. From recent studies, it is clear that the accuracy of the segmentation process often drops due to the constraints imposed by image visualization and lack of visual information present in surgical frames [1, 6]. Factors such as age, injury, and lifestyle can further change the appearance of tissues structures. These lead to large variations which subsequently increase variance in intra-class feature distributions. Moreover, accurate ground-truth labels and high-quality surgical data are often inaccessible. Failures caused by change in frame distributions without further indications in segmenting scenes are common occurrences for segmentation models which referred to as domain shift problem in the deep learning terminology. Therefore, in medical image analysis frequently a predictive model requires uncertainty maps to gain trust, prior to the technological adaption in a clinical routine [12].

In this study, we proposed a framework for surgical scene segmentation in the context of knee arthroscopy, where the baseline segmentation model (micro model) learns the segmentation processes. Subsequently, the add-ons macro architecture for failure predictions and corrections (FPC), learns segmenting process from out-of-distribution (OOD) data. The macro model, FPC, takes the benefit of exploring the learned high-level feature maps of the base model and use the confidence maps to recognize the OOD features where the micro model may experience failures. The error maps were injected into the correction module, and subsequently, during the correction phase, the OOD features were corrected through evaluation of injected features and confidence maps. A fusion block is applied to combine initial segmented

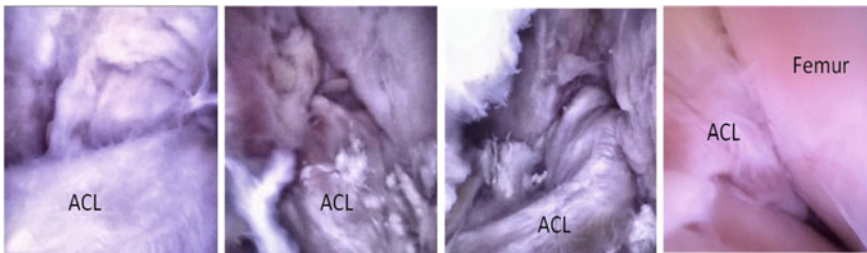


**Fig. 1** Arthroscopic video sequences of different cadaver knee experiments. Variation in their appearances are observed mostly due to illuminations, color cast, shadow and dark area, camera pose and tissue physiological changes



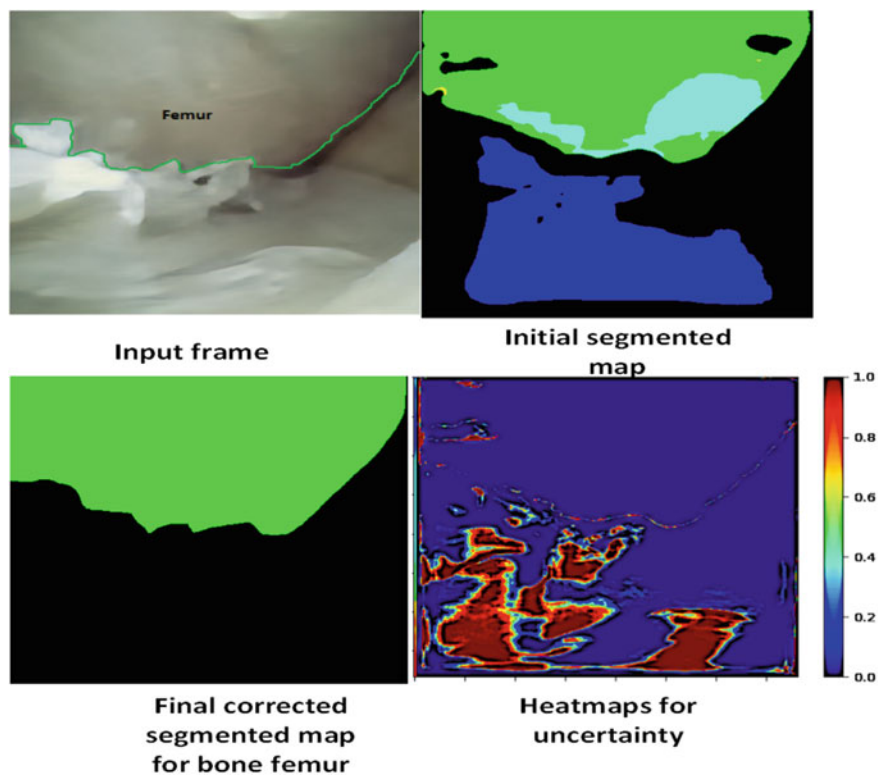


**Fig. 2** The pipeline of the proposed segmentation model. Segmentation model takes an input and provides initial segmented maps. FPC model evaluates input image as well as feature maps, and confidence maps from segmentation model to learn true confidence in terms of failure probability in predicted initial segmented map- as it is shown in heatmap. During the correction and fusion stage, model learns the failures, therefore, infer misclassified labels. Learnable fusion phase learns map to map translation by combining the initial confidence, error and corrections

maps and corrected pixels labels where failures were detected to achieve final segmented maps as is shown in Fig. 2. Our model able to flag silent failures with true failure probability and their corrections. The proposed model opens a room in a clinical routine to adapt a deep learning-based predictive model.

## 2 Related Work

Feature- and texture-less regions of arthroscopic video frames intensify both short- and long-term dependency problems in a fully convolutional neural network (FCN) [6, 8]. In another study, we have proposed an advanced segmentation network that takes the benefits of multiscale feature map propagation mechanisms between the

stacked convolutional layers and use shape information to overcome the above problems [6]. The model achieved highest accuracy to segment arthroscopic data compared to the state-of-art models, hence, it has been used as baseline segmentation model in this study.

Failure in prediction is common to any segmentation network. It has been seen that failure in segmenting scene structure is most likely caused by the change in distributions between the training and test dataset, which is most likely the situation observed in all knee arthroscopic experiments [2, 4, 6–8]. Failure prediction to tackle OOD data has received a lot of research attentions [12–20]. In the medical domain, deep learning techniques are usually trained with limited datasets termed as in-of-distribution (IOD). Thus, the ability to know uncertainty in predictions is an important feature to increase fidelity. Recently, many approaches have been proposed to estimate uncertainty by analyzing the entropy of the predictions such as the Bayesian neural network, Monte-Carlo (MC) dropout, and deep ensemble models [13–15]. It has been found that these approaches are inefficient to capture aleatoric and epistemic uncertainty types. Common aleatoric uncertainty sources in arthroscopy include semantic borders and distant and/or underexposed parts of the scene. Moreover, large variance in the intra-class similarity of the same tissue structures causes ambiguous labels, meaning that segmentation models can fail even with high confidence. Hence, the maximum probability distribution (MPD) on softmax output inefficiently identifies failures [19, 20]. The recent progress in this direction is based on a learning strategy where the model learns pixel-wise confidence for segmentation [16–20]. For instance, methods proposed by Kuhn, Christopher B et al. used a separate introspective model to predict errors experienced by the segmentation models [19]. In their work, they used the same architecture as the segmentation model for introspective error prediction. During training, the same feature of the encoder is used, therefore, only the decoder is trained to predict errors from the feature map.

In this study, we extend the previous idea of error prediction for segmentation tasks and proposed failure detection and correction model in the context of knee arthroscopy. Arthroscopic scenes highly experience aleatoric uncertainty which is observed in our dataset as well. High variations between the intra-class similarity make the segmentation process a complex problem to solve. The proposed FPC module is an extension of the introspective concept similar to that of Kuhn [19]. By evaluating feature maps of segmentation networks and knowing where the base model experiences failures, the proposed FPC model also provides a correction mechanism to improve segmentation accuracy.

### 3 Cadaver Experiment and Data Collection

This study uses images of knee joint from several cadaveric experiments performed as a part of our ongoing research activity in knee arthroscopy. In summary, for the cadaver experiments we used a newly developed prototype stereo arthroscope (imaging system) to record arthroscopic video sequences. Details of the imaging

system and cadaver experiments are available in the references [3, 6]. Cadaver experiments were conducted at QUT's Medical and Engineering Research Facility (MERF).

A large variation in frame distributions among the cadaver arthroscopic video sequences was observed. The dataset quality was additionally affected by illumination conditions. Uneven illumination inside the knee cavity can change the distribution of the frame information; in many occurrences, overexposed (saturated) frames were observed [2–4, 11]. Image artifacts due to floating tissue, debris, noise, and blur are common to most frames. The slight discrepancy of view angle also occurred in the cadaveric experiments, which became a major source of variation in tissue appearance in the dataset.

## 4 Methodology

In the subsection ‘Model’, we explain our proposed cascade model for segmenting arthroscopic surgical scenes as well to gain dependability in predictions. The first stage baseline model segments surgical scenes where the second stage cascade network learns the failures of the baseline network and improves the segmentation accuracy through label correction in OOD.

### 4.1 Model

The framework proposed here is independent of the baseline segmentation model. Therefore, any network which serves a segmentation purpose can be used with our proposed framework as baseline model. The baseline network incorporates the residual learning strategy as a building block to mitigate information loss during the down-sample phase in each layer. Mathematically, residual learning strategy is defined as follows [6];

$$X_{l+1} = \sigma(X_l + F(X_l, W_l)) \quad (1)$$

where  $F(\cdot)$  known as a residual function,  $\sigma$  denotes the activation functions,  $X_l$  and  $X_{l+1}$  are the output feature map of previous and current layer, respectively, and  $W_l$  denotes the learnable parameters of convolutional blocks (Fig. 3).

The Baseline segmentation network establishes dense connections among the layers to strengthen feature propagation and enable better gradient flow. Dense connections are achieved by fusing all previous layer outputs from the encoder and decoder layers, and between the encoder and decoder. It is proved to be an effective way to tackle long- and short- range dependency problems. The learnable spatial and channel-wise attention mechanism is used to model crucial and important features where provides minimal score to others. In the spatial domain,

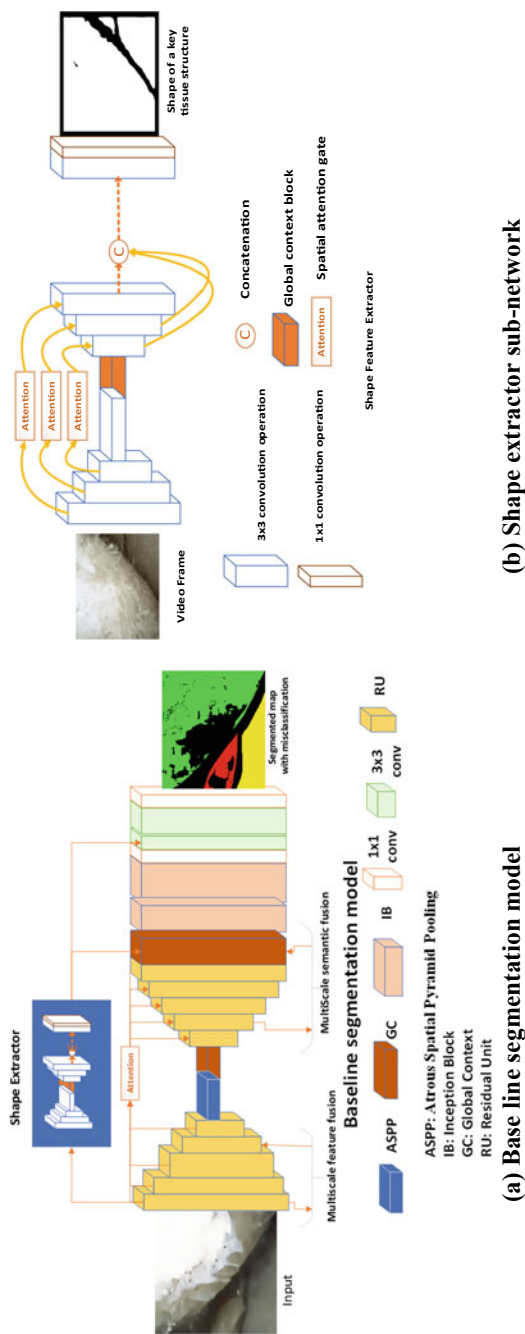


**Fig. 3** Baseline segmentation model obtained from our previous work [6]. **a** Represent the segmentation model which uses shape and spatial feature to learn segmenting process in the context of knee surgery. **b** Represents the shape extractor module which extracts shape of key tissue structures. Model use dense connections between the layers to strength to feature propagation

surgical scenes are limited by discriminative features. In direct field-of-view, the tissue signature shows channel-wise dependency, therefore, it is used to calibrate feature maps [6]. Multiscale features from the feature maps are extracted by Atrous Spatial Pyramid Pooling (ASPP) and Inception Block [21, 22]. The bridge layer ASPP module is used in conjunction with the global context block to extract multiscale features from high-level feature maps as well as to model global contextual information [23]. The Inception block is used at the end of the decoder. Moreover, the network contains a sub-network block that extracts shapes of the key tissue structure, which are then defined as shape features [6]. In surgical scenes, shapes of key tissue structures are informative, for instance, the shape of bone structure is not similar to the shape of tissue type meniscus.

The failure prediction and correction (FPC) model is added to the baseline network (Fig. 4). Similar to the previous introspection approach, we used the same architecture as the baseline segmentation network. The FPC model contains two subnetworks, a failure detection module (FD), and a label correction module (LC). The failure detection module is based on the fully conventional network strategy. The encoder block takes an input image and a previously predicted confidence map from the segmentation network. High level feature maps of the segmentation model are shared between the two blocks to enable feature map evaluation and to ease the training process. The ground-truth errors are calculated from the errors previously experienced by the segmentation network. The FD module provides a binary image where label ‘1’ denotes that the segmentation network is most likely to fail and ‘0’ where the segmentation network achieved success.

During the training of FD model, a set of input image  $I_{\text{Image}} = \{I_1, I_2, \dots, I_N\}$  and its corresponding obtained confidence maps  $C_{\text{confidence}} = \{C_1, C_2, \dots, C_N\}$  were given as input of the network. Therefore, the input of the FD module is

$$S = I_{\text{Image}} + C_{\text{confidence}} \quad (2)$$

where  $+$  denotes channel-wise concatenation. The aim of the FCN model is to determine error probability  $y = f(I|I_{\text{pred}}, F_c)$  and  $y \in [0, 1]$ , from a function of features  $F_c = \{F_{\text{ext}} + F_{\text{Prev,ext}}\}$  and previously computed label weights  $l_{\text{pred}}$ . The network achieved the ability to explore entire feature maps that has been used previously and the recent one associated with label weights and provide a true error score to justify label accuracy. It enables the fault detection ability directly from features; hence, it learns the ability of the initial learned segmentation model.

The LC module follows the baseline segmentation architecture with dense connections, global context, and multiscale feature extraction. The LC module also shares the high-level feature maps of the segmentation model. Similar to the FD model, the LC model takes an input image and a previously predicted confidence map from the segmentation network as described in Eq. (2). The LC module extracts features from the input, and in conjunction with the shared feature maps obtained from the segmentation network, it learns the correct pixel-wise label  $l_c = \{l_1, \dots, l_k\}$

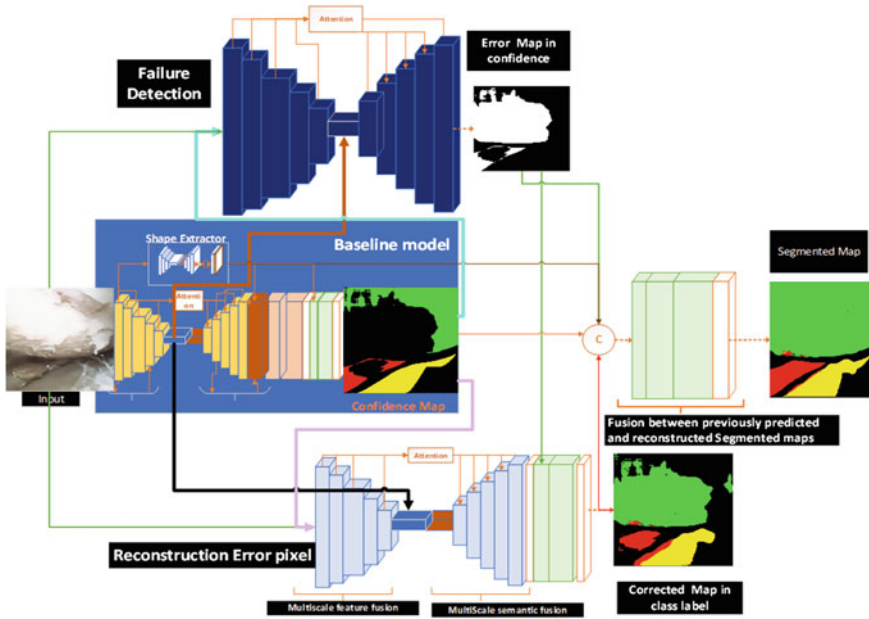


Fig. 4 Complete segmentation model including model add-on FPC module (macro section)

from  $F_c = \{F_{\text{ext}} + F_{\text{Prev.ext}}\}$  where  $k$  is the maximum number of class in segmentation maps. To do so, the high-level semantic features and error map were concatenated at higher label, work as a weight mask to learn pixel semantic labels in successive convolution operations. It enables failure-aware, pixel-wise label corrections to occur  $l_k = \text{Cov}(X_i, \theta)$ , where  $\text{Cov}$  is the convolutional operations and  $X_i = F_{\text{semantic}} + W_{\text{Failure Score}}$  where  $+$  denotes channel-wise concatenation and  $X_i$  is the conjugated features map masked with error score  $W_{\text{Failure Score}} = \{P_i, i \in [0, 1]\}$  for each pixel.

Additionally, the learnable fusion block enables the fusion mechanism among the confidence maps obtained from modules. It takes a confidence map from initial segmentation, error map, and corrected labels for the error map as inputs, therefore, the final segmented maps are then computed through successive convolutional operations. It provides a way to segment the surgical scene in an end-to-end fashion. Due to corrected pixel maps providing only the pixel labels which are retrieved, and others are as background, therefore, other ways to compute the final segmentation map can be done through the direct fusion between corrected pixel maps and initial segmented maps.

In short, the proposed method learns medical image segmentation process in two directions—(i) Baseline segmentation network learns the frame segmenting process from in-of-distribution data (training dataset), and (ii) The FPC learns the frame segmenting process from out-of-distribution data (training and validation dataset).

## 4.2 Training

The segmentation and FPC networks were trained using Adam optimizer. The feature ‘LearningRateScheduler’ of the Keras library was used to achieve an adaptive learning rate. Initial learning set to 0.001. The micro module that is the base-line segmentation network was trained with categorical cross-entropy (CE) and dice similarity coefficient loss function which is defined as follows;

$$\text{DCE}(T, P) = 0.5 * \text{CCE}(T, P) + (1 - \text{Dice}(T, P)) \quad (3)$$

Categorical cross-entropy (CE) is defined as follows;

$$\text{CCE} = - \sum_i^C t_i \log t_p \quad (4)$$

Here,  $C$  denote the total category that is the class number and denote the ground-truth label and predicted label. The Dice similarity coefficient (DCE) defined as follows;

$$\text{Dice}(T, P) = \frac{2 * \sum T_i P_i + \text{Smooth}}{(\sum T_i + \sum P_i \text{Smooth})} \quad (5)$$

In order to tackle data imbalanced problems, this study found that it is more effective to increase low-frequency tissue structure repeatedly with their augmented versions. We used scale-shift and center crop augmentation methods. The rotation augmentation method was applied to rotate frames from 0 to 180° with 10-degree increments. The dataset contains one additional cadaveric experimentation which was conducted initially to record only the key tissue structure femur. That cadaveric experiment’s data was not included for the cross-validation but used for the training of the micro-segmentation network. Frames were preprocessed using the method proposed in Ref. [4].

The data from all three cadaveric experiments were used for the four-fold-cross-validation. Therefore, during each validation experiment, three cadaveric experimental data along with 1st cadaveric experimental data were used to train the network. The trained network was then validated on the rest of the cadaveric experimental data. During the training phase, the segmentation model was trained till the validation accuracy falls. During the cross-validation experiments as reported in the Ref. [6], it has been found that for two datasets (experiments 1 and 4), the model achieved higher segmentation accuracy. This is due to, the imaging conditions being relatively high, less color cast was observed in these two cadaveric experiments. Moreover, minimal tissue degenerations were observed. Therefore, in this study, these two experiments were excluded.

For the complete test, all four cadaveric experimental datasets were combined. Among them, 60% frames were used in training the micro-segmentation model and approx. 15% were used to obtain confidence maps. 30% frames were used to test

complete model. The 15% frames were included for FPC model training along with the training frames. We recommended to use complete dataset especially when during the training of LC module. After the training of the micro-segmentation network, we set its trainable parameter to false. After this step, we inferred segmentation maps and calculated errors. These confidence maps and error maps were then used to train macro model.

Misclassification for each pixel is identified by two maps, (i) binary segmented maps which express where the segmentation model fails, and (ii) error maps which express the true class labels of the error pixels in a frame. The binary segmented maps were then used to train FD modules while setting the segmentation model to freeze (untrainable). We used binary cross-entropy as a loss function and Adam optimizer with a learning rate of 0.001. The LC module was trained following a similar procedure but used the same loss function of the segmentation network. Both modules LC and FD take input frames and confidence maps from the segmentation network. The output of the model is the corrected segmented maps. As stated before, it has significant importance for clinical settings to know where a network experiences failure, this model also provides a probabilistic map for further clinical assessments.

## 5 Results and Discussion

The cross-validation results of the segmentation network based on Ref. [6] and the proposed model are shown in Fig. 5. Our proposed method achieved higher segmentation accuracy.

The proposed add-ons module learns the segmentation policy from the inconsistent OOD; therefore, it flags the failures and correct them. Doing so, the model accuracy improvements for the challenging cadaver were 0.188, 0.324, 0.09, 0.086 for ACL, meniscus, femur, and tibia respectively. The Average improvement were 0.2415, 0.183, 0.0385, 0.036, and 0.14 for ACL, meniscus, femur, tibia, and tools, respectively. With training test data split, our proposed model achieved dice similarity 0.9112, 0.963, 0.862, and 0.992 for the tissue type ACL, femur, tibia, and tools. In similar study, Y. Jonmohamadi et al. achieved the average dice similarity coefficients for femur, tibia, ACL, and meniscus were 0.78, 0.50, 0.41, and 0.43, respectively, with the advanced architecture U-Net++ [11]. In this context, our proposed model achieved highest accuracy among the state-of-the-art methods for knee arthroscopy to segment full surgical scenes [1, 6, 8, 11].

Here worthwhile to mention that ACL is the most challenging and important tissue structure of the knee anatomy- the ligament which controls the leg motions and rotations. Tissue type femur and ACL are the potential candidates for landmark tissue structures to track soft tissue. The learning model also found that segmentation model often experiences failure near to the border of each tissue structures (aleatoric uncertainty) as it is shown in Fig. 6—Upper image, which were detected and corrected by our model.



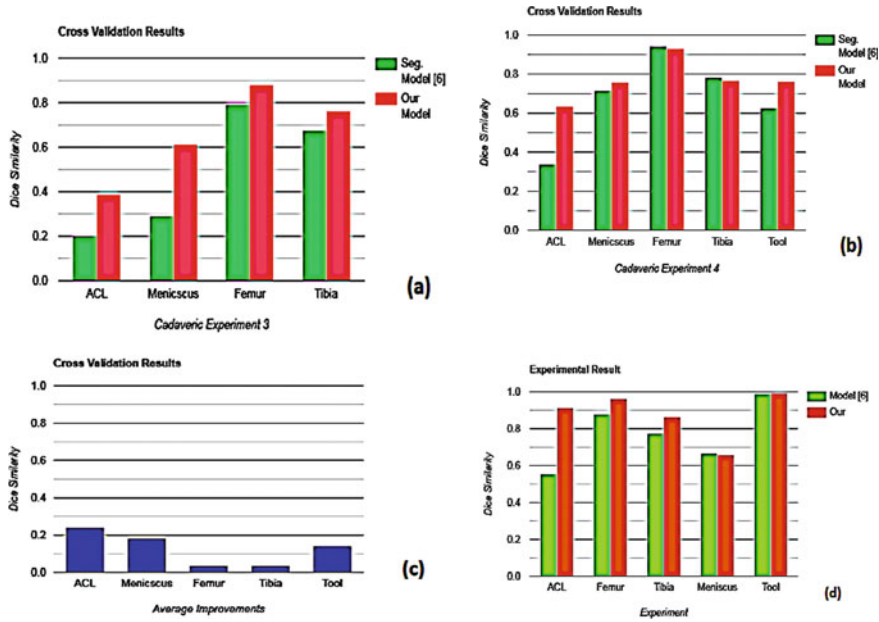


**Fig. 5** Cross-validation results of the segmentation network based on Ref. [6] and our proposed method. As shown in **a**, **b**, our proposed model consistently achieved higher accuracy to recognize tissue type ACL. With our proposed model the average improvements were 0.2415, 0.183, 0.0385, 0.036 for ACL, MENISCUS, FEMUR, and TIBIA, respectively, as shown in **c**

## 6 Conclusion

We proposed a complete segmentation model which learns scene segmentation from both in- and out- distributions. The first stage model learns the segmentation process from a dataset which we called in-distribution learning. The macro model learns where the segmentation model fails and subsequently compensates pixel-wise misclassifications. Final segmented maps are obtained through the fusion of the initial segmented maps and corrected segmented maps when failures are detected. Our proposed method was compared with the previously published methods and we achieved the highest dice similarity coefficient in segmenting key tissue structures and tools. An automatic method like this enables the ability to segment knee arthroscopic scenes in an intra-operative manner has a tremendous impact on surgical outcomes in conventional arthroscopy as well as in robotic arthroscopy. Furthermore, the model provides an error map (confidence in segmentation) which describes where the segmentation model most likely to experience failure has obvious demand in the medical field to gain clinicians' attention.

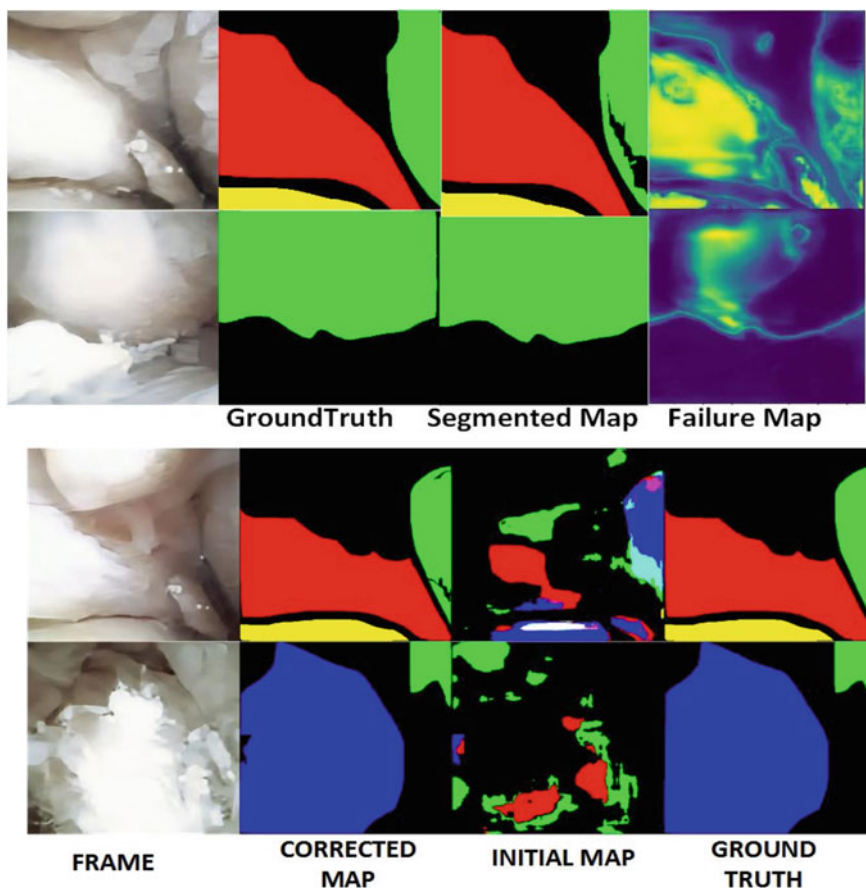


**Fig. 6** Examples frames of segmented maps. Upper image shows segmented maps obtained from our model along with pixel failure maps obtained from initial segmentation model mostly caused by degenerative tissue types, saturations, and shadow in border regions cause failures in prediction (domain shift problem) which were addressed in error maps and subsequently restored segmented maps. The bottom image represents segmented maps obtained from initial network and the corrected obtained from proposed failure and correction model

**Acknowledgements** This work is supported by Australian Indian Strategic Research Fund (Project AISRF53820), The Medical Engineering Research Facility and QUT Centre for Robotics.

## References

1. Jonmohamadi Y, Ali S, Liu F, Roberts J, Crawford R, Carneiro G, Pandey AK (2021) 3D semantic mapping from arthroscopy using out-of-distribution pose and depth and in-distribution segmentation training. In: International conference on medical image computing and computer-assisted intervention (MICCAI). Springer, Cham, pp 383–393
2. Ali S, Jonmohamadi D, Takeda Y, Roberts J, Crawford R, Brown C, Pandey D, Ajay K (2021) Arthroscopic multi-spectral scene segmentation using deep learning. arXiv preprint [arXiv:2103.02465](https://arxiv.org/abs/2103.02465)
3. Shahnewaz A, Jonmohamadi Y, Takeda Y, Roberts J, Crawford R, Pandey AK (2020) Supervised scene illumination control in stereo arthroscopes for robot assisted minimally invasive surgery. *IEEE Sens J* 21(10):11577–11587
4. Ali S, Jonmohamadi Y, Crawford R, Fontanarosa D, Pandey AK (2021) Surgery scene restoration for robot assisted minimally invasive surgery. arXiv preprint [arXiv:2109.02253](https://arxiv.org/abs/2109.02253)
5. Wu L, Jaiprakash A, Pandey AK, Fontanarosa D, Jonmohamadi Y, Antico M, Strydom M, Razjigaev A, Sasazawa F, Roberts J, Crawford R (2020) Robotic and image-guided knee arthroscopy. In: Handbook of robotic and image-guided surgery. Elsevier, pp 493–514
6. Ali S, Crawford P, Maire D, Pandey A, Ajay K (2021) Towards robotic knee arthroscopy: multi-scale network for tissue-tool segmentation. arXiv preprint [arXiv:2110.02657](https://arxiv.org/abs/2110.02657)
7. Ali S, Pandey AK (2022) ArthroNet: monocular depth estimation technique toward 3D segmented maps for knee arthroscopic. *Intell Med*
8. Ali S, Pandey AK (2022) Towards robotic knee arthroscopy: spatial and spectral learning model for surgical scene segmentation. In: Proceedings of international joint conference on advances in computational intelligence. Springer, Singapore, pp 269–281
9. Shahnewaz A, Pandey AK (2020) Color and depth sensing sensor technologies for robotics and machine vision. In: Machine vision and navigation. Springer, Cham, pp 59–86
10. Jansen-van Vuuren RD, Shahnewaz A, Pandey AK (2020) Image and signal sensors for computing and machine vision: developments to meet future needs. In: Machine vision and navigation. Springer, Cham, pp 3–32
11. Jonmohamadi Y, Takeda Y, Liu F, Sasazawa F, Maicas G, Crawford R, Roberts J, Pandey AK, Carneiro G (2020) Automatic segmentation of multiple structures in knee arthroscopy using deep learning. *IEEE Access* 8:51853–51861
12. Gonzalez C, Gotkowski K, Bucher A, Fischbach R, Kaltenborn I, Mukhopadhyay A (2021) Detecting when pre-trained nnu-net models fail silently for covid-19 lung lesion segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 304–314
13. Neal RM (2012) Bayesian learning for neural networks. Vol 118. Springer Science & Business Media
14. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International conference on machine learning. PMLR, pp 1050–1059
15. DeVries T, Taylor GW (2018) Leveraging uncertainty estimates for predicting segmentation quality. arXiv preprint [arXiv:1807.00502](https://arxiv.org/abs/1807.00502)
16. DeVries T, Taylor GW (2018) Learning confidence for out-of- distribution detection in neural networks. arXiv preprint [arXiv:1802.04865](https://arxiv.org/abs/1802.04865)
17. Corbière C, Thome N, Bar-Hen A, Cord M, Pérez P (2019) Addressing failure prediction by learning model confidence. arXiv preprint [arXiv:1910.04851](https://arxiv.org/abs/1910.04851)
18. Zhang P, Wang J, Farhadi A, Hebert M, Parikh D (2014) Predicting failures of vision systems. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3566–3573
19. Kuhn CB, Hofbauer M, Lee S, Petrovic G, Steinbach E (2020) Introspective failure prediction for semantic image segmentation. In: 2020 IEEE 23rd international conference on intelligent transportation systems (ITSC). IEEE, pp 1–6
20. Rahman QM, Sünderhauf N, Corke P, Dayoub F (2022) Fsnet: a failure detection framework for semantic segmentation. *IEEE Robot Autom Lett* 7(2):3030–3037

21. Lv Y, Ma H, Li J, Liu S (2020) Attention guided U-Net with atrous convolution for accurate retinal vessels segmentation. *IEEE Access* 8:32826–32839
22. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1–9
23. Cao Y, Xu J, Lin S, Wei F, Hu H (2019) Gcnets: non-local networks meet squeeze-excitation networks and beyond. In: *Proceedings of the IEEE/CVF international conference on computer vision workshops*

# Chapter 32

## A Multigene Genetic Programming Approach for Soil Classification and Crop Recommendation



Ishrat Khan and Pintu Chandra Shill

**Abstract** The economy of Bangladesh depends to a large extent on agriculture. Besides, a large number of the total population are employed in this sector. In Bangladesh, the population is fast expanding while the overall amount of arable land is constantly diminishing. Because various crops require different soil types, identifying and selecting the proper kind of soil is critical to ensuring optimal crop yield while working with limited land resources. In this study, we present a soil classification method using symbolic regression of multigene genetic programming. Dataset for this work is collected from Soil Resource Development Institute, Government of the people's republic of Bangladesh. GPTIPS toolbox is used to select the appropriate features for training and developing a mathematical model. In a short period of time, the model generates correct results for both the training and testing datasets. Besides, the error rate for soil type classification is extremely low. Finally, suitable crops are recommended based on the accurate classification. According to the results, the proposed multigene genetic programming (MGGP)-based approach performs the best in terms of accuracy, with an accuracy of 98.04 percent. Moreover, our proposed soil classification method outperforms many existing soil classification methods.

**Keywords** Soil classification · Multigene genetic programming · Symbolic regression · Mathematical model · Machine learning

---

I. Khan (✉) · P. C. Shill

Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

e-mail: [ishratriya@gmail.com](mailto:ishratriya@gmail.com)

P. C. Shill

e-mail: [pintu@cse.kuet.ac.bd](mailto:pintu@cse.kuet.ac.bd)

## 1 Introduction

Soil is an essential component in the agricultural field for crop production. Our survival is dependent on food, which is obtained either directly or indirectly from the crop. However, not every type of soil is ideal for every crop [1, 2]. Soil type identification and selection are the first steps in every crop growing process. In Bangladesh, Soil Resource Development Institute (SRDI) has recognized over five hundred soil series [3]. Soil series refers to a set of soils that are made up of the same parent materials and are subjected to the same drainage, vegetation, and climatic conditions. Each soil series has its own name and is called for its location (for example, Gopalpur series, Satla series, Ramgoti series, and so on). It serves as a beginning path for classification of soil in Bangladesh, with the purpose of establishing a program for linking it to international systems for classification of soil (FAO or USDA). Soil series are nominated based on the geographic location belonging to each series to make it easier to distinguish one series from another. And soil samples from various soil series have varied qualities that are appropriate for different crops. In this situation, a soil test is performed to determine the nutrients, composition, and other components present in the soil in order to manually identify which samples correspond to which soil series. The existence of information and actual situations guide soil categorization theories [4]. Chemical analysis was also used to determine the type of soil in the laboratory using different substances. But this is costly, time-consuming, and difficult to get for most farmers. Machine learning-based soil type identification, on the other hand, is based on the varied attributes of the soil [5]. Machine learning is a discipline of computer science that has seen recent advancements and also aids in automating the assessment and processing performed by humans, hence lowering the stress on human power. Instead of human labor, by offering data or the fundamental algorithm, it draws its own conclusions from it.

In this paper, we have proposed a multigene genetic programming approach for soil classification using symbolic regression. For correctly classified soil samples, crops are recommended for specific locations of Bangladesh's Upazila.

This paper is organized as follows. Section 2 describes related works that summarized research in the field of soil classification, Sect. 3 presents overview of multigene genetic programming, methodology for classifying soil samples is explained at Sect. 4 and Sect. 5 describes results analysis including performance comparison with existing methods and crop recommendation following the classification. Finally, Sect. 6 discusses the conclusion and future works.

## 2 Related Works

Soil classification has gotten a lot of interest from the agricultural world for a variety of reasons. Pandith et al. [6] predicted mustard crop production using Naive Bayes, k-Nearest Neighbor (k-NN), Multinomial Logistic Regression, Random Forest, and

Artificial Neural Network (ANN) using authentic data containing 5000 samples from the Soil Testing Laboratory, Directorate of Agriculture Department, Talab Tillo, Jammu. Renuka et al. [7] used k-NN, SVM, and Decision Tree machine learning approaches to estimate sugarcane crop yields using data from Kaggle.com and data.gov.in. Using well-known classification methods such as J48, OneR, BF Tree, and Naive Bayes, Arooj et al. [8] provided a data mining study for soil categorization. The experiment was conducted using data from Pakistan's Kasur area. Taher et al. [9] classified soil data using k-NN, Decision tree, Naive Bayes and Random Forest classifiers and using data from soil science Department, Ahmadu Bello University, Zaria. The dataset consists of 400 soil samples from the North West Zone of Nigeria. Rahman et al. [10] classified the soil data using information from different Upazilas in Bangladesh's Khulna district. They employed weighted k-NN, Bagged trees, and Gaussian SVM, then compared the results of the three algorithms to come up with a model for identifying soil types and the appropriate crop that may be grown in that soil type.

These studies show that machine learning approaches for soil type classification are becoming more popular due to their faster findings and lower costs when compared to laboratory-based chemical-oriented methodologies. However, to the best of our knowledge, all of these previously existing works lack in accuracy level. In this scenario where we need a potential model which can be used for more accurate soil classification. As a result, by correctly identifying soil types, this technology can help to enhance crop production and crops can be recommended for correctly classified soil samples.

### 3 Multigene Symbolic Regression Using GP

Darwin's Natural Selection Principle inspired Genetic Programming [11]. It is applied to a population of individuals. Here, every individual in the population is a potential solution. The initial population is formed at random and gradually developed over generations. Crossover, mutation, elitism, and gene detection are all examples of genetic activities used in evolutionary processes. Figure 1 depicts a flowchart of a genetic algorithm.

GP is a machine learning approach which is supervised that looks for patterns in a program space rather than a data space. A robust variation of GP is multigene genetic programming (MGGP). Each individual in MGGP is made up of one or more genes, which are organized into a typical GP tree. Individuals arrange genes progressively to increase fitness [12]. Weighting coefficients are incorporated with each gene in multigene symbolic regression. The tree structure of the MGGP model with input variables ( $x_1, x_2, x_3$ ) and output variable ( $f$ ) is shown in Fig. 2. A bias term  $b_0$  and weight variables  $b_1, b_2, \dots, b_n$  are used to scale each output gene or tree to increase the accuracy of symbolic regression. A numeric number is added or subtracted from the resulting expression to determine bias [13].

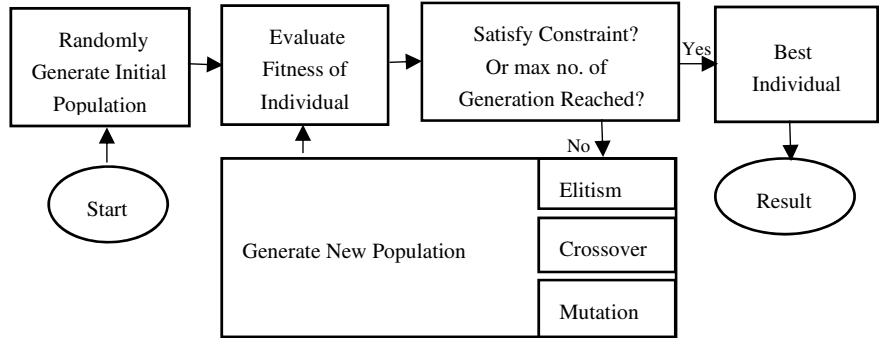


Fig. 1 Work flow of genetic programming

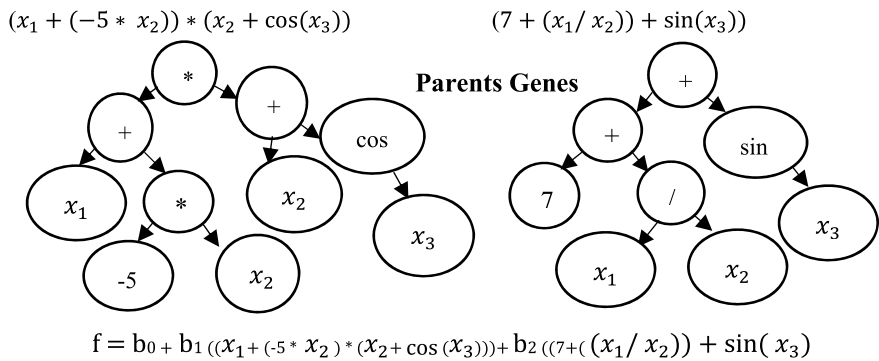


Fig. 2 Tree structure of MGGP model

4 Methodology

The overall functional block diagram of proposed method is illustrated in Fig. 3.

4.1 Data Collection

Two category datasets are collected for conducting our work.

4.1.1 Soil Dataset

To carry out our research, we have collected necessary data regarding soil from Soil Resource Development Institute (SRDI), Government of the people’s republic of Bangladesh [3]. Some Upazilas in the District of Khulna provided us with the data.



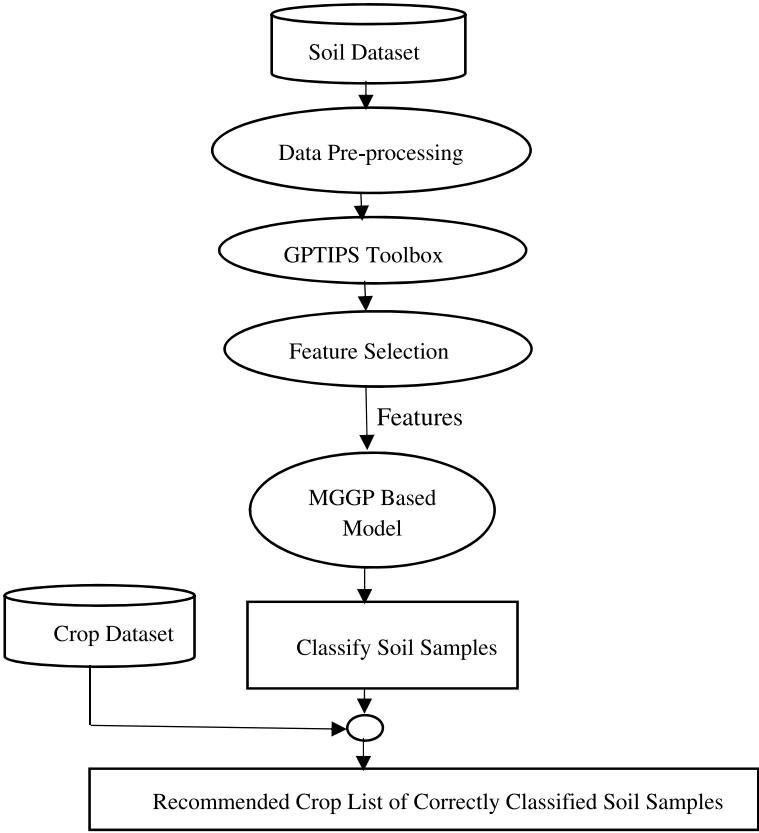


**Fig. 3** Overall functional block diagram of proposed model

'Rupsha', 'Dakop', 'Dighalia', 'Terokhada', 'Fultola', and 'Koyra' are Upazilas. These 6 Upazilas include a total of 15 soil series. We have dealt with nine soil series which have large number of samples in our research and soil classes are formed by combining land types. In order to address the issue of class imbalance, we have eliminated additional soil series with fewer than or equal to 10 samples. And finally, combining the land type along with these nine soil series, total 11 soil classes are obtained. Table 1 shows the soil classes that are employed in the proposed technique.

Soil samples from each of these classes have different chemical features. Finally, the dataset consists of chemical features of different soil samples belonging to different soil classes. Out of 510 samples, 80% of the samples are utilized for training purpose in the model, while the leftover 20% are used for testing purpose.

**Table 1** Soil classes

| Soil classes             | Class labels |
|--------------------------|--------------|
| Ghior mid low land       | 1            |
| Bajoya mid high land     | 2            |
| Harta mid low land       | 3            |
| Barisal mid high land    | 4            |
| Jhalokathi mid high land | 5            |
| Isshwardi mid high land  | 6            |
| Barisal mid low land     | 7            |
| Komolkathi mid high land | 8            |
| Gopalpur high land       | 9            |
| Ghior mid high land      | 10           |
| Dumuria mid high land    | 11           |

**Table 2** Samples of crops for many classes, map units and Upazilas

| Class label | Map unit | Upazila code | List of crops                |
|-------------|----------|--------------|------------------------------|
| 10          | 5        | 200          | Potato, Wheat                |
| 1           | 6        | 100          | Legume                       |
| 4           | 4        | 300          | Seaseme (mixed), Rice (Aush) |

**4.1.2 Crop Dataset**

For crop dataset, the corresponding Upazila code and map unit of corresponding class labels of that six Upazilas of Khulna district are collected from SRDI, Bangladesh [3]. Different sorts of crops thrive in various Upazila regions and soil series. We have gathered all of the data obtained from SRDI and organized it as per their rule. Table 2 lists some instances of crops that are appropriate for different soil types in various map units and show a short instance of the crop dataset as well.

**4.2 Data Preprocessing**

This is the process of modifying the data if it is required. In the soil dataset, certain attributes are missing. Mean, median, etc. value are used to replace the missing values for that attribute. Then randomization is done in dataset to guarantee that the soil data is distributed evenly.

**Table 3** Selected chemical features

| Features               | Description                     |
|------------------------|---------------------------------|
| $x_1$ : pH             | Soil's pH value                 |
| $x_2$ : Salinity       | Ds per meter                    |
| $x_3$ : Organic Matter | Percentage composition          |
| $x_4$ : Calcium        | Mili equivalent per 100 g. soil |
| $x_5$ : Magnesium      | Mili equivalent per 100 g. soil |
| $x_6$ : Potassium      | Mili equivalent per 100 g. soil |
| $x_7$ : Sulfur         | $\mu$ g/gram soil               |
| $x_8$ : Boron          | $\mu$ g/gram soil               |
| $x_9$ : Copper         | $\mu$ g/gram soil               |
| $x_{10}$ : Iron        | $\mu$ g/gram soil               |
| $x_{11}$ : Manganese   | $\mu$ g/gram soil               |
| $x_{12}$ : Zinc        | $\mu$ g/gram soil               |

**4.3    Feature Selection**

Feature selection is required before model creation to exclude irrelevant and duplicate features. GPTIPS toolbox is used, which automatically selects the input attributes that best predict the desired output attributes [14]. In the situation where there are over 1000 irrelevant input attributes, then also GPTIPS has been demonstrated to be successful in feature selection. GPTIPS, on the other hand, may be used as a stand-alone feature selection method to choose features or non-linear combinations of features as inputs which may be applied to different kinds of modeling approaches. Due to this, we have used it in our work. Features selected for this work are shown in Table 3.

**4.4    Inputs and Tuning Parameters of MGGP**

Input set comprises twelve features (Features listed in Table 3). Following function set F have been used to train the model using GPTIPS:

$$F = \{+, -, \sin, \cos, \tan, plog, \text{sqrt}, \text{iflte}, pdiv, gth, lth\}$$

Parameters of tuning employed in MGGP are shown in Table 4.

**Table 4** Tuning parameters of MGGP

| Parameters                 | Detail input |
|----------------------------|--------------|
| Population size            | 1200         |
| Used no. of generation     | 800          |
| Selection procedure        | Tournament   |
| Maximum depth of tree      | 9            |
| Size of tournament         | 10           |
| Category of crossover used | Two point    |
| Number of genes            | 7            |
| Crossover probability      | 0.85         |
| Mutation probability       | 0.10         |
| Elite fraction             | 0.15         |

### 4.5 Discovery of Mathematical Model

The selected chemical features of soil samples are supplied to GPTIPS toolbox and are trained by MGGP using tuning parameters shown in Table 4. It develops a mathematical model at the end of the training period. After that, soil samples are classified, and the model's accuracy is calculated. In MGGP, the fitness function used to verify the quality of fit was the root mean square error (RMSE). The lower the value, the higher the accuracy. The following is the equation for the RMSE:

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (1)$$

## 5 Performance Analysis

The multigene genetic programming (MGGP)-based method is used to classify the soil samples. Table 5 illustrates the result of the proposed method. Out of 102 testing samples, only 2 samples were misclassified.

Throughout the training period, as the number of generations grows, the fitness value is extremely near to zero, indicating a better trained model. Figure 4 depicts a visual representation of fitness versus generation.

As we can see from the figure that it decreases in fitness until it is near enough to zero for the higher generation of the MGGP evaluation procedure. In this work,

**Table 5** Result of proposed MGGP-based method

|                     | Classification model | Accuracy |
|---------------------|----------------------|----------|
| Our proposed method | MGGP-Based model     | 98.04%   |

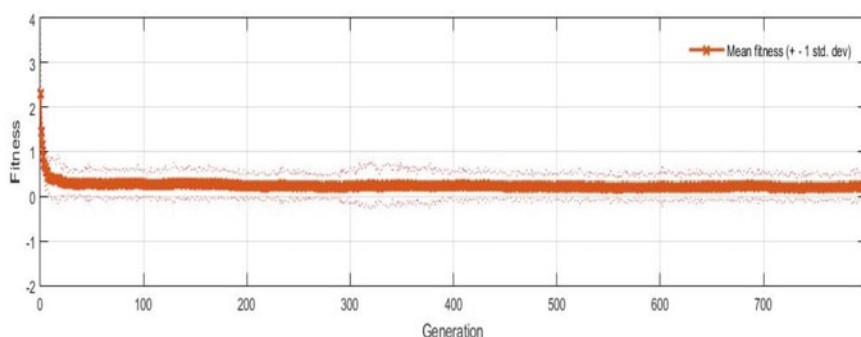


**Fig. 4** Fitness versus generation for soil type classification

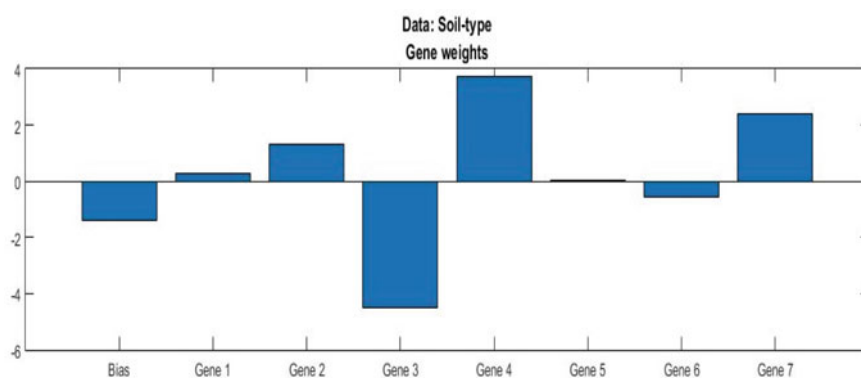


**Fig. 5** Bias and weights of genes

number of generation used is 800. Best fitness value obtained is 0.04724. Figure 5 shows the weight of each gene as well as the bias.

It is seen that different genes have different weights. Here, total number of genes considered are 7 and the bias value is  $-1.397$  for the model which is finally incorporated with the resulting expression. A numeric value is either added or subtracted from the resulting expression to represent bias described in Sect. 3. To suit the outcome of symbolic regression, the bias term and weight for each variable is necessary.

For both training and testing sets, the RMS error was calculated. RMSE rate obtained for training set is 0.047245 and testing set is 0.16619. Figure 6 depicts a visual representation of the RMS error rate. The MGGP model appears to be well-trained, as seen by Fig. 6. Besides, it is seen that the soil type prediction is extremely positive, and the rate of error for the testing data set is extremely low.

Figure 7 shows a comparison of the rate of error of the actual and predicted results for both training and testing cases.

During the training phase, the rate of prediction which is positive for each sample of data is extremely close to the actual outcome. The prediction which is positive

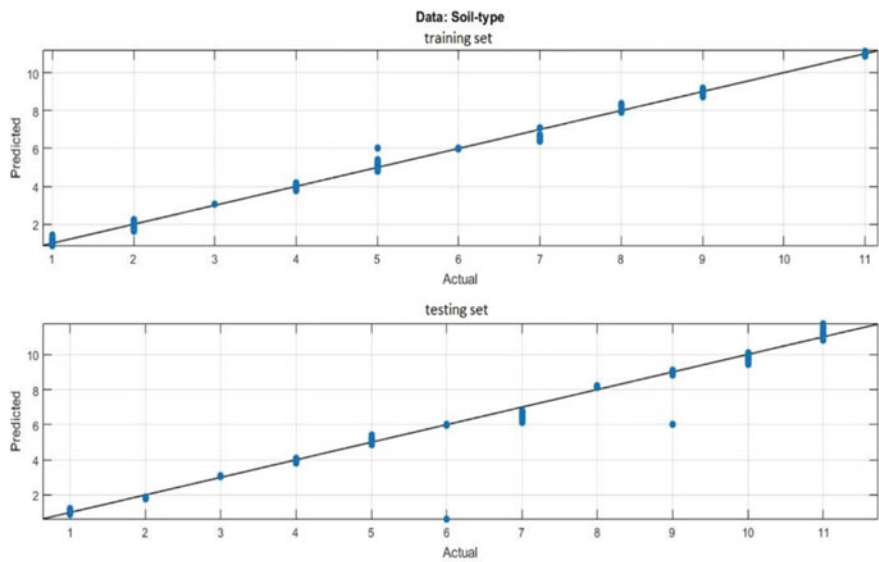


Fig. 6 Illustration of error rate

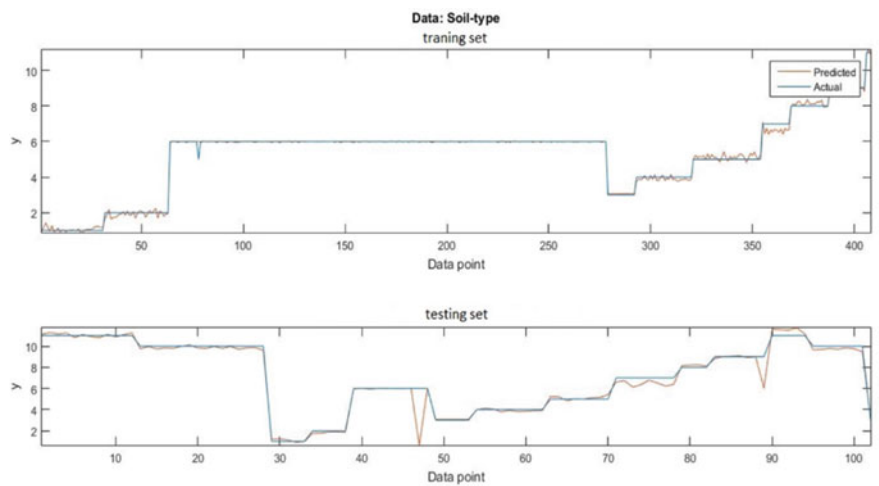


Fig. 7 Illustration of actual as well as predicted result

is also near enough to the actual outcome in testing. It is seen that, certain points in the actual and predicted results that overlap. In both of the cases, the error rate is extremely less which is insignificant. Table 6 displays the model’s mathematical expression. The generated expression combines each gene’s result with the bias term.

**Table 6** Developed mathematical model using MGGP

$$F(x_1, x_2, \dots, x_{12}) = 0.05052 x_1 + 0.05052 x_6 + 0.05052 x_9 + \sin(7.249 \sin(x_8)) 0.05052 + 0.05052 \cos(\sin(x_1)) + 0.05052 \text{plog}(\sin(\text{gth}(x_3 + x_8, \tan(x_6)))) \cos(x_3 + 8.377)) + 0.05052 \sin(\text{gth}(x_9 \sin(x_8), x_8)) + 0.05052 \text{plog}(\sin(\sin(\sin(x_8))) \tan(\text{pdiv}(x_{11}, x_3))) + \cos(\text{gth}(\cos(\text{plog}(x_9 \sin(x_8))), \text{gth}(8.489, 1.341 x_8))) 0.2888 + 0.05052 \text{plog}(\sin(\sin(\text{gth}(8.489, x_9)) \cos(x_3 + x_8)) \cos(x_8 + x_{11})) + \text{plog}(\text{plog}(7.259 \sin(x_9)) \text{plog}(x_2 \sin(x_9)) \tan(\sin(\text{gth}(x_3 + x_9, \tan(x_6)))))) 0.05052 + 0.05052 \cos(x_1) + 0.05052 \sin(x_1) + \text{plog}(\cos(x_3 + 8.377) \text{gth}(2 x_3, \tan(\tan(x_6)))) 0.05052 + 0.05052 \cos(\text{plog}(\sin(x_8))) - 4.505 \text{gth}(8.472, x_2) + \text{pdiv}(x_{10} + \text{plog}(x_{10} \sin(\text{sqrt}(\text{iflte}(x_9, x_3, x_6, x_5)))) \sin(x_9)) + \text{lth}(x_3, \text{pdiv}(x_3, x_8)) + x_3 - 5.055, x_3 + x_{10} + \cos(x_4 + x_8)) 2.386 + 1.3 \text{gth}(\text{gth}(\text{sqrt}(\text{plog}(x_8 \sin(x_8))), x_2), x_9 \text{gth}(8.489, x_9)) + 0.2888 \text{pdiv}(\text{l}_8 \text{l}, \tanh(x_5)) + 3.739 \text{gth}(\cos(x_8 x_9 \sin(x_8)), \text{gth}(\cos(\cos(\text{gth}(0.815, x_3))), x_8)) + 0.05052 \text{lth}(x_4, 7.156) + 0.05052 \text{lth}(x_8, 7.259) + 0.05052 x_{11} \text{gth}(8.377, x_2) + 0.05052 x_3 \text{gth}(8.489, x_2) - \cos(\text{plog}(0.3406 x_8)) 0.5724 + 0.4283 \text{gth}(x_2, \cos(\sin(x_8))) + \text{sqrt}(\text{iflte}(2 x_3, \text{pdiv}(x_3, x_8), \text{gth}(8.472, x_2), x_4)) 0.05052 + 0.05052 \text{lth}(\text{iflte}(x_6, x_8, x_{12}, 8.489), 8.489) - 1.397$$

**Table 7** Performance comparison on same dataset

| Proposed method      |          | Rahman et al. [10] method |          |
|----------------------|----------|---------------------------|----------|
| Classification model | Accuracy | Classification model      | Accuracy |
| MGGP-Based model     | 98.04%   | Weighted k-NN             | 92.93%   |
|                      |          | Bagged Trees              | 90.91%   |
|                      |          | Gaussian SVM              | 94.95%   |

5.1 Comparison with Other Existing Methods

We have chosen a current and well-known study in soil classification to determine the genuine performance of our proposed approach. The identical dataset is used in each of the studies. It is for this reason that we have compared them. Table 5 shows a comparison of accuracy of the proposed method and another existing method using the same dataset (Table 7).

The accuracy of Rahman et al. [10] method using three classification models are compared with our proposed multigene genetic programming (MGGP)-based method. Rahman et al. [10] used weighted k-NN, Gaussian SVM and Bagged tress as classification models and the accuracy of the three approaches are 92.93%, 94.95%, and 90.91%, respectively. The accuracy of the proposed MGGP-based method is 98.04%. Thus, it is seen that our proposed method produces a more dependable model than the previous technique for soil classification.

5.2 Crop Recommendation

Following the correct classification of soil samples, crops suitable for correct classes are recommended using the provided map unit of the related Upazila from crop

**Table 8** List of recommended crops for Ghior mid high land

|                                       |                 |                     |                        |                |
|---------------------------------------|-----------------|---------------------|------------------------|----------------|
| Code of Upazila                       |                 | 200                 |                        |                |
| Name of Upazila                       |                 | Dighalia            |                        |                |
| Map unit                              |                 | 5                   |                        |                |
| Class label                           |                 | 10                  |                        |                |
| Name of class                         |                 | Ghior Mid High Land |                        |                |
| Without irrigation, recommended crops | Rabi Season     | Barley              | Mustard                | Flax           |
|                                       | Kharif 1 Season | Rice (Bona Aush)    | Sesame                 | Foxtail millet |
|                                       | Kharif 2 Season | Rice (Ropa Amon)    | Rice (Deep Water Amon) |                |
| With irrigation, recommended crops    | Rabi Season     | Wheat               | Rice(Boro)             | Potato         |
|                                       | Kharif 1 Season | Rice (Ropa Aush)    |                        |                |
|                                       | Kharif 2 Season | Rice (Bona Amon)    |                        |                |

dataset. Let, Ghior Mid High Land class is considered whose number of map unit is 5 belongs to Dighalia Upazila. Table 8 shows the recommended list of crops for testing sample that are accurately classified for this class.

## 6 Conclusion

Determining the appropriate soil type is crucial for optimal crop production. In this paper, we have proposed a multigene genetic programming (MGGP)-based model to classify soil samples. Soil classification using this method is simple and inexpensive. The soil samples belongs to which soil class are efficiently classified by the proposed MGGP based method. The mathematical model's rate of error is quite low, indicating that the prediction is fairly accurate. Many existing approaches are inadequate in terms of soil classification accuracy and crop recommendation for definite soil type. In the future, we will focus on giving fertilizer recommendations, as well as data from other districts to make this model more dependable and accurate which will greatly improve agriculture sector in Bangladesh.

## References

1. Srivastava P, Shukla A, Bansal A (2021) A comprehensive review on soil classification using deep learning and computer vision techniques. *Multimedia Tools Appl* 80(10):14887–14914
2. Uddin M, Hassan M (2022) A novel feature based algorithm for soil type classification. *Complex Intell Syst* 2(5):1–17
3. Soil Resource Development Institute (SRDI). <http://www.srdi.gov.bd/>. Last accessed 28 May 2022



4. Daryati D, Widiyanti I, Septiandini E, Ramadhan MA, Sambowo KA, Purnomo A (2019) Soil characteristics analysis based on the unified soil classification system. *J Phys Conf Ser* 1402(2):022028
5. Padarian J, Minasny B, McBratney AB (2020) Machine learning and soil sciences: a review aided by machine learning tools. *Soil Discuss* 6(1):35–52
6. Pandith V, Kour H, Singh S, Manhas J, Sharma V (2020) Performance evaluation of machine learning techniques for mustard crop yield prediction from soil analysis. *J Sci Res* 64(2):394–398
7. Renuka ST (2019) Evaluation of machine learning algorithms for crop yield prediction. *Int J Eng Adv Technol* 8(6):4082–4086
8. Arooj A, Riaz M, Akram MN (2018) Evaluation of predictive data mining algorithms in soil data classification for optimized crop recommendation. In: 2018 International conference on advancements in computational sciences (ICACS), IEEE, pp 1–6
9. Taher KI, Abdulazeez AM, Zebari DA (2021) Data mining classification algorithms for analyzing soil data. *Asian J Res Comput Sci* 8(2):17–28
10. Rahman SAZ, Mitra KC, Islam SM (2018) Soil classification using machine learning methods and crop suggestion based on soil series. In: 2018 21st International conference of computer and information technology (ICCIT). IEEE, pp 1–4
11. Koza JRGP (1992) On the programming of computers by means of natural selection. Genetic programming. MIT Press
12. Searson DP (2015) GPTIPS 2: an open-source software platform for symbolic data mining. In: Handbook of genetic programming applications. Springer, Cham, pp 551–573
13. Golap MAU, Hashem MMA (2019) Non-invasive hemoglobin concentration measurement using MGGP-based model. In: 2019 5th International conference on advances in electrical engineering (ICAEE). IEEE, pp 1–6
14. GPTIPS. <https://sites.google.com/site/gptips4matlab>. Last accessed 28 May 2022

# Chapter 33

## Soft Error Tolerant Memristor-Based Memory



Muhammad Sheikh Sadi , Md. Mehedy Hasan Sumon, and Md. Liakot Ali

**Abstract** Memories, which act as data storage devices, are crucial to computer systems. These are extensively used in application-specific integrated circuits and in other microprocessor-based systems where millions of transistors are integrated on a single chip. Since the memories generally account for a large portion of the chip area, these components suffer more space radiation than others. The sensitivity to radiation of semiconductor memories has become a critical issue to ensure the reliability of electronic systems. Recently, memristor emerges as a newly fabricated device that is becoming popular among researchers for its immune to radiation, non-volatility, and good switching behavior. However, research on soft error tolerance in memristor-based memories is still negligible. This paper presents a new method to tolerate soft errors in memristor-based memory. The proposed method can correct single, and double-bit soft errors with lesser information overhead concerning existing techniques.

**Keywords** Soft error · Memristor · Double-bit errors · Transient fault · Memory systems

---

M. S. Sadi (✉) · Md. M. H. Sumon · Md. L. Ali  
Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh  
e-mail: [sadi@iict.buet.ac.bd](mailto:sadi@iict.buet.ac.bd)

Md. M. H. Sumon  
e-mail: [0416312013@iict.buet.ac.bd](mailto:0416312013@iict.buet.ac.bd)

Md. L. Ali  
e-mail: [liakot@iict.buet.ac.bd](mailto:liakot@iict.buet.ac.bd)

## 1 Introduction

Clock frequencies in the several GHz ranges, supply voltages below one volt, and load capacitances of circuit nodes down to femtofarads are the consequences of continuous downscaling of CMOS technologies. Since the width of the components of a transistor within a chip is lowering down in current dimensions, changing the state of the transistor requires very little energy. Hence, the soft error rate in both memory and logic chips is increasing as a result of adopting nanoscale technology [1]. Even, debugging immunity problems is getting increasingly problematic with the diminishing size of electronic systems [2]. Nonetheless, the devices will achieve the major physical cutoff points [3]. This certainty has impelled research on new devices and rationale styles that can conceivably supplant CMOS. Consequently, chip-manufacturing industries are employing researchers to investigate alternative innovations for the evolution of computing devices. A memristor is a highly promising innovation as an alternative [4]. A memristor preserves data in the form of resistance and the data is retained after power has been removed, thus enduring nonvolatile. It is relatively immune to radiation, and it has fewer parasitic side effects. It provides appealing possibilities, for example, huge capacity limit, low power utilization, modest fabrication for memory chips, and so forth. Organizations, for example, Hewlett-Packard and Hynix are expecting to market memristor-based devices within a short period [5]. Despite having these advantages, high manufacturing, and in-field fault rates, including clustered failures, are predicted to be a problem for these new memories (decreasing reliability). These failures can arise in the memory cells, peripherals, and any other places of the memory system. Moreover, critical reliability concerns may arise in these memory systems such as process variability and endurance degradation that are mostly generated from Nano-scale measurement systems in memristor [6]. They may influence the ordinary activity of memory cells; for example, the process variations can deviate within different resistance values, or degrading endurance may carry out dynamic deviation to several resistances due to aging mechanisms [7]. Hence, while interest in memristive devices is sharply increasing, effective application of this technology should address reliability concerns (in addition to its functional property) up to the extent that it can address numerous challenges before the memristor genuinely replaces current memory and computing technologies.

Research on reliability concerns is still in its early stages. Realizing the erroneous actions of memory systems, the development of suitable fault models, and error detection and correction methodologies may improve the reliability of memristor to a great extent. The existing techniques have one major flaw: they are not simultaneous fault recovery approaches [8]. The output value is invalid during the recovery time, and the system operation should be suspended. A small interruption in system functioning is inappropriate in some critical applications, such as the international thermonuclear experimental reactor (ITER), and can lead to undesirable consequences [9].

Hence, this paper develops a methodology that can address the significant challenges arising from reliability concerns faced by modern high dense memristor. The

proposed methodology can detect and correct any combination of two-bit errors in the memristor.

The paper is organized as follows. Section 2 illustrates related work in the area of soft error tolerance. The proposed methodology is illustrated in Sect. 3. Section 4 analyzes the performance of the proposed method compared to the existing ECCS. Section 5 concludes the paper.

## 2 Related Work

Several approaches exist in the literature for tolerating faults in the memristor. These are outlined shortly as follows.

To isolate a malfunctioning memristor, Manem et al. [10] proposed turning off the access transistor. Individual access transistor control, on the other hand, involves significant routing and space requirements.

Usage of excess memristor columns as a replacement for a few columns with added malfunctioning memristor is proposed in [11]. The disadvantage is that this type of design rises the interface complication between the MBC and peripherals.

Software-level optimizations, instead, frequently aggregate retraining of fault-aware networks and mapping among weights. Weight remapping, as well as retraining, can mitigate crossbar faults in deep neural network models [12]. Even though these techniques can offer crossbar fault tolerance, the vast amount of these faults are benign and minimal effects may arise due to these faults [13].

To limit the effect of device variations, analog programming and frequency-modulated signals are proposed in [14]. Noise injection or device variation helps prevent overfitting for simple jobs, but it reduces performance on more complicated tasks, according to Rahimi et al. [15].

Tolerating sneak path errors in memristor-based devices by eliminating sneak paths with the help of Constraint-coding of array bits has been investigated in [16, 17]. The disadvantage of this elimination strategy is: that it sets stricter limits on the content of the crossbar array than is required, thus resulting in a high rate of loss [18].

The fault enumeration technique understands its impact on crossbar memory structures [19]. However, a genuine set of crossbars has a huge number of modeled faults (in tens of millions), making the complete fault simulation process unrealistic.

## 3 the Proposed Methodology

In this section, we have shown a double error correction scheme for any number of bits of the data stream in the memristor-based memory. At first,  $n$  bit data is divided into  $i$  number of blocks. Check bit generators produce check bits for each data block, and the final codeword is formed according to the proposed method. At the beginning of decoding, the received codeword is divided into  $i$  blocks. Syndromes, check bits,

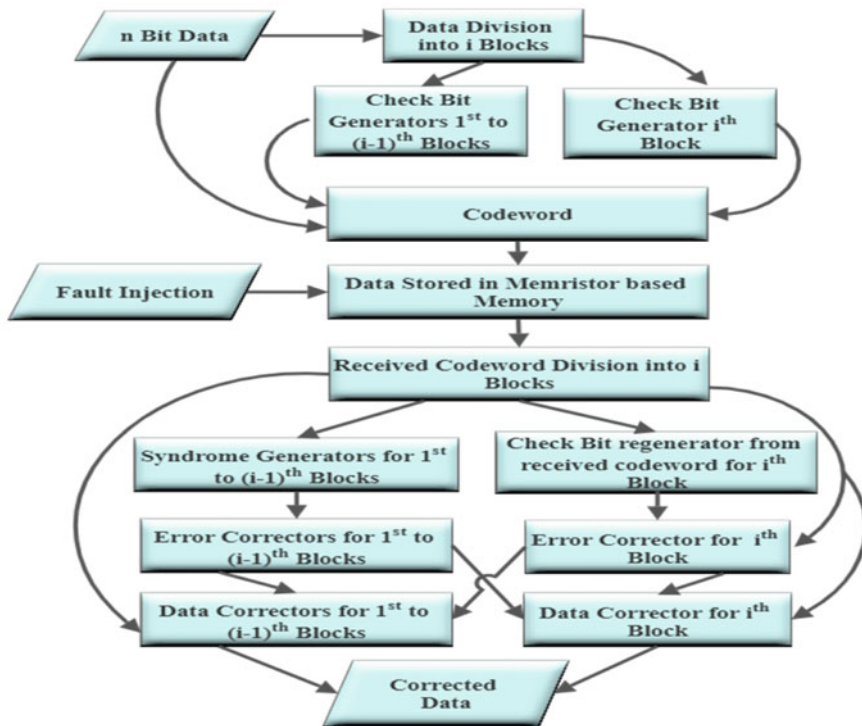


**Fig. 1** Flowchart of the proposed method for double error correction

and error correctors' outputs are generated according to the proposed method. Data correctors are designed to recover original data by using received data and error correctors' outputs. For generating check bits, and error correctors' outputs, two separate techniques are used: one for 1st to  $(i - 1)$ th blocks and the other for  $i$ th block data which have been elaborately discussed in this section. The flowchart of the proposed method for double error correction is illustrated in Fig. 1.

### 3.1 Encoding

The encoding section is accomplished in three steps. In the first step,  $n$  bits data is divided into  $i$  data blocks. If  $n$  is divisible by 11, each block contains 11 bits. Otherwise, the first  $(i - 1)$  rows contain 11 bits in each block, and  $i$ th block contains the remaining bits of the  $n$  bit data stream. The number of data bits in the  $i$ th block is equal to  $11 * i - n \leq 11$ . Data block division for an example of the 32-bit system is shown in Fig. 2.

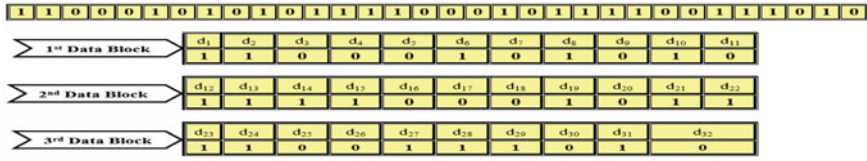


**Fig. 2** Data block division for a 32-bit data stream

The second step is the generation of check bits for each block. Check bits for 1st to  $(i - 1)$ th blocks are generated by Hamming code with additional parity technique for single error correction and double error detection which are given below.

$$\begin{aligned}
 C_1(j) &= d_1(j) \oplus d_2(j) \oplus d_4(j) \oplus d_5(j) \oplus d_7(j) \oplus d_9(j) \oplus d_{11}(j) \\
 C_2(j) &= d_1(j) \oplus d_3(j) \oplus d_4(j) \oplus d_6(j) \oplus d_7(j) \oplus d_{10}(j) \oplus d_{11}(j) \\
 C_3(j) &= d_2(j) \oplus d_3(j) \oplus d_4(j) \oplus d_8(j) \oplus d_9(j) \oplus d_{10}(j) \oplus d_{11}(j) \\
 C_4(j) &= d_5(j) \oplus d_6(j) \oplus d_7(j) \oplus d_8(j) \oplus d_9(j) \oplus d_{10}(j) \oplus d_{11}(j) \\
 C_5(j) &= C_1(j) \oplus C_2(j) \oplus d_1(j) \oplus C_3(j) \oplus d_2(j) \oplus d_3(j) \oplus d_4(j) \\
 &\quad \oplus C_4(j) \oplus d_5(j) \oplus d_6(j) \oplus d_7(j) \oplus d_8(j) \oplus d_9(j) \oplus d_{10}(j) \\
 &\quad \oplus d_{11}(j), \text{ where, } j = 1, 2, 3, \dots, (i - 1).
 \end{aligned}$$

Figure 3 represents a MATLAB Simulink model of the check bit generation for 1st data block where inputs are taken through memristor.

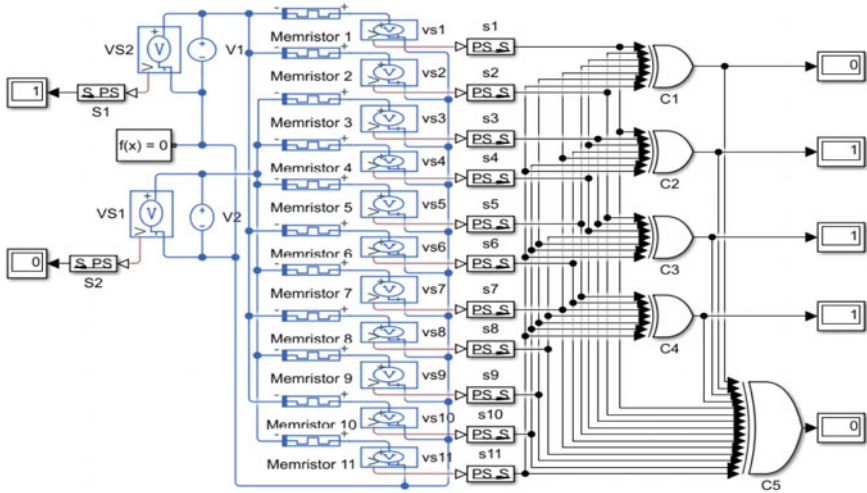


**Fig. 3** MATLAB Simulink model of check bit generation for 1st data block where inputs are taken through memristor



|                                 |                     |                 |                     |                 |                 |                 |                     |                 |                 |                 |                 |                 |                 |                 |                     |
|---------------------------------|---------------------|-----------------|---------------------|-----------------|-----------------|-----------------|---------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------------------|
| C <sub>1</sub> (1)r             | C <sub>2</sub> (1)r | D <sub>1</sub>  | C <sub>3</sub> (1)r | D <sub>2</sub>  | D <sub>3</sub>  | D <sub>4</sub>  | C <sub>4</sub> (1)r | D <sub>5</sub>  | D <sub>6</sub>  | D <sub>7</sub>  | D <sub>8</sub>  | D <sub>9</sub>  | D <sub>10</sub> | D <sub>11</sub> | C <sub>5</sub> (1)r |
| 0                               | 1                   | 1               | 1                   | 1               | 0               | 0               | 1                   | 0               | 1               | 0               | 1               | 0               | 1               | 0               | 0                   |
| 1 <sup>st</sup> Segmented Block |                     |                 |                     |                 |                 |                 |                     |                 |                 |                 |                 |                 |                 |                 |                     |
| C <sub>1</sub> (2)r             | C <sub>2</sub> (2)r | D <sub>12</sub> | C <sub>3</sub> (2)r | D <sub>13</sub> | D <sub>14</sub> | D <sub>15</sub> | C <sub>4</sub> (2)r | D <sub>16</sub> | D <sub>17</sub> | D <sub>18</sub> | D <sub>19</sub> | D <sub>20</sub> | D <sub>21</sub> | D <sub>22</sub> | C <sub>5</sub> (2)r |
| 0                               | 1                   | 0               | 0                   | 1               | 1               | 1               | 1                   | 0               | 0               | 0               | 1               | 1               | 1               | 1               | 1                   |
| 2 <sup>nd</sup> Segmented Block |                     |                 |                     |                 |                 |                 |                     |                 |                 |                 |                 |                 |                 |                 |                     |
| D <sub>23</sub>                 | D <sub>24</sub>     | D <sub>25</sub> | D <sub>26</sub>     | D <sub>27</sub> | D <sub>28</sub> | D <sub>29</sub> | D <sub>30</sub>     | D <sub>31</sub> | D <sub>32</sub> | X <sub>1r</sub> | X <sub>2r</sub> | X <sub>3r</sub> | X <sub>4r</sub> | X <sub>5r</sub> | X <sub>6r</sub>     |
| 1                               | 1                   | 0               | 0                   | 1               | 1               | 1               | 0                   | 1               | 0               | 1               | 1               | 1               | 1               | 1               | 0                   |
| 3 <sup>rd</sup> Segmented Block |                     |                 |                     |                 |                 |                 |                     |                 |                 |                 |                 |                 |                 |                 |                     |
| X <sub>7r</sub>                 | X <sub>8r</sub>     | X <sub>9r</sub> | X <sub>10r</sub>    | X <sub>11</sub> |                 |                 |                     |                 |                 |                 |                 |                 |                 |                 |                     |
| 1                               | 0                   | 1               | 0                   | 1               |                 |                 |                     |                 |                 |                 |                 |                 |                 |                 |                     |

Fig. 5 Received codeword division into segmented blocks

32 bit memristor-based memory by altering the bit-values. For example, the value of the 12th bit (original bit is 1) of the dataword is changed into 0, and that of the 19th bit (original bit is 0) is changed into 1 are two cases where we have injected faults and used in this paper to illustrate the proposed methodology.

### 3.3 Error Correction

Syndrome bits  $S_5, S_4, S_3, S_2$ , and  $S_1$  for 1st to  $(i - 1)$ th blocks are calculated for single error correction and double error detection by the XOR operation of regenerated check bit (from received data bits) and its corresponding check bit at receiving end is calculated by the following formulae.

$$\begin{aligned}
 C_1(j)' &= D_1(j) \oplus D_2(j) \oplus D_4(j) \oplus D_5(j) \oplus D_7(j) \oplus D_9(j) \\
 &\quad \oplus D_{11}(j); \quad S_1(j) = C_1(j)r \oplus C_1(j)'C_2(j)' = D_1(j) \\
 &\quad \oplus D_3(j) \oplus D_4(j) \oplus D_6(j) \oplus D_7(j) \oplus D_{10}(j) \\
 &\quad \oplus D_{11}(j); \quad S_2(j) = C_2(j)r \oplus C_2(j)'C_3(j)' = D_2(j) \\
 &\quad \oplus D_3(j) \oplus D_4(j) \oplus D_8(j) \oplus D_9(j) \oplus D_{10}(j) \\
 &\quad \oplus D_{11}(j); \quad S_3(j) = C_3(j)r \oplus C_3(j)'C_4(j)' = D_5(j) \\
 &\quad \oplus D_6(j) \oplus D_7(j) \oplus D_8(j) \oplus D_9(j) \oplus D_{10}(j) \\
 &\quad \oplus D_{11}(j); \quad S_4(j) = C_4(j)r \oplus C_4(j)'C_5(j)' = C_1(j)r \\
 &\quad \oplus C_2(j)r \oplus D_1(j) \oplus C_3(j)r \oplus D_2(j) \oplus D_3(j) \oplus D_4(j) \\
 &\quad \oplus C_4(j)r \oplus D_5(j) \oplus D_6(j) \oplus D_7(j) \oplus D_8(j) \oplus D_9(j) \\
 &\quad \oplus D_{10}(j) \oplus D_{11}(j); \quad S_5(j) = C_5(j)r \oplus C_5(j)'
 \end{aligned}$$

Generated error corrector's outputs  $\{e1(j)$  to  $e11(j)$  including  $DED(j)\}$  from the syndrome bits are shown in Table 1.

The formulae of computing  $e1(j), e2(j), \dots, e11(j), DED(j)$  for  $j = 1, 2, \dots, (i - 1)$  blocks are as follows.

$$DED = S_5'(S_4 + S_3 + S_2 + S_1), e1 = S_5S_4'S_3'S_2S_1,$$





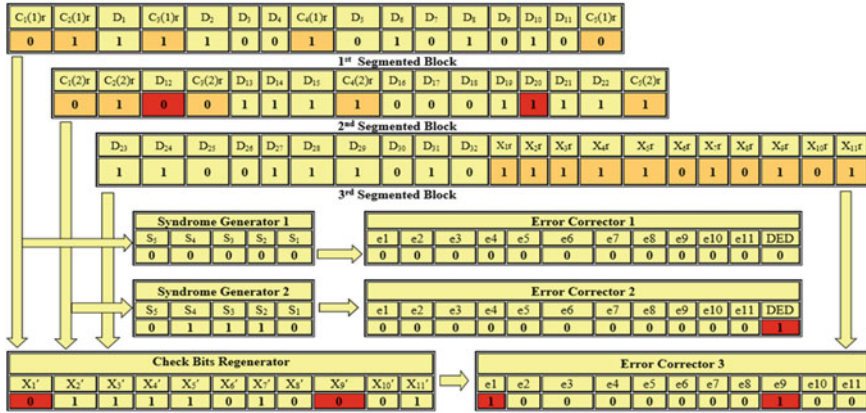


Fig. 6 Error corrector's outputs of the received codeword

Then, we can consider the mentioned example, and calculate  $e_1(j)$ ,  $e_2(j)$ , ...,  $e_{11}(j)$ ,  $DED(j)$  for different blocks. Here, among the syndrome bits  $S_5S_4S_3S_2S_1$  for 1st segmented block, and  $e_1(1)$ ,  $e_2(1)$ , ...,  $e_{11}(1)$ ,  $DED(1)$ , all are 0 which indicates there is no error as shown in Fig. 6. For the 2nd segmented block, the syndrome bit  $S_5$  is zero, and  $S_4S_3S_2S_1$  (=1110) is non-zero which indicates that double errors occur as illustrated in Fig. 6. Therefore,  $e_1(2)$ ,  $e_2(2)$ , ...,  $e_{11}(2)$ —all are 0, but  $DED(2)$  is 1 which indicates a double error is detected. To find the error corrector's outputs of the 3rd block, at first check bits ( $X_1'$ ,  $X_2'$ , etc.) will be regenerated as like encoding process. Then, these bits will be XORed with received check bits ( $X_{1r}$ ,  $X_{2r}$ , etc.) according to their positions, and  $e_1(3)$ ,  $e_2(3)$ , ...,  $e_{11}(3)$  will be computed. Here, we have injected errors in the 12th (1st bit of 2nd segmented block) and 20th (9th bit of 2nd segmented block) data bits. Nevertheless, the proposed method can detect other double-bit errors as well. Therefore, regenerated parity bit  $X_9'$  and received parity bit  $X_{9r}$  are different which will set high  $e_9(3)$  pin.

Similarly, regenerated parity bit  $X_9'$  and received parity bit  $X_{9r}$  are different which set  $e_9(3)$  pin. The other 9 pins  $e_2(3)$ ,  $e_3(3)$ ,  $e_4(3)$ ,  $e_5(3)$ ,  $e_6(3)$ ,  $e_7(3)$ ,  $e_8(3)$ ,  $e_{10}(3)$ ,  $e_{11}(3)$  remain 0. These  $e_1(3)$ ,  $e_2(3)$ , ...,  $e_{11}(3)$  bits will be used with outputs of 1st and 2nd error correctors to correct the double error. The generation of  $e_1(3)$ ,  $e_2(3)$ , ...,  $e_{11}(3)$  is shown in Fig. 6.

### 3.4 Data Correction

Two separate sets of equations are used to correct data: one for 1st to  $(I - i)$ th block, and the other for the  $i$ th block. The required formulae of data correction are shown in Table 2.

**Table 2** The formulae for error correction

| Data correction formulae for $j = 1, 2, \dots, (i - 1)$ th data blocks                                                                                                                                         | Data correction formulae for $i$ th/last data block                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $D_{1C}(j) = [e1(j) * (DED(j))' + e1(i) * DED(j)] \oplus D_1(j)$<br>$D_{2C}(j) = [e2(j) * (DED(j))' + e2(i) * DED(j)] \oplus D_2(j)$<br>$D_{11C}(j) = [e11(j) * (DED(j))' + e11(i) * DED(j)] \oplus D_{11}(j)$ | At first we need to find $Q$<br>$Q = DED(1) + DED(2) + \dots + DED(i - 1);$<br>Then, $D_{1C}(i) = D_1(i) \oplus [e1(1) \oplus e1(2) \oplus \dots \oplus e1(i)] * Q'$<br>$D_{2C}(i) = D_2(i) \oplus [e2(1) \oplus e2(2) \oplus \dots \oplus e2(i)] * Q'$<br>$D_{11C}(i) = D_{11}(i) \oplus [e11(1) \oplus e11(2) \oplus \dots \oplus e11(i)] * Q'$ |
| Here, $D_{1C}(j), D_{2C}(j), \dots, D_{11C}(j)$ are corrected data output for the $j$ th block                                                                                                                 | Here, $D_{1C}(i), D_{2C}(i), \dots, D_{11C}(i)$ are corrected data output for the last block                                                                                                                                                                                                                                                      |

Since,  $\{e1(1), e2(1), \dots, e11(1), DED(1)\}$ —all bits are zero, according the formulae of the 1st coloumn of Table 2, there is no change in data output  $\{D_{1C}(1), D_{2C}(1), \dots, D_{11C}(1)\}$ . From the outputs of error correctors for 2nd data block, and 3rd data block, it is found that  $DED(2), e1(3)$ , and  $e9(3)$  are 1. According to the mentioned formulae, these bits alter the erroneous data bits  $D_1(2)$  and  $D_9(2)$ , and thus corrected data bits  $D_{1C}(2)$  and  $D_{9C}(2)$  are determined. Other bits of 2nd data block remain unchanged. For correcting 3rd data block, at first  $Q$  has been calculated. Here,  $Q$  is equal to 1, and  $Q'$  is equal to 0. According to the formulae of the 2nd coloumn of Table 2, this  $Q'$  does not make any change in the received data bits of 3rd block. We have shown the corrected output for the mentioned example as illustrated in Fig. 7a–c.

## 4 Experimental Analysis

This section is comprised of two sub-sections. The experimental setup is detailed at first and then the results are analyzed.

### 4.1 Experimental Setup

The tools which are used to implement the proposed method are MATLAB R2018a, Intel Core i7 processor, and 16 Gigabytes of RAM. A model has been developed with MATLAB R2018a Simulink to evaluate the performance of the proposed method. A simple memory circuit is simulated by using ideal memristor with nonlinear dopant drift approach. Data encoder, syndrome generator, error corrector, and data corrector circuits have been implemented and used in this model. The model is designed for 32-bit dataword. The proposed model simulation with MATLAB R2018a is simple.

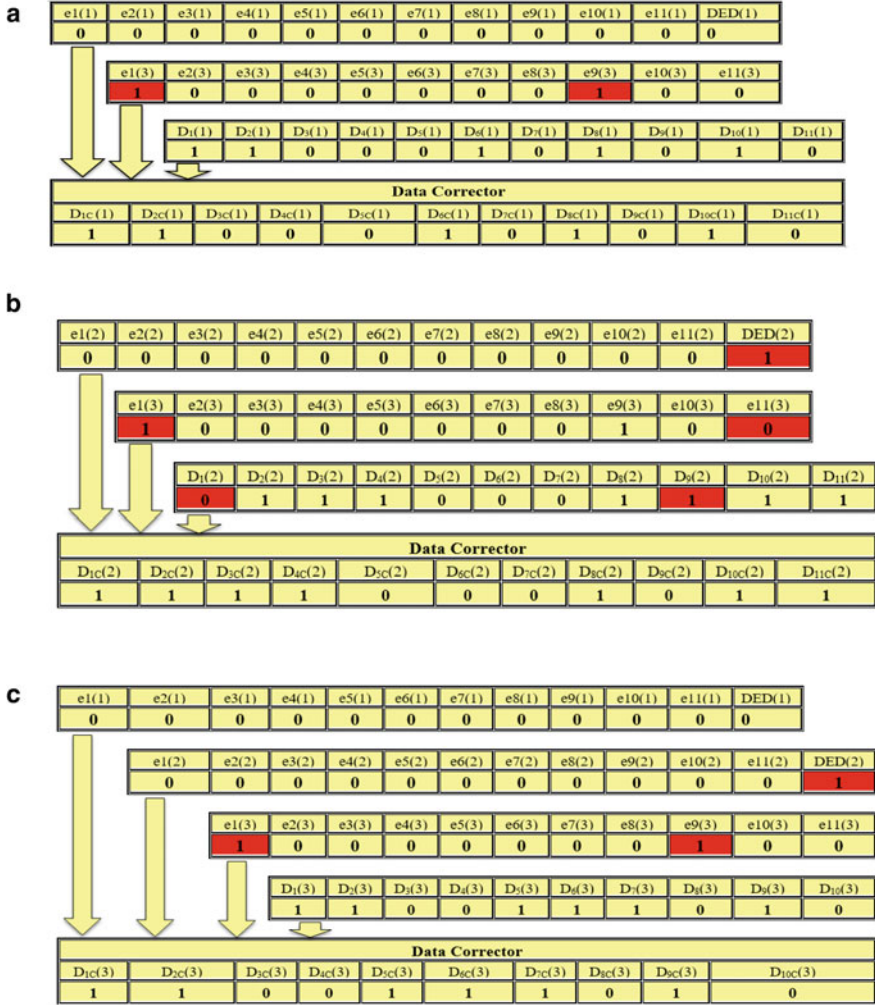


**Fig. 7** **a** Data correction for the 1st block of the received codeword, **b** data correction for the 2nd block of the received codeword, **c** data correction for the 3rd block of the received codeword

However, the hardware implementation with practical memristor is a challenge because its commercial availability is lower still now.

## 4.2 Results

The performance of the proposed methodology is evaluated using a 32-bit dataword. This section compares the proposed method's performance to that of other ECCs like

Hamming code [20], and MC [21] in terms of correction coverage, bit overhead, and so on. The formulae that are used to calculate the bit overhead and code rate, which represent the amount of information redundancy in a coding scheme, are shown in (1), and (2), respectively.

$$\text{Bit Overhead} = \frac{\text{Number of Parity Bits}}{\text{Number of Data Bits}}$$

(1)

$$\text{Code Rate} = \frac{\text{Number of Data Bits}}{\text{Number of Codeword}}$$

(2)

Table 3 depicts a comparison of Hamming code [20], MC [21], and the proposed method. Here, the second row represents the number of original data bits, the third row represents the number of check bits which is needed for error correction, the fourth one represents the number of codeword bits, the fifth and sixth represents bit overhead and code rate respectively. The last two rows represent single and double error correction coverage of the mentioned methods, respectively. For an efficient error correction coding technique, low bit-overhead and higher code rates are required, and error correction coverage should be high as well.

Figure 8 shows the comparison among the code rates of different error correction methods. For the same bit system, the code rate is the highest in Hamming coding and the lowest for the MC coding. The code rate is increased with the increase of data bits for all mentioned methods. The code rates of the proposed method are 53.33%, 57.14%, and 59.25% for 32, 64, and 128 bits, respectively. Besides, the code rate of the MC method is 60.37%, 64.00%, and 65.97% for 32, 64, and 128 bit-systems, respectively. All the mentioned methods can correct single bit error. Hamming code can detect and correct only single bit error and its error correction coverage (when all combinations of errors are considered) is much lower than the MC and proposed method. The MC and proposed method both can correct 100%

**Table 3** Performance comparison among different ECCS

| Parameter                            | Error correction method |       |       |       |       |       |                 |       |       |
|--------------------------------------|-------------------------|-------|-------|-------|-------|-------|-----------------|-------|-------|
|                                      | Hamming                 |       |       | MC    |       |       | Proposed method |       |       |
| Number of data bits                  | 32                      | 64    | 128   | 32    | 64    | 128   | 32              | 64    | 128   |
| Number of check bits                 | 6                       | 7     | 8     | 28    | 48    | 88    | 21              | 36    | 66    |
| Number of codeword bits              | 38                      | 71    | 136   | 60    | 112   | 216   | 53              | 100   | 194   |
| Bit overhead (%)                     | 18.75                   | 10.94 | 6.15  | 87.50 | 75.00 | 68.75 | 65.63           | 56.25 | 51.56 |
| Code rate (%)                        | 84.21                   | 90.14 | 94.12 | 53.33 | 57.14 | 59.25 | 60.37           | 64.00 | 65.97 |
| Single error correction coverage (%) | 100                     | 100   | 100   | 100   | 100   | 100   | 100             | 100   | 100   |
| Double error correction coverage (%) | 0                       | 0     | 0     | 100   | 100   | 100   | 100             | 100   | 100   |

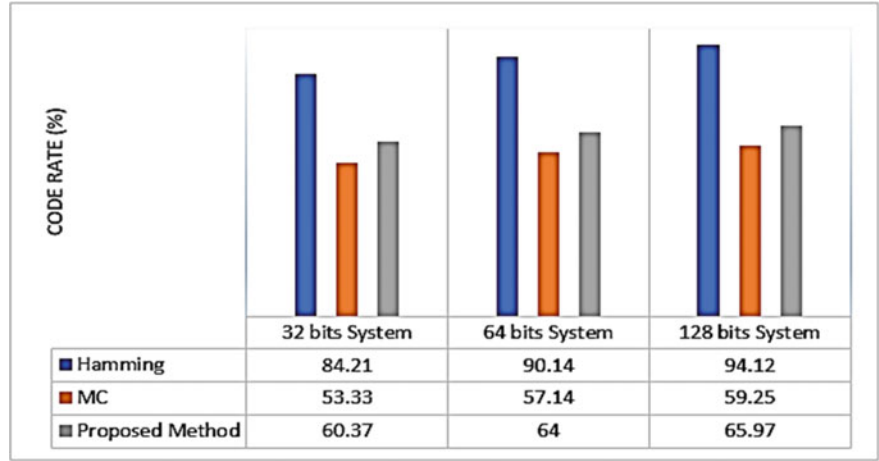


**Fig. 8** Comparison of the code rate of different error correction methods

double-bit errors. Nonetheless, the proposed method has significant higher code rate (lesser information overhead) than MC.

5 Conclusions

This paper shows an efficient soft error tolerant method for memristor-based memory systems. It can correct all double-bit soft errors in any size of dataword. The proposed methodology requires a lesser bit overhead concerning existing ECC. Although the proposed method cannot correct all combinations (e.g., all possible three-bit errors or higher) of soft errors, it is an effective approach where single and/or two-bit errors matter a lot. The proposed methodology can be enhanced to find alternative ways for maximizing the error correction coverage by keeping other constraints unaffected or making a tradeoff among them.

**Acknowledgements** We, the authors, are grateful to the Institute of Information and Communication (IICT), Bangladesh University of Engineering and Technology (BUET) for providing all possible support to perform this research.

## References

1. Siddiqui MSM, Ruchi S, Van Le L, Yoo T, Chang I, Kim TT (2020) SRAM radiation hardening through self-refresh operation and error correction. *IEEE Transa Device Mater Reliab* 20(2):468–474
2. Patnaik A et al (2018) An on-chip detector of transient stress events. *IEEE Trans Electromagn Compat* 60(4):1053–1060
3. Zhimov VV, Cavin RK, Hutchby JA, Bourianoff GI (2003) Limits to binary logic switch scaling—a gedanken model. *Proc IEEE* 91(11):1934–1939
4. Yang S, Adeyemo A, Bala A, Jabir A (2017) Novel techniques for memristive multifunction logic design. *Integration*, 65:219–230
5. Hamdioui S, Taouil M, Haron NZ (2015) Testing open defects in memristor-based memories. *IEEE Trans Comput* 64(1):247–259
6. Pouyan P, Amat E, Rubio A (2018) Memristive crossbar memory lifetime evaluation and reconfiguration strategies. *IEEE Trans Emerg Top Comput* 6(2):207–218
7. Benoist A et al (2014) 28nm advanced CMOS resistive RAM solution as embedded non-volatile memory. In: *IEEE international reliability physics symposium*, pp 2E.6.1–2E.6.5
8. Zandevakili H, Mahani A (2019) Memristor-Based hybrid fault tolerant structure with concurrent reconfigurability. *IEEE Embedded Syst Lett* 11(3):73–76
9. Leroux et al (2014) Design of an MGy radiation-tolerant resolver-to-digital converter IC for remotely operated maintenance in harsh environments. *Fusion Eng Des* 89(9–10):2314–2319
10. Manem H, Rose GS, He X, Wang W (2010) Design considerations for variation tolerant multi-level CMOS/Nano memristor memory. In: *Proceedings of 20th great lakes symposium VLSI (GLSVLSI)*, pp 287–292
11. Liu C et al (2017) Rescuing memristor-based neuromorphic design with high defects. In: *4th ACM/EDAC/IEEE design automation conference (DAC)*, pp 1–6
12. Zhang S et al (2020) Lifetime enhancement for RRAM-based computing in-memory engine considering aging and thermal effects. In: *2nd IEEE international conference on artificial intelligence circuits and systems (AICAS)*, pp 11–15
13. Chen CY, Chakrabarty K (2021) Pruning of deep neural networks for fault-tolerant memristor-based accelerators. In: *58th ACM/IEEE design automation conference (DAC)*, pp 889–894
14. Eshraghian JK et al (2019) Analog weights in ReRAM DNN accelerators. In: *Proceedings of IEEE international conference on artificial intelligence circuits systems (AICAS)*, pp 267–271
15. Rahimi M et al (2020) Complementary metal-oxide semiconductor and memristive hardware for neuromorphic computing. *Adv Intell Syst* 2(5)
16. Cassuto Y, Kvatinsky S, Yaakobi E (2013) Sneak-path constraints in memristor crossbar arrays. In: *IEEE international symposium on information theory*, pp 156–160
17. Sotiriadis PP (2006) Information capacity of nanowire crossbar switching networks. *IEEE Trans Inf Theory* 52 (7):3019–3032
18. Cassuto Y, Kvatinsky S, Yaakobi E (2014) On the channel induced by sneak-path errors in memristor arrays. In: *International conference on signal processing and communications (SPCOM)*, pp 1–6
19. Fieback M et al (2019) Device-aware test: a new test approach towards DPPB level. In *IEEE international test conference (ITC)*, pp 1–10
20. Hamming RW (1950) Error detecting and error correcting codes. *Bell Syst Tech J* 26(2):147–160
21. Argyrides C, Pradhan DK, Kocak T (2011) Matrix codes for reliable and cost-efficient memory chips. *IEEE Trans Very Large Scale Integr (VLSI) Syst* 19(3)

# Chapter 34

## Efficient Malaria Cell Image Classification Using Deep Convolutional Neural Network



Sohag Kumar Mondal, Monira Islam, Md. Omar Faruque,  
Mrinmoy Sarker Turja, and Md. Salah Uddin Yusuf

**Abstract** Malaria is a life-threatening disease that affects millions of people, each year. The way of diagnosing malaria is visually examining blood smears under the microscope by skilled technicians for parasite-infected red blood cells. An automatic malaria classification based on machine learning can boost up the diagnosing process more effectively and efficiently to detect malaria in the earlier stage. In this research, an efficient and accurate convolutional neural network based model is proposed to classify malaria parasitic blood cell from microscope slides as whether infected or uninfected. The customized model trained on the NIH dataset consists of 27,578 RBC images, the accuracy of this model is 99.35% with 99.70% sensitivity and 99.00% specificity. The proposed model shows outform compare to others in terms of the performance indicators such as sensitivity, specificity, precision,  $F_1$  score, and Matthews's correlation coefficient.

**Keywords** Computer aided diagnosis · Convolutional neural network · Malaria · Blood smear · Microscopic RBC

## 1 Introduction

The diseases malaria transmitted to peoples through the infected female Anopheles mosquitoes caused by Plasmodium parasites is a widespread awful life-threatening disease; it still stands as a health concern especially in third world countries. In the year 2017, World Health Organization (WHO) was reported worldwide almost 219 million cases of malaria, across 87 countries [1].

---

S. K. Mondal · M. Islam · Md. O. Faruque · M. S. Turja · Md. S. U. Yusuf (✉)  
Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [suyusuf@eee.kuet.ac.bd](mailto:suyusuf@eee.kuet.ac.bd)

M. Islam  
e-mail: [monira@eee.kuet.ac.bd](mailto:monira@eee.kuet.ac.bd)



Computer aided efficient and accurate diagnostic system could be controlled and prevent malaria more effectively by detecting it in earlier stages [2]. The conventional diagnosis method is the visually examining of blood smears under the microscope for infected RBCs from blood smears by trained technician, however, the traditional process is time inefficient and the detection accuracy depends on the knowledge and experiences of microscopists. Along with manual recognition of malaria parasites, computer-aided automatic malaria detection techniques based on feature extraction by machine learning models is becoming more popular nowadays.

Nowadays, computer-aided diagnosis (CAD) systems are becoming more popular, mostly those are based on image analysis, feature extraction, image segmentation and classification where machine learning (ML) techniques are used [3]. We proposed a five layer CNN model, which is specifically designed for learning two-dimensional data for example images. After being done some pre-processing like resizing, rescaling, normalization and augmentation we put out processed data into our proposed five layers CNN model for training, in forward propagation the trainable parameters were trained and in backward propagation those parameters were updated to minimize losses.

## 2 Related Works

Malaria is one of the major life-threatening diseases over the world that has caused attained deep research interests among researchers [1]. Formerly, mostly diagnosis for detecting malaria was done in a heavily arranged laboratory which required a great deal of human expertise. Automated malaria detection systems can eliminate many of these problems and in recent days a lot of studies are being done to improve the accuracy, efficiency, quality, and complexity.

For automatic classification of malaria parasitic RBC from microscopic images of RBC smear, convolutional neural network (CNN) has taken much attention and become more popular among other machine learning models from the researchers in recent time. Dong et al. introduced the three popular CNN models [4], namely, AlexNet, GoogLeNet and LeNet-5. Furthermore, they trained a SVM classifier for comparison and feature ranking, the authors have shown the advantages of CNN over SVM in terms of learning capacity over input images. In [5], VGG, a transfer learning based pre-trained model was used for feature extraction and classification if infected RBCs. The VGG based model has beaten other prognostic models with 98.22% accuracy, 99.1% precision, 97.81% sensitivity, 98.71% specificity, and F1 score of 98.37%. Zhaohui et al. evaluated the performance of their 9 layers custom-built CNN architecture [2]. The authors reported 97.37% accuracy, 96.99% sensitivity, 97.75% specificity, 97.73% precision, and 97.36%  $F_1$  score. In [6], the authors also used the Binary model (SM vs nMI) approach to evaluate the performance got 96% accuracy. OSICNN Model used by Vikas et al. for detecting malaria parasite achieved 98.3% accuracy [3]. A customized CNN model constructed with four convolutional layers and two dense layers was used to detect infected RBC with an accuracy of 96.46%

[7]. An Attentive Dense Circular Net (ADCN) approach to detect parasite RBCs was used by Quan et al. with 97.47% accuracy, 97.86% sensitivity, 97.07% specificity, and 97.50%  $F_1$  score [8].

Most of the researchers focused on improving the accuracy over validation and testing set by using conventional machine learning and deep learning algorithms in their studies. A few exceptions in the works Quinn et al. [9] who also considered the issue of computational complexity and efficiency of their models. Though both approaches experience a considerable drop in model accuracy in pursuit of computation efficiency. Rosado et al. reported sensitivity of 80.5% and specificity of 93.8% for trophozoite detection and 98.2% and 72.1%, respectively, for WBC using SVM.

### 3 Proposed Methodology

#### 3.1 Cell Image Dataset of Malaria Infection

In this work, the authors used NIH (National Institute of Health) dataset which is publicly available on NID website consist of 27,558 segmented RBC images. The dataset was divided into two parts; those are named parasitized and uninfected which has an equal number of instances. Using a CCD (charge-coupled device) the images were generated by placing it on the conventional light microscope, the data was taken from 200 subjects of two groups parasitized and uninfected [10]. In this dataset there were exist some miss labeled data, some of the images were labeled as parasitized when those are uninfected and vice-versa [1]. Some samples of NIH dataset images are shown in Fig. 1, the parasitic RBC images are shown in Fig. 1a and uninfected images are shown in Fig. 1b.

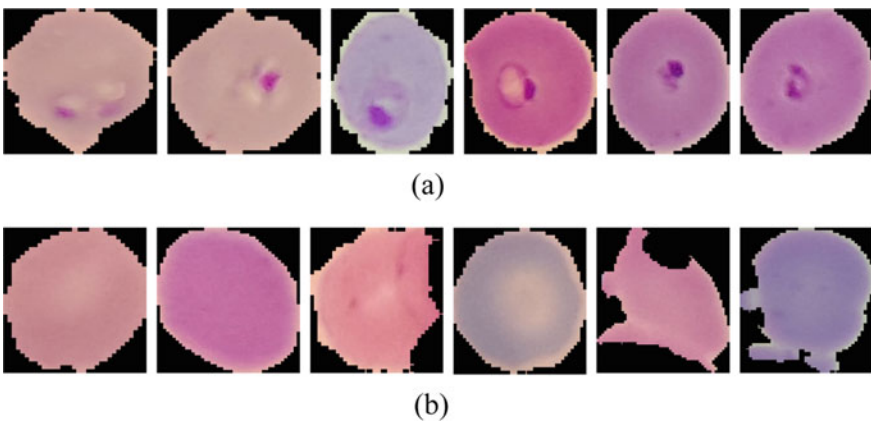


**Fig. 1** **a** Parasitized blood smear images and **b** Uninfected blood smear images

**Table 1** Data augmentation parameters

| Augmentation parameter | Value          |
|------------------------|----------------|
| Share range            | 0.2            |
| Zoom range             | 0.2            |
| Random rotation        | 20°            |
| Width shift            | (0.05, − 0.05) |
| Height shift           | (0.05, − 0.05) |
| Horizontal flip        | True           |

**3.2 Data Processing and Normalization**

Preprocessing on the input images is required to reduce variations of input images that would increase the subsequent processing steps, training time and computational complexity. Before we insert the data into our CNN model we apply rescaling into our entire image dataset from its original size to  $224 \times 224$  on RGB channel. We normalized each pixel of our images to enhance local brightness and contrast and at last, we applied the data augmentation method to introduce variability in the dataset. Table 1 shows the data augmentation summary which has been applied in the data preprocessing step. The total dataset was divided into three sets for training, testing, and validating the model; there were 70% data in the training set, 15% in the validation set, and the remaining 15% were in the testing set.

**3.3 Proposed Model Architecture**

For automatic malaria detection classification shows a very vital role. In a complete convolutional neural networks (CNN) it has two parts based on its functionality, one for feature extraction and another is for classification. Generally, the hidden layers including convolution and pooling layers perform feature extraction and activation functions for classification.

In this work, to detect and classify malaria parasitic RBC we designed a custom deep CNN architecture from scratch. The custom architecture has consist of a total of 5 layers, with 3 convolution layer and 2 dense layers, three convolution layers with 45% dropout to reduce overfitting as shown in Fig. 2. In our model, we use Rectified Linear Unit (ReLU) as activation function in intermediate layers and Softmax in the output layer. In the convolutional operation, a Kernal of size  $2 \times 2$  with padding set to same and used pool size of  $2 \times 2$ . To calculate the error and update the weight and biases, while the model performing back propagation categorical cross entropy, shown in Eq. (1) was used as a loss function (Table 2).

$$\text{Loss} = -(y \log(p) + (1 - y)\log(1 - p))$$

(1)

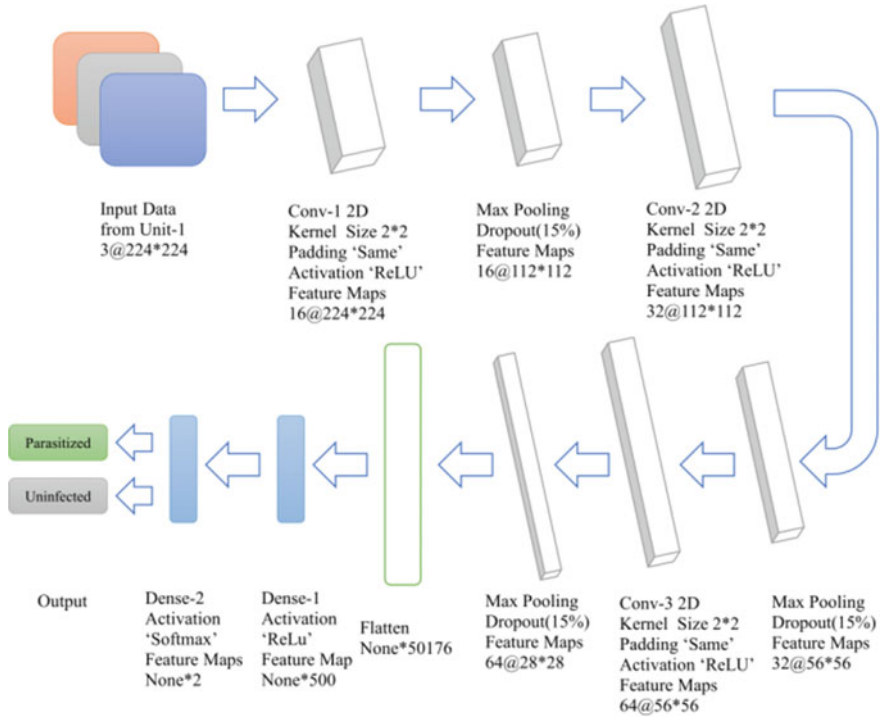


Fig. 2 Proposed five layer CNN custom structure for malaria classification

Table 2 Hype parameter values for the model setting

| Hyper parameters | Type/Value                |
|------------------|---------------------------|
| Epochs           | 50                        |
| Batch size       | 32                        |
| Optimizer        | Adam                      |
| Loss function    | Categorical cross entropy |
| Metrics          | Accuracy                  |
| Input size       | 224 × 224                 |
| Pooling          | 2 × 2                     |

## 4 Results and Analysis

### 4.1 Model Outcomes

The performance evaluation parameter of our proposed CNN architecture is accuracy, sensitivity, specificity, precision,  $F_1$  score, and Matthews's correlation coefficient which is shown in Table 3. In the testing set, there were a total of 4019 cell images

from both instances, among those, 2022 and 1971 images were correctly detected as parasitic and uninfected RBC, respectively, by the model, 20 cell images were falsely detected as uninfected and 6 of those has been detected as parasitic when those were uninfected. The confusion matrix based on the testing set is shown in Fig. 3. The performance of our proposed model has been evaluated based on some criteria that are given below. Accuracy, sensitivity, specificity, precision,  $F_1$  score, negative predictive value, false positive rate, false discovery rate, false negative rate, and Matthews correlation coefficient are the measuring parameters has been calculated using the equations given below based on our performance matrix shown in Fig. 3. The performance of the model on both training and validation set is shown in Fig. 4, in Fig. 4a shown training accuracy and validation accuracy versus the number of epochs, where Fig. 4b shows training loss and validation loss versus the number of epochs.

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

$$\text{Sensitivity} = TP/(TP + FN) \quad (3)$$

$$\text{Specificity} = \frac{TN}{(FP + TN)} \quad (4)$$

$$\text{Precision} = TP/(TP + FP) \quad (5)$$

$$F1 \text{ Score} = 2TP/(2TP + FP + FN) \quad (6)$$

$$\text{Negative Predicted Value} = TN/(TN + FN) \quad (7)$$

$$\text{False Positive Rate} = FP/(FP + TN) \quad (8)$$

$$\text{False Discovery Rate} = FP/(FP + TP) \quad (9)$$

$$\text{False Negative Rate} = FN/(FN + TP) \quad (10)$$

Matthews Correlation Coefficient

$$= \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (11)$$

**Table 3** Outcomes of the proposed model

| Measuring parameter              | Accuracy (%) |
|----------------------------------|--------------|
| Accuracy                         | 99.35        |
| Sensitivity                      | 99.70        |
| Specificity                      | 99.00        |
| Precision                        | 99.02        |
| $F_1$ Score                      | 99.36        |
| Negative predictive value        | 99.70        |
| False positive rate              | 1.00         |
| False discovery rate             | 0.98         |
| False negative rate              | 0.30         |
| Matthews correlation coefficient | 98.71        |

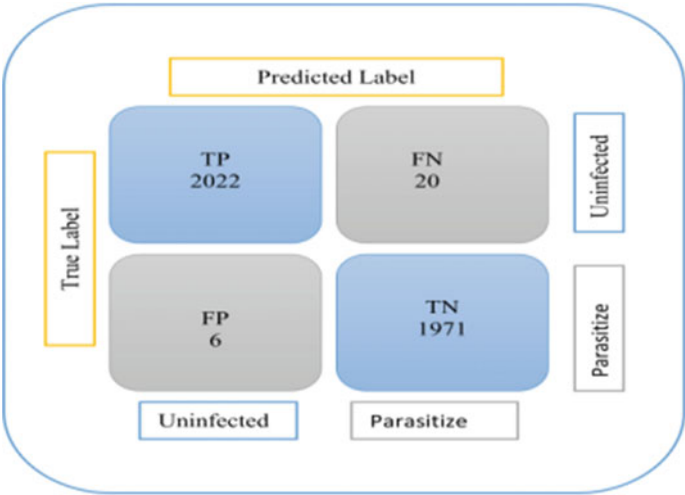


**Fig. 3** Confusion matrix for the testing phase in malaria classification

4.2 Data Processing and Normalization

See Table 3 and Fig. 4.

4.3 Performance Comparison of the Proposed Model with State of Art

To evaluate the performance of different classification models, Quan et al. [8] implemented six CNN models on the same dataset including ResNet50 [11], DenseNet121

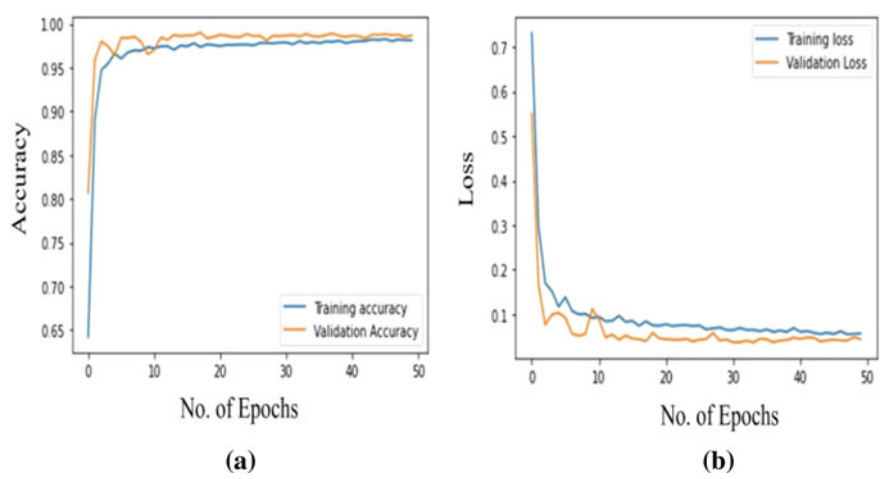


**Fig. 4** **a** Epoch versus classification accuracy, **b** No. of Epoch versus loss on both training and validation set

**Table 4** Comparison results with state of arts

| Model Name          | Accuracy (%) | Sensitivity (%) | Specificity (%) | $F_1$ -Score (%) | Trainable parameters |
|---------------------|--------------|-----------------|-----------------|------------------|----------------------|
| ResNet50 [11]       | 88.47        | 89.61           | 87.31           | 88.57            | $23.6 \times 10^6$   |
| DenseNet121 [12]    | 90.94        | 92.51           | 89.33           | 91.03            | $7.04 \times 10^6$   |
| DPN92 [13]          | 87.88        | 86.81           | 88.98           | 87.85            | $37.7 \times 10^6$   |
| Customized CNN [14] | 94.00        | 93.10           | 95.10           | 94.10            | $0.43 \times 10^6$   |
| Designed CNN [15]   | 94.61        | 95.20           | 92.87           | 94.34            | $0.51 \times 10^6$   |
| Our proposed model  | 99.35        | 99.70           | 99.00           | 99.36            | $10 \times 10^6$     |

[12], DPN92 [13], Customized CNN [14], Designed CNN [15] and evaluated the corresponding results shown in Table 4, our findings and results of our proposed model also mentioned in the same table.

#### 4.4 Graphical User Interface (GUI) Design

Using Gradio, an open source python library a Graphical User Interface (GUI) has been developed and deployed on web. The user interface features the ability to drag and drop RBC images to check whether the cell image is affected in malaria or not.

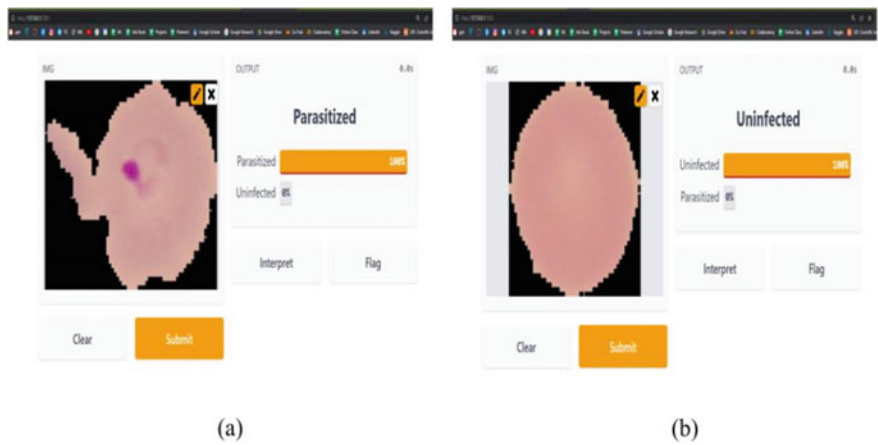


**Fig. 5** **a** Graphical user interface detected parasitized RBC, **b** Graphical user interface detected uninfected RBC

In Fig. 5, it's showing our designed Graphical User Interface. It has a drop down menu to select images from computer local disk, a submit button to turn this image into the machine learning model and a clear button to clear the images and result.

## 5 Discussion

In this work, a highly accurate and time-efficient CNN structure is developed to detect malaria parasites from microscopic RBC. Our proposed CNN model is simpler in architecture, it is only five layers deeper which has three trainable convolutional layers and 2 trainable dense layers, where other transfer learning-based models, for example, Resnet50 has 50 layers with 23 million trainable parameters. The accuracy of our proposed model is 99.35% where the accuracy is Resnet50 is 88.47% on the same dataset [8] shown in Table 4. In Table 4, the results show that the proposed CNN model has exhibited better performance compared to the other transfer learning based and custom model.

## 6 Conclusion

In this work, we proposed a simple, accurate and time efficient model for malaria parasite detection, during modeling this it takes into consideration not only classification accuracy but also computational efficiency which is why we developed a time efficient simple model. The proposed model has higher precision and faster convergence speed, lesser trainable layers, and parameters. The proposed CNN architecture



is the best performing model, achieving an accuracy of 99.35%, the sensitivity of 99.70% with 99.36% F1 score when it trained on  $224 \times 224 \times 3$  images which is comparably best among the model which has been developed on the same datasets in reviewed literatures. Some image data in the NIH dataset was erroneous including label error, when we look at it closely we see some of the uninfected fresh RBC images are placed in a parasitized category, and with help of a local expert the dataset has been corrected.

An implementation of automated end to end diagnosis of malaria parasite using cell-image of blood smear images is our future aspect.

## References

1. Fuhad KM, Tuba JF, Sarker M, Ali R, Momen S, Mohammed N, Rahman T (2020) Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application. *Diagnostics* 10(5):329
2. Liang Z, Powell A, Ersoy I, Poostchi M, Silamut K, Palaniappan K, Guo P, Hossain MA, Sameer A, Maude RJ, Huang JX (2016) CNN-based image analysis for malaria diagnosis. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp 493–496
3. Kashtriya V, Doegar A, Gupta V, Kashtriya P (2019) Identifying malaria infection in red blood cells using optimized stepincrease convolutional neural network model. *Int J Innovative Technol Exploring Eng* 8(9S):813–818
4. Dong Y, Jiang Z, Shen H, Pan WD, Williams LA, Reddy VV, Benjamin WH, Bryan AW (2017) Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In: 2017 IEEE EMBS international conference on biomedical and health informatics (BHI). IEEE, pp 101–104
5. Chakradeo K, Delves M, Titarenko S (2021) Malaria parasite detection using deep learning methods. *Int J Comput Inf Eng* 15(2):175–182
6. Morang'a CM, Amenga-Etego L, Bah SY, Appiah V, Amuzu DS, Amoako N, Abugri J, Oduro AR, Cunningham AJ, Awandare GA, Otto TD (2020) Machine learning approaches classify clinical malaria outcomes based on haematological parameters. *BMC Med* 18(1):1–16
7. Masud M, Alhumyani H, Alshamrani SS, Cheikhrouhou O, Ibrahim S, Muhammad G, Hossain MS, Shorfuzzaman M (2020) Leveraging deep learning techniques for malaria parasite detection using mobile application. *Wirel Commun Mobile Comput*
8. Quan Q, Wang J, Liu L (2020) An effective convolutional neural network for classifying red blood cells in Malaria diseases. *Interdisc Sci Comput Life Sci* 12:217–225
9. Rosado L, Da Costa JMC, Elias D, Cardoso JS (2016) Automated detection of malaria parasites on thick blood smears via mobile devices. *Procedia Comput Sci* 90:138–144
10. Quinn JA, Nakasi R, Mugagga PK, Byanyima P, Lubega W, Andama A (2016) Deep convolutional neural networks for microscopy-based point of care diagnostics. In: *Machine learning for healthcare conference*. PMLR, pp 271–281
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778
12. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4700–4708
13. Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J (2017) Dual path networks. *Adv Neural Inf Process Syst* 30

14. Liang Z, Powell A, Ersoy I et al (2016) CNN-based image analysis for malaria diagnosis. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 493–496, Shenzhen, China
15. Mohanty I, Pattanaik PA, Swarnkar T (2018) Automatic detection of malaria parasites using unsupervised techniques. In: International conference on ISMAC in computational vision and bio-engineering. Springer, Cham, pp 41–49

## Chapter 35

# Detection of Acute Myeloid Leukemia from Peripheral Blood Smear Images Using Transfer Learning in Modified CNN Architectures



Jeba Fairouz Rahman and Mohiuddin Ahmad 

**Abstract** Acute myeloid leukemia (AML), the most fatal hematological malignancy, is characterized by immature leukocyte proliferation in the bone marrow and peripheral blood. Conventional diagnosis of AML, performed by trained examiners using microscopic images of a peripheral blood smear, is a time-consuming and tedious process. Considering these issues, this study proposes a transfer learning-based approach for the accurate detection of immature leukocytes to diagnose AML. At first, the data was resized and transformed at the pre-processing stage. Then augmentation was performed on training data. Finally, the pre-trained convolutional neural network architectures were used with transfer learning. Transfer learning through modified AlexNet, ResNet50, DenseNet161, and VGG-16 were used to detect immature leukocytes. After model training and validation using different parameters, models with the best parameters were applied to the test set. Among other models, modified AlexNet achieved 96.52% accuracy, 94.94% AUC and an average recall, precision, and  $F_1$ -score of 97.00%, 97.00%, and 97.00%, respectively. The investigative results of this study demonstrate that the proposed approach can aid the diagnosis of AML through an efficient screening of immature leukocytes.

**Keywords** Acute myeloid leukemia · Immature leukocyte · Transfer learning · Convolutional neural network · AlexNet

---

J. F. Rahman (✉)

Department of Biomedical Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

e-mail: [rahman2215556@stud.kuet.ac.bd](mailto:rahman2215556@stud.kuet.ac.bd)

M. Ahmad

Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

e-mail: [ahmad@eee.kuet.ac.bd](mailto:ahmad@eee.kuet.ac.bd)

## 1 Introduction

Acute myeloid leukemia (AML) is the most fatal hematological malignancy (HM) among the four leukemia subtypes. It is the most frequent (28.3%) HM in Bangladesh and the onset of AML (64.6%) is predominately observed in young adults aged 20–49 years [1]. AML is characterized by clonal growth of myeloid blast cells in the bone marrow and peripheral blood. Immature white blood cells inhibit the bone marrow from performing its function, such as producing red blood cells and platelets, leaving the immune system susceptible [2, 3]. Efficient screening and classification of immature leukocytes are crucial for AML diagnosis.

Due to rapid progression, if not recognized and treated early, AML can be fatal in months or even weeks [4]. The conventional way of diagnosing leukemia is a microscopic examination of peripheral blood smears, however alternative methods are also employed [5]. The conventional diagnosis procedure is labor-intensive and time-consuming. As a result, because of overpopulation and insufficient manpower in the healthcare sector especially in the developing countries, it almost takes weeks to distribute the results. An automated method of immature leukocyte detection and classification can aid the diagnosis of AML, overcoming the limitations of the conventional diagnosis procedure, particularly in developing countries.

Several machine learning (ML)-based techniques were used in recent times to fulfill this objective of automated or semi-automated healthcare diagnosis. In a recent study [6], random forest (RF) achieved 92.99% accuracy in classifying mature and immature leukocytes and 93.45% accuracy in classifying four subtypes of AML cells, where the proposed method segmented each image using Multi-Otsu thresholding, extracted sixteen features from the cell and nucleus mask, identified the most important features using Gini importance. In [7], 93.57% accuracy was achieved for classifying three subtypes of AML using a momentum backpropagation neural network, where six features were extracted after image segmentation using the active contour. In [8], the authors used k-means clustering-based image segmentation and extracted ten morphological features from cell mask and nucleus mask, and finally achieved 95.89% accuracy for healthy and AML blood cell classification. RF with synthetic minority oversampling technique (SMOTE) was proposed in [9], for classifying three immature leukocytes with 90% accuracy, where Otsu thresholding was used for segmentation and three features were extracted. In [10], a framework was proposed where segmentation was done using k-means clustering, sixty cytomorphological features were extracted and finally, the support vector machine (SVM) achieved the best accuracy of 95% for detecting AML cells and 87% for four subtypes of AML, respectively.

Using ML is time-consuming for controlling big amounts of data as it involves several steps e.g., image segmentation, feature extraction, etc., and may give erroneous results. Furthermore, diverse cytomorphological behaviors of leukocytes in AML are responsible for lowering the performance of ML models.

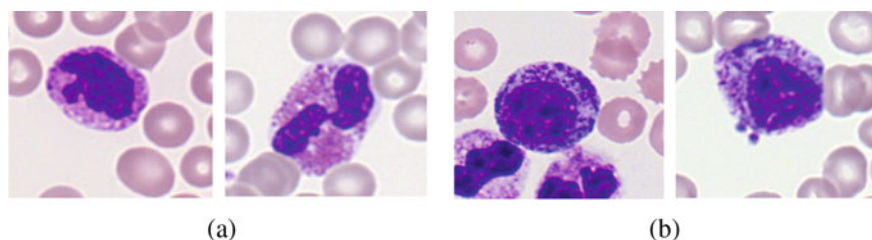


**Fig. 1** Sample images of **a** Mature leukocytes, and **b** Immature leukocytes

Addressing these issues, this study proposes a transfer learning approach with slight modification of pre-trained CNN models to accurately detect immature leukocytes for diagnosing AML. Using pre-trained CNN models saves time since no segmentation and feature engineering is required. Furthermore, they do not require a large dataset to provide fruitful results. Transfer learning was applied for training the CNN models with pre-trained weights. Different hyperparameters and empirical architectures were selected to achieve the best performance.

The rest of this paper is organized as follows: Sect. 2 describes the proposed methodology in several subsections. Section 3 illustrates the results in detail. Finally, Sect. 4 concludes the paper with future works.

## 2 Methodology

### 2.1 Dataset

The dataset [11] was collected from the cancer imaging archive [12]. It contains 18,365 single-cell images of leukocytes with ground truth labels from the peripheral blood smears of 100 AML patients and 100 healthy patients. Figure 1 shows typical images of the mature and immature leucocytes present in the dataset.

### 2.2 Proposed Architecture

The conceptual workflow of the proposed architecture is illustrated in Fig. 2. Morphological images of leukocytes were refined during pre-processing (Sect. 2.3). Then the training set was fed to CNN models with pre-trained weights. Finally, models with the best parameters were evaluated on the test set for performance evaluation. The final output is the type of leukocyte (mature or immature), required for the diagnosis of AML. The python programming language with various python and PyTorch APIs was used to implement the architecture.

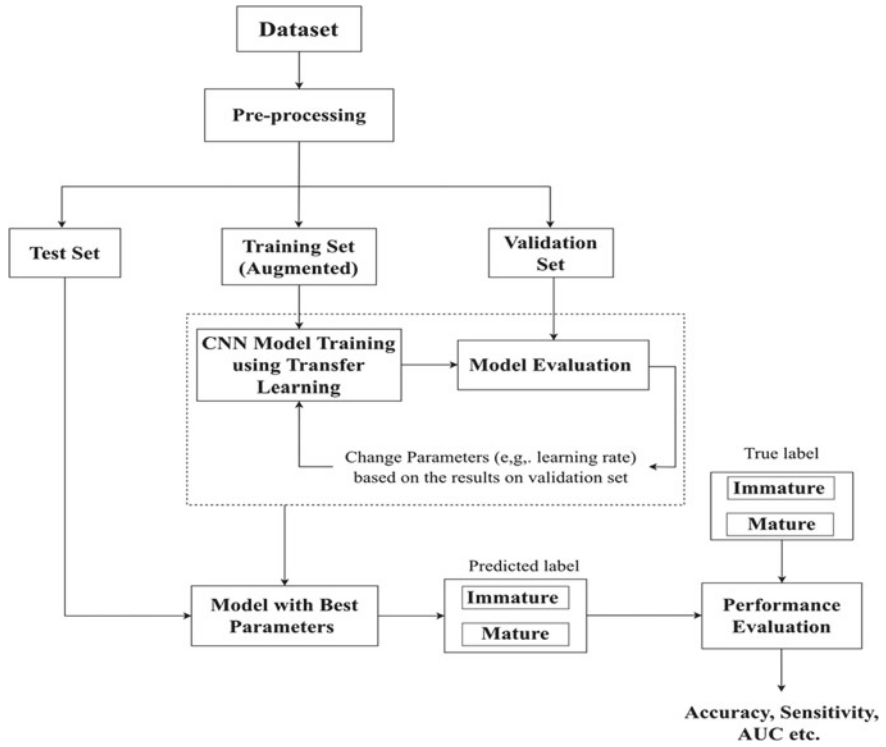


Fig. 2 Workflow of the proposed architecture

2.3 Pre-processing

A pre-processing step was conducted before the images were fed into the proposed structure. The first step was to split the data into three divisions: training, validation, and test set, each with its target labels. We used fixed splitting of the dataset to identify which model performs the best. The utilized dataset, for training, validation, and test, is presented in Table 1. Then, the original images were resized from  $400 \times 400 \times 1$  pixels to  $224 \times 224 \times 1$  pixels to reduce dimensionality, and computations, and assist the network to perform better in less time with simpler calculations. After that, we converted each image into a tensor. Finally, we applied horizontal flip and vertical flip to augment the images of the training set only.

Table 1 Data distribution

| Leukocyte type | Total image | Train  | Validation | Test |
|----------------|-------------|--------|------------|------|
| Immature       | 3532        | 2824   | 354        | 354  |
| Mature         | 14,833      | 11,865 | 1484       | 1484 |

2.4 Transfer Learning with CNN Models

Using transfer learning, the knowledge of a previously trained model can be utilized as the basis for a new task. In this study, several CNN architectures were used via transfer learning. The basic idea behind this was to take a pre-trained model that was trained from a huge amount of data, such as ImageNet, and then transfer the network to the binary classifier (immature and mature leukocyte for this study). This was accomplished by replacing several layers of the classifier part of the model with newly added layers. While retraining the model, only the newly added fully connected (FC) layers were trainable. The convolutional layers of the model were chosen as non-trainable (i.e., weights remain constant and do not update themselves). Since those layers of pre-trained CNN models are capable of providing rich and generic feature representations of the data provided, while the newly added FC layers (along with other layers as shown in Fig. 3) were only treated as a classification layer that was trained on the training set (see Table 1).

2.4.1 Pre-trained CNN Models

DenseNet161, ResNet50, AlexNet, and VGG-16 were used in this study. The layers of the classifier part for these models were removed with newly added FC layers, ReLU, dropout layers, batch normalization layers, and finally Log-Softmax layer for classification. Figure 3 illustrates the transfer learning approach in the modified AlexNet used in this study.

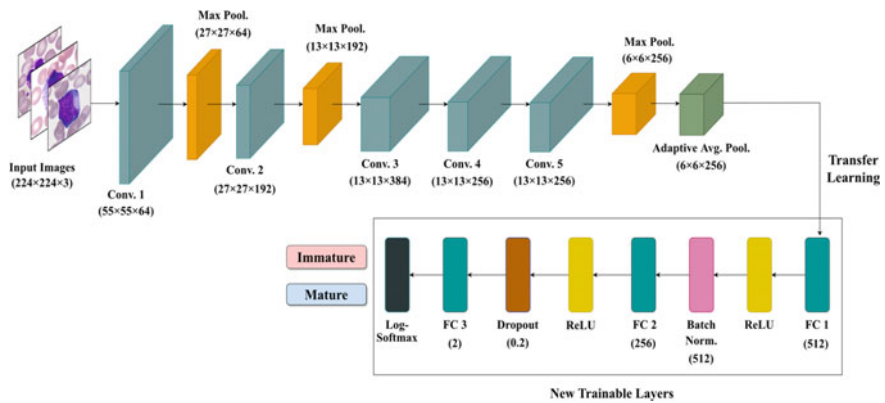


Fig. 3 Modified AlexNet architecture used in this study

## 2.5 Hyperparameters and Empirical Architectures

Different CNN architectures (e.g., ResNet50, DenseNet161, VGG-16, and AlexNet) were utilized by replacing some layers (e.g., FC layer, ReLU layer, dropout layer, batch normalization layer, and log softmax layer) and by changing the hyperparameters (e.g., learning rate, weight decay, batch size, epoch, etc.) for obtaining the optimized parameters for the respective CNN architecture. Table 2 represents a list of all the pre-trained CNN architectures and their respective hyperparameters that were tuned to obtain the best possible results for that respective CNN model.

## 2.6 Evaluation Criterion

The model's performance was quantitatively evaluated with accuracy, precision, recall (sensitivity), specificity, and F1-score as evaluation metrics. For this purpose, the model's classification report along with the confusion matrix of true positive (TP), false positive (FP), false negative (FN), and true negative (TN) was utilized. In addition, AUC was used to determine how well leukocyte classification is ranked rather than measuring absolute values.

## 3 Results

The findings of the test set along with model training and validation (see Table 1) for the proposed study are reported in this section. On the same dataset, state-of-the-art approaches were compared to the proposed approach at the end of this section. Bold texts in Tables 3, 4, 5, 6, 7 and 8 indicate the best results.

Among several CNN models, only AlexNet, DenseNet161, ResNet50, and VGG-16 performed better in comparison. Following that, these models were tweaked using different hyperparameters to observe whether there was any improvement in the performance. Among these, the modified AlexNet architecture outperformed other CNN architecture in terms of performance. Table 2 depicts the various parameters that were examined before reaching the final structure tuned to perform the best.

Table 3 shows the training and validation results of various pre-trained CNN models which were modified with the parameters listed in Table 2. In comparison to the other models, the modified AlexNet has produced the best outcomes in the shortest amount of time. It achieved 99.25% training and 97.38% validation accuracy with 0.02601 training loss and 0.09731 validation loss respectively after completing 35 epochs with a batch size of 64 for training and 32 for validation, as shown by the provided curves in Figs. 4 and 5.

Table 4 shows the evaluation of the findings obtained using the proposed approach on the test dataset. The highest accuracy, sensitivity, specificity, and AUC values



**Table 2** Parameters tested before reaching the best model

| CNN model   | Hyperparameter           | Value                        | Tuned value |
|-------------|--------------------------|------------------------------|-------------|
| ResNet50    | Number of FC layers      | 1, 2, 3                      | 3           |
|             | Number of dropout layers | 1, 2                         | 1           |
|             | Dropout rate             | 0.2, 0.3, 0.5                | 0.5         |
|             | Epoch                    | 20, 25, 30                   | 25          |
|             | Batch size               | 8, 32, 64, 128               | 64          |
|             | Optimizer                | SGD, Adam                    | Adam        |
|             | Learning rate            | 0.01, 0.001, 0.0001, 0.00001 | 0.0001      |
|             | Weight decay             | 0.01, 0.0001                 | 0.0001      |
| DenseNet161 | Number of FC layers      | 1, 2, 3                      | 3           |
|             | Number of ReLU           | 1, 2, 3                      | 2           |
|             | Number of dropout layers | 1, 2                         | 1           |
|             | Dropout rate             | 0.2, 0.3, 0.5                | 0.2         |
|             | Epoch                    | 20, 25, 30                   | 25          |
|             | Batch size               | 8, 32, 64                    | 64          |
|             | Optimizer                | SGD, Adam                    | Adam        |
|             | Learning rate            | 0.01, 0.001, 0.0001, 0.00001 | 0.0001      |
|             | Weight decay             | 0.01, 0.0001                 | 0.0001      |
| AlexNet     | Number of FC layers      | 1, 2, 3                      | 3           |
|             | Number of ReLU           | 1, 2, 3                      | 2           |
|             | Number of dropout layers | 1, 2                         | 1           |
|             | Dropout rate             | 0.2, 0.3, 0.5                | 0.2         |
|             | Epoch                    | 25, 35, 40                   | 35          |
|             | Batch size               | 8, 32, 64                    | 64          |
|             | Optimizer                | SGD, Adam                    | Adam        |
|             | Learning rate            | 0.001, 0.0001, 0.00001       | 0.00001     |
|             | Weight decay             | 0.0001, 0.00001              | 0.00001     |
| VGG-16      | Number of FC layers      | 1, 2, 3                      | 3           |
|             | Number of dropout layers | 1, 2                         | 2           |
|             | Dropout rate             | 0.2, 0.3, 0.5                | 0.2         |
|             | Epoch                    | 25, 35, 40                   | 35          |
|             | Batch size               | 8, 32, 64                    | 64          |
|             | Optimizer                | SGD, Adam                    | Adam        |
|             | Learning rate            | 0.001, 0.0001, 0.00001       | 0.0001      |

**Table 3** Experimental results of modified CNN models after model training and validation

| CNN model   | Epoch | Training Acc. (%) | Training loss  | Validation Acc. (%) | Validation loss |
|-------------|-------|-------------------|----------------|---------------------|-----------------|
| ResNet50    | 25    | 95.82             | 0.10806        | 94.75               | 0.14999         |
| DenseNet161 | 25    | 97.62             | 0.06247        | 96.24               | 0.13663         |
| AlexNet     | 35    | <b>99.25</b>      | <b>0.02601</b> | <b>97.38</b>        | <b>0.09731</b>  |
| VGG-16      | 35    | 90.87             | 0.27693        | 89.12               | 0.22369         |

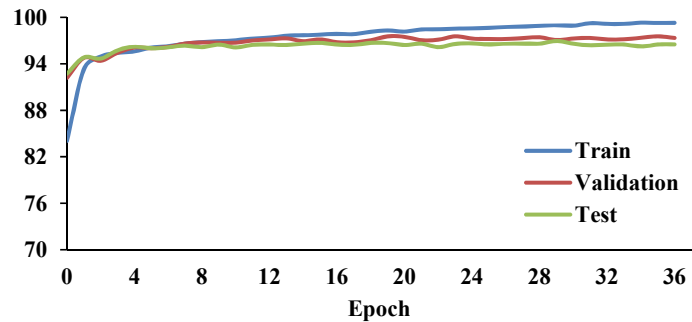


**Fig. 4** Training, validation, and test accuracy of modified AlexNet for the proposed approach

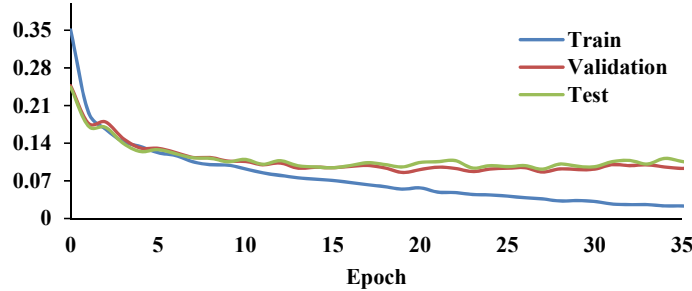


**Fig. 5** Training, validation, and test loss of modified AlexNet for the proposed approach

are written in bold. In Table 4, modified AlexNet achieved the highest accuracy, specificity, and AUC score with a value of 96.52%, 97.51%, and 94.94%, respectively, whereas ResNet50 achieved the highest 92.93% sensitivity in detecting immature leukocytes. Figure 6 represents the comparison of test results as mentioned in Table 4.

A more in-depth understanding of the model’s quantitative evaluation is shown in Table 5 where the weighted average is calculated using the weights of each class from the supported samples. Among all modified CNN models, modified AlexNet achieved the highest classification report. For modified AlexNet, the weighted average of

**Table 4** Test results of modified CNN models for the proposed approach

| CNN model   | Accuracy (%) | Sensitivity (%) | Specificity (%) | AUC (%)      |
|-------------|--------------|-----------------|-----------------|--------------|
| ResNet50    | 95.48        | <b>92.93</b>    | 96.09           | 94.51        |
| DenseNet161 | 95.43        | 90.67           | 96.56           | 93.62        |
| AlexNet     | <b>96.52</b> | 92.34           | <b>97.51</b>    | <b>94.94</b> |
| VGG-16      | 90.97        | 90.39           | 91.10           | 90.75        |

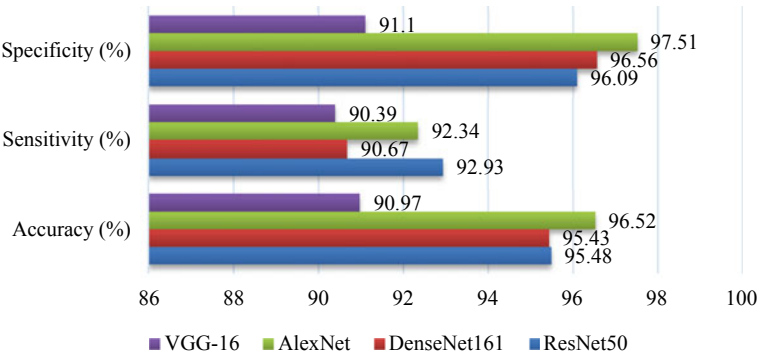


**Fig. 6** Comparison of the test results for the modified CNN models used in this study

recall, 97.0% implies that 3% of samples have immature leukocytes, which means the proposed method incorrectly fails to exclude them. And, 3% of samples are incorrectly classified among all identified true classes, according to the weighted average of precision of 97.0%. The F1 score with a weighted average of 97.0% indicates that modified AlexNet has improved precision and recall for identifying leukocyte type.

The confusion matrix of Fig. 7 illustrates the detailed class-wise investigation of correct and incorrect predictions for the modified AlexNet architecture. Among 354 immature samples, 27 samples are wrongly classified as mature. And, 37 mature samples are wrongly predicted as immature. Table 6 represents the comparison between the test results of AlexNet and the modified AlexNet architecture used in this study. The modified AlexNet performed better than the typical AlexNet architecture.

Performance of the modified AlexNet model was also investigated by changing the split ratio of the dataset: training (60, 70, 80) %, validation (20, 15, 10) % and testing (20, 15, 10) %. From the test results of the model presented in Table 7, it is observed that using a larger training set increases the model’s performance.

**Table 5** Classification report for test set using modified CNN models

| CNN model   | Classes          | Precision   | Recall      | F1-score    | Support |
|-------------|------------------|-------------|-------------|-------------|---------|
| ResNet50    | Mature           | 0.98        | 0.96        | 0.97        | 1484    |
|             | Immature         | 0.85        | 0.93        | 0.89        | 354     |
|             | Weighted average | 0.96        | 0.95        | 0.96        | 1838    |
| DenseNet161 | Mature           | 0.98        | 0.97        | 0.97        | 1484    |
|             | Immature         | 0.86        | 0.91        | 0.88        | 354     |
|             | Weighted average | 0.96        | 0.95        | 0.95        | 1838    |
| AlexNet     | Mature           | 0.98        | 0.98        | 0.98        | 1484    |
|             | Immature         | 0.90        | 0.92        | 0.91        | 354     |
|             | Weighted average | <b>0.97</b> | <b>0.97</b> | <b>0.97</b> | 1838    |
| VGG-16      | Mature           | 0.98        | 0.91        | 0.94        | 1484    |
|             | Immature         | 0.71        | 0.90        | 0.79        | 354     |
|             | Weighted average | 0.92        | 0.91        | 0.91        | 1838    |

**Fig. 7** Confusion matrix for modified AlexNet on the test dataset

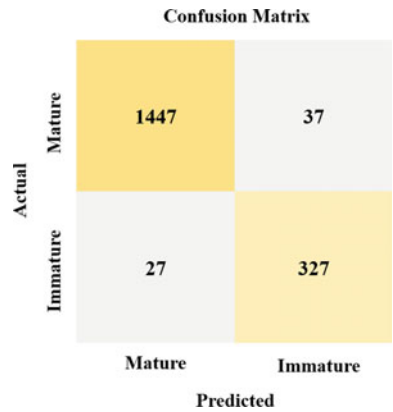


**Table 6** Test results of AlexNet and modified AlexNet for the proposed approach

| CNN model               | Accuracy (%) | Sensitivity (%) | Specificity (%) | AUC (%)      |
|-------------------------|--------------|-----------------|-----------------|--------------|
| AlexNet                 | 84.77        | 89.55           | 83.63           | 86.60        |
| <b>Modified AlexNet</b> | <b>96.52</b> | <b>92.34</b>    | <b>97.51</b>    | <b>94.94</b> |

**Table 7** Test results of modified AlexNet model while using three different sets of training, validation and testing

| Amount of test set (%) | Accuracy (%) | Sensitivity (%) | Specificity (%) | AUC (%)      |
|------------------------|--------------|-----------------|-----------------|--------------|
| 20                     | 96.23        | <b>94.34</b>    | 97.30           | 94.82        |
| 15                     | 96.44        | 91.90           | <b>97.89</b>    | 94.89        |
| 10                     | <b>96.52</b> | 92.34           | 97.51           | <b>94.94</b> |

## 4 Discussion

The most important finding of this study is that the inclusion of nonlinearity (i.e., RELU layer) and batch normalization layer in-between the fully connected layers in the case of a typical AlexNet architecture improved the performance for immature leukocytes detection and achieved state-of-the-art accuracy (96.52%), for binary classification of acute myeloid leukemia. Moreover, this study also presents a framework for modifying a pre-trained model for use in applying transfer learning to detect immature leukocytes for the diagnosis of AML.

The proposed modified AlexNet architecture surpasses the other modified CNN architectures (ResNet50, DenseNet161, and VGG-16) in leukocyte classification, with the highest training and validation accuracy of 99.25% and 97.38% respectively as shown in Table 3. Comparatively, the model produced lower training and validation losses than the others. The test set results of Table 4 indicate that the proposed modified AlexNet architecture achieved the highest accuracy, specificity, and AUC among the other models. From the test set results of Table 5, it can be observed that the modified AlexNet achieved the highest precision, recall, and  $F_1$ -score on a class-wide basis. Therefore, modified AlexNet performed the best in leukocyte classification among the other models used in this study.

The test set results of Table 6 indicate that the addition of new layers at the classifier part of a typical AlexNet architecture significantly improved the classification performance of the modified AlexNet architecture. This could be because of the inclusion of the batch normalization layer and ReLU nonlinearity in-between the FC layers. Instead of using softmax, in this study, log-softmax was utilized because of its better numerical performance and gradient optimization than softmax.

Table 8 shows the state-of-the-art comparison as mentioned in Sect. 1. The modified AlexNet achieved the best results concerning the accuracy of binary classification where random forest and SVM were implemented in [6, 8, 10]. Comparing the size of the data used in [6], this study utilized a much larger dataset and outperformed the method used in [6] by a 4.5% margin.

Since early diagnosis is critical for the effective treatment of AML patients, the proposed approach can speed up AML diagnosis by detecting immature leukocytes, which could save lives in impoverished nations where the diagnosis can take weeks. Though the proposed model cannot identify AML on its own, it can be utilized as a support tool to assist doctors in reducing the time and cost associated with AML diagnosis. This study may serve as a foundation to aid the future research on the computer-aided diagnosis of AML using deep neural networks.

**Table 8** State-of-the-art methods

| Authors          | Model                   | Classification type | Dataset used                                                       | Accuracy      |
|------------------|-------------------------|---------------------|--------------------------------------------------------------------|---------------|
| [6]              | Random forest           | Binary              | 600 Mature leukocytes, 674 Immature leukocytes                     | 92.99%        |
|                  |                         | Multi               | 78 Erythroblasts, 26 Monoblasts, 70 Promyelocytes, 500 Myeloblasts | 93.45%        |
| [7]              | ANN                     | Multi               | 873 AML cells                                                      | 93.57%        |
| [8]              | Random forest           | Binary              | 130 AML cells, 130 Normal cells                                    | 95.89%        |
| [9]              | Random forest           | Multi               | 50 AML cells                                                       | 90.00%        |
| [10]             | SVM                     | Binary              | 165 AML cells, 165 Normal cells                                    | 95.00%        |
|                  |                         | Multi               | 165 AML cells                                                      | 87.00%        |
| <b>This work</b> | <b>Modified AlexNet</b> | <b>Binary</b>       | <b>14,833 Mature leukocytes, 3532 Immature leukocytes</b>          | <b>96.52%</b> |

## 5 Conclusion

This study presents an approach of using pre-trained CNN architecture with a slight modification to detect immature leukocytes from morphological images for the diagnosis of acute myeloid leukemia. Using transfer learning approaches, the study modified and analyzed several pre-trained networks to improve the performance of leukocyte detection. The modified networks produced 90.97%-96.52% test accuracy on test samples for identifying immature leukocytes from mature ones. Lack of training samples in immature class, uneven data distribution and intra and inter-class similarity, information loss during image resizing, etc. contributed to a drop in performance of the proposed approach. Classifying immature leukocytes into different subtypes, reducing computational time, and evaluating the competency of the proposed approach with other AML datasets are the future works of this study.

## References

1. Hossain MS et al (2014) Diagnosed hematological malignancies in Bangladesh-a retrospective analysis of over 5000 cases from 10 specialized hospitals. *BMC Cancer* 14(1):1–7
2. Saultz JN, Garzon R (2016) Acute myeloid leukemia: a concise review. *J Clin Med* 5(3):33
3. American Society of Hematology. <https://www.hematology.org/>. Last accessed 05 Nov 2022
4. Kumar CC (2011) Genetic abnormalities and challenges in the treatment of acute myeloid leukemia. *Genes Cancer* 2(2):95–107

5. Ahmed N et al (2019) Identification of leukemia subtypes from microscopic images using convolutional neural network. *Diagnostics* 9(3):104
6. Dasariraju S, Huo M, McCalla S (2020) Detection and classification of immature leukocytes for diagnosis of acute myeloid leukemia using random forest algorithm. *Bioengineering* 7(4):120
7. Harjoko A et al (2018) Classification of acute myeloid leukemia subtypes M1, M2 and M3 using active contour without edge segmentation and momentum backpropagation artificial neural network. In: MATEC web of conferences. EDP Sciences, p 01041
8. Alagu S, Bagan KB (eds) (2021) Computer assisted classification framework for detection of acute myeloid leukemia in peripheral blood smear images. In: *Innovations in computational intelligence and computer vision*. Springer
9. Wiharto W, Suryani E, Putra YR (2019) Classification of blast cell type on acute myeloid leukemia (AML) based on image morphology of white blood cells. *Telkomnika* 17(2):645–652
10. Kazemi F, Najafabadi TA, Araabi BN (2016) Automatic recognition of acute myelogenous leukemia in blood microscopic images using k-means clustering and support vector machine. *J Med Signals Sens* 6(3):183
11. Matek C et al (2019) A single-cell morphological dataset of leukocytes from AML patients and non-malignant controls [Data set]. <https://doi.org/10.7937/tcia.2019.36f5o9ld>. Last accessed 05 Nov 2022
12. Clark K et al (2013) The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *J Digit Imaging* 26(6):1045–1057

# Chapter 36

## Automatic Bone Mineral Density Estimation from Digital X-ray Images



Abdullah Al Mahmud and Kalyan Kumar Halder

**Abstract** Bones are the most significant anatomical moving parts in humans. Osteoporosis is a bone disease that is a serious global public health problem. It is one of the most severe diseases that may be diagnosed sooner using medical imaging techniques. The bone mineral density (BMD) test is the gold standard for diagnosing osteoporosis. The goal of this research is to establish a computer-aided diagnostic system capable of detecting osteoporosis more effectively using digital X-ray images. The methodology in this paper consists of two stages: pre-processing and post-processing. In pre-processing, the digital X-ray images are rescaled and denoised to improve their quality. These images are then used to estimate BMD scores, which, in turns, are utilized to identify the patient's condition. A comparison with existing methods confirms that the proposed approach is fairly promising for detecting osteoporosis conditions.

**Keywords** Bone mineral density · Osteoporosis · Osteopenia · X-ray image · MATLAB

## 1 Introduction

A metric known as bone mineral density (BMD) measures the porous quality of bones. The loss of bone strength occurs in osteoporosis, a condition characterized by reduced bone mass and microarchitectural changes in bone tissue. Osteoporosis is a type of degenerative joint disease that predominantly affects the hip, knees, and

---

A. Al Mahmud (✉)

Institute of Information and Communication Technology, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [abmahmud11@gmail.com](mailto:abmahmud11@gmail.com)

K. K. Halder

Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [kalyan@eee.kuet.ac.bd](mailto:kalyan@eee.kuet.ac.bd)



spine. It raises the risk of fracture in both men and women [1]. The most frequent way of measuring bone strength is to use BMD to track bone mass decrease [2]. It is a crucial standard for the diagnosis of osteoporosis. It refers to the amount of minerals per square centimeter of bones. BMD is utilized in scientific medication as an oblique indicator of osteoporosis and fracture risk. Automated estimation of the BMD values of various parts of the body is an essential requirement of computer-aided osteoporosis diagnosis [3]. Although it may affect anybody, the chance of having osteoporosis increases with age [4].

The primary concept of [5] is to check and evaluate an image processing set of rules that calculates the BMD from digital X-ray (each real-time and open-source) photographs. This technique can lessen the expense of diagnosis as well as the time of tracking the affected person's status. Additionally, it is a non-invasive technique. The authors in [6] evaluated that osteoporosis is a circumstance that weakens the bones due to the decrease of the mineral calcium within the bones. It is referred to as the "lethal silence" due to the fact that the signs and symptoms of the disorder will pass ahead all of the time without caution before. The affected person will be recognized, while the bonehead is damaged already. Women are more likely than men to suffer from osteoporosis due to the fact that women have lesser bone mass than men [7].

The BMD is conventionally measured through a special technique called dual-energy X-ray absorptiometry (DEXA) or dual X-ray absorptiometry (DXA) scanning. But this scheme is more costly than X-ray. Therefore, it must be advantageous if someone could estimate the exact outcome using digital X-ray images evading the time taken by DEXA or DXA scanning and the higher expense of these approaches [5].

Zaia et al. [8] applied the fractal lacunarity analysis to develop a prototype function that precisely illustrates the change of pixel mass density in the bone image. Medical applications imply that the prototype might be effective in the earlier detection of osteoporosis. Akkus et al. [9] provided a model that is developed to predict osteoporosis risk factors and assessing osteoporosis influencing factors in postmenopausal Turkish women using a multiple binary logistic regression technique. An artificial neural network is another way that has been used to assess osteoporosis in [10]. The development of deep learning algorithms aids in the assessment of diseases more precisely than conventional techniques.

Furthermore, segmentation of the region of interest (ROI) is essential for proper health diagnosis. Thus, an automated method for segmenting bone regions and improving the accuracy of BMD calculation is required [11]. However, BMD changes in comparison throughout the race and ethnic groups. There is a large race and ethnic variations in BMD in African and Asian origins. Additionally, variations of body sizes substantially attenuate or reverse the BMD values among US Caucasian people and Asian people [12].

The main purpose of this paper is to measure the BMD values from digital X-ray images. The goal is to determine whether a patient is normal, osteoporotic, or osteopenic by using the BMD values calculated through an improved image processing algorithm as well as making comparisons with the BMD values received from the same patient's bone densitometer.

## 2 Proposed Method

The representation of the proposed method is shown through a block diagram in Fig. 1. At the initial stage, the input X-ray images of any size are collected from diagnostic centers or laboratories, and then, an image resizing operation is taken to resize the images to a fixed dimension. After that, the images are converted to grayscale images to reduce the computation demands. The pixel intensity values of the grayscale images range from 0 to 255.

The next significant step is ROI segmentation. In this process, a threshold value is determined from the grayscale X-ray image using the Otsu's method [13]. This threshold value is used to convert the image into a black and white image. The border pixels of the white regions are then identified, which are used later to crop the ROI region in a rectangular shape.

The ROI images may contain noise which affects the proper estimation of bone diameter and thickness, and it can give incorrect BMD values. For this reason, an anisotropic diffusion filter is employed to obtain a noise-free image. The anisotropic diffusion filter [14] has caught the eyes of many people since its invention by Perona and Malik in 1990. The filter applies the law of diffusion on the pixel values of an image to smooth the textures in it. Anisotropic filter generates two patterns to the grayscale values; the dark regions became more dark, and the light regions became more white, while the gray regions between black and white are diminished. The use of a threshold function ensures that no diffusion occurs across edges. Therefore, the filter preserves the edges of the image as well as highlights the important ones [15].

In the post-processing, an intensity profile graph of the filtered image is plotted using MATLAB "improfile" function. The function gives a plot of pixel intensities versus distance along a horizontal line segment on the bone. Using this plot, the thickness ( $t$ ) and the outer diameter ( $d$ ) of the bone are determined. The volume per area (VPA) is then calculated by Khan et al. [5]

$$\text{VPA} = t \cdot t \left( 1 - \frac{t}{d} \right) \quad (1)$$

The VPA is multiplied by specific density ( $\rho$ ) to get the BMD value as

$$\text{BMD} = \text{VPA} \cdot \rho \quad (2)$$

By comparing the estimated BMD value of a patient with a standard range of BMD values, it is demonstrated whether the particular patient has osteoporosis or not.

## 3 Simulation Experiments

The proposed method is implemented in MATLAB to measure the BMD values of different patients and to diagnose whether they are suffering from osteoporosis.

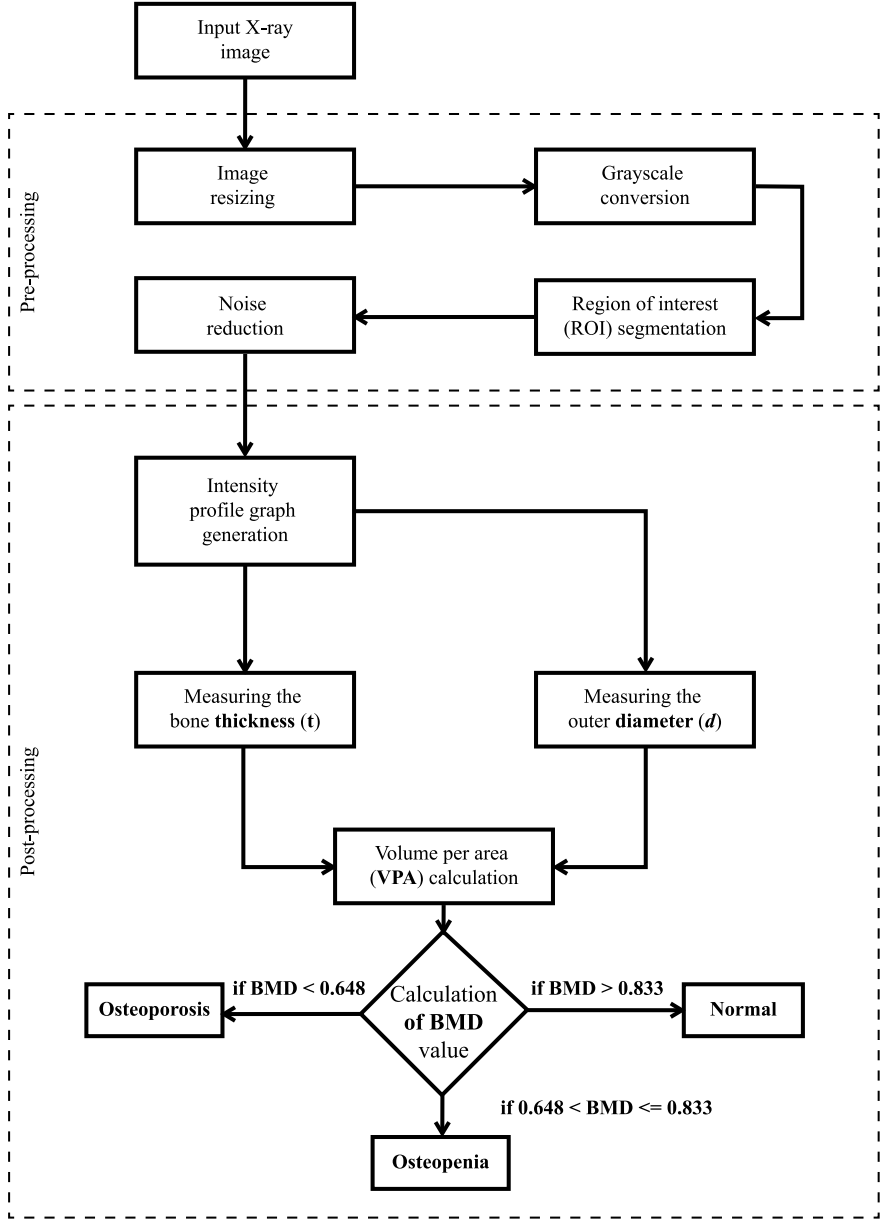
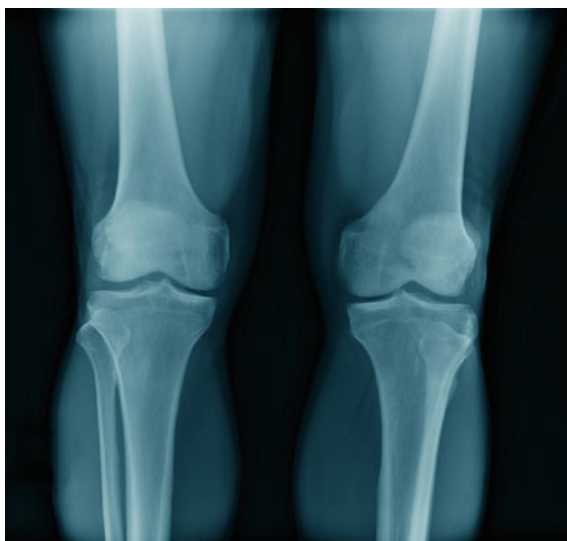


Fig. 1 Block diagram of the proposed method

**Fig. 2** Digital version of an X-ray image of a patient



A variety of X-ray images of femur is collected from [16] which contains images of both normal persons and osteoporosis patients. A sample of the dataset is shown in Fig. 2, which is a medical X-ray image without applying any filtering or pre-processing schemes.

### 3.1 Pre-processing Results

An input digital X-ray image is then resized and converted into a grayscale image. Grayscale images are widely applied in identifying characteristics of any image rather than processing directly on color images since they simplify the technique and also minimize computational complexity. Figure 3a, b show the resized image and the grayscale image, respectively.

ROI segmentation refers to a technique of determining the region of interest in an input frame. Once the grayscale image is obtained, it is taken for the next phase to find the ROI of that X-ray image. In ROI segmentation, only the bone regions are kept, and all other regions are discarded. The rectangle-shaped ROI is shown in Fig. 3c.

The ROI image might have noise; thus, the noise must be removed to obtain the best outcome from BMD estimation. To perform such an operation, an anisotropic filter is employed on the ROI image. The filtered image is shown in Fig. 3d, which is sharper and clearer than that of before filtering.

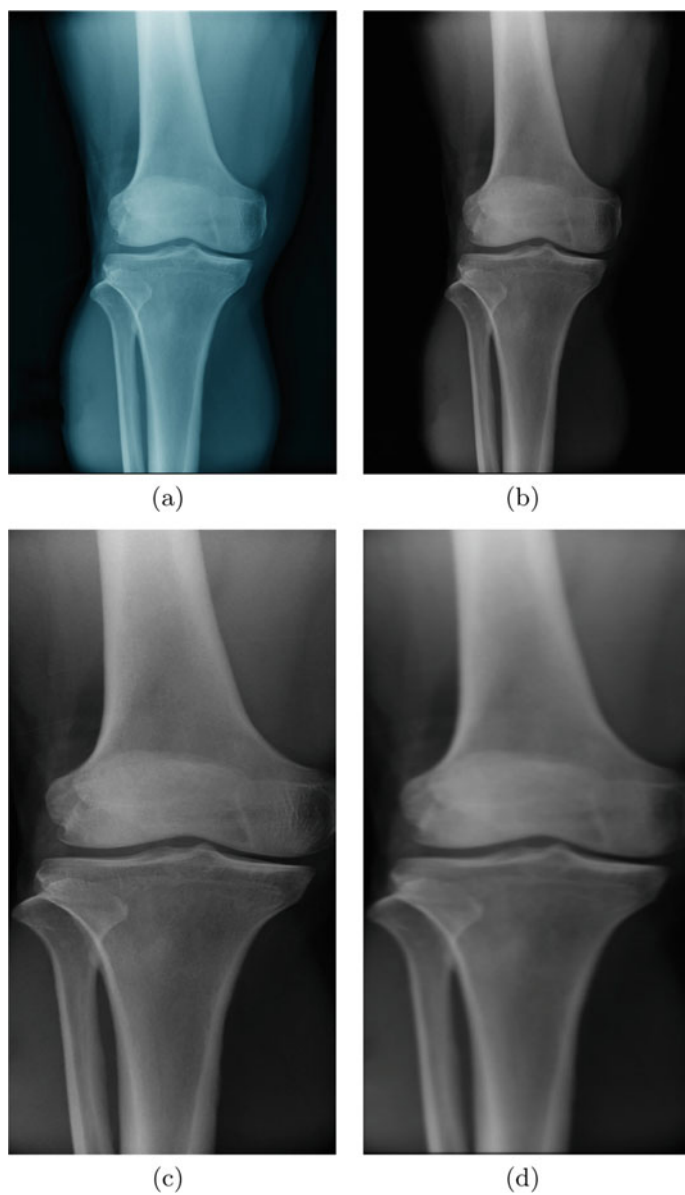


**Fig. 3** Results of pre-processing stage: **a** resized image, **b** grayscale image, **c** ROI image, and **d** filtered image

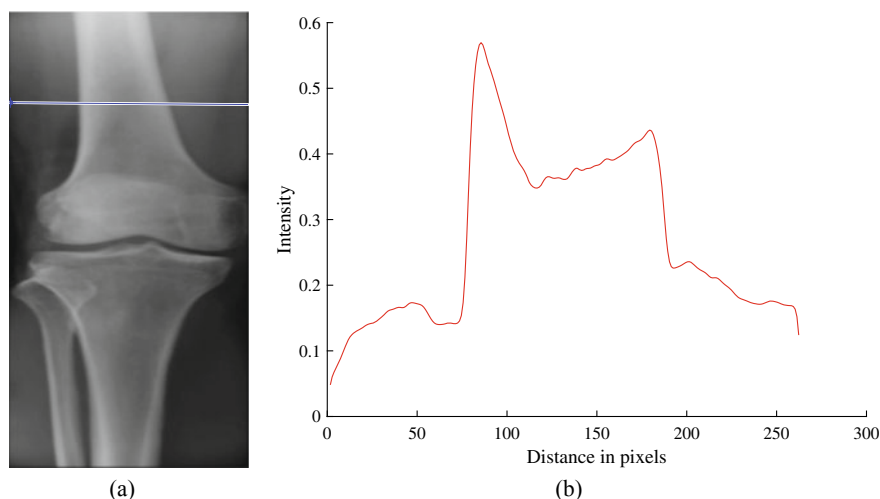


**Fig. 4** Results of post-processing stage: **a** line segment over an image and **b** plot of intensity along the line segment

### 3.2 Post-processing Results

The intensity profile, a line graph that typically shows the pixel values along the image's line segment, is generated using a MATLAB function called “improfile”. The line could be specified automatically or manually with a mouse on the ROI image. Figure 4a, b show a typical line segment on the ROI image and the corresponding intensity profile curve, respectively. As can be seen from the intensity profile, there is a sharp change in intensities where the bone region starts and finishes. The outer diameter and the thickness of the bone are determined by calculating the distance along the horizontal axis, which, in turns, are used to calculate the BMD values.

As the dataset contains the X-ray images of femurs, the specific density ( $\rho$ ) of this region is considered as  $0.35 \pm 0.19 \text{ g/cm}^3$  [5]. Once the BMD score is less than 0.648, it is considered as osteoporosis, if the BMD score belongs from 0.648 to 0.833 that is considered osteopenia, and if the score is above 0.833 that is considered as normal [17].

Table 1 shows a clear comparison of the performance of the proposed method with respect to the benchmark result of bone densitometer [16] and also that of the method in [5]. The test results on 10 different X-images are presented here. Out of 10 images, the method in [5] detects 7 patients' conditions accurately compared to the benchmark data, whereas this proposed method detects a higher of 9 patients' conditions accurately. For the 8th image, both methods were unable to detect osteoporosis; rather they estimated it as an osteopenia, which is still an indicator of weakening of bones. Therefore, the proposed method can be an alternative to the conventional DEXA scanning technique.

**Table 1** Comparison of results

| Image S. No. | Benchmark result [16] | Results of [5] |                     | Proposed method |                     |
|--------------|-----------------------|----------------|---------------------|-----------------|---------------------|
|              |                       | BMD value      | Patient's condition | BMD value       | Patient's condition |
| 1            | Normal                | 0.9593         | Normal              | 1.4151          | Normal              |
| 2            | Normal                | 1.1671         | Normal              | 1.6523          | Normal              |
| 3            | Normal                | 0.7195         | Osteopenia          | 1.0953          | Normal              |
| 4            | Osteoporosis          | 0.5833         | Osteoporosis        | 0.5825          | Osteoporosis        |
| 5            | Osteoporosis          | 0.1114         | Osteoporosis        | 0.4269          | Osteoporosis        |
| 6            | Normal                | 0.9723         | Normal              | 1.3964          | Normal              |
| 7            | Osteoporosis          | 0.3013         | Osteoporosis        | 0.5786          | Osteoporosis        |
| 8            | Osteoporosis          | 0.7338         | Osteopenia          | 0.7399          | Osteopenia          |
| 9            | Normal                | 0.5725         | Osteoporosis        | 1.1167          | Normal              |
| 10           | Osteoporosis          | 0.5746         | Osteoporosis        | 0.4311          | Osteoporosis        |

## 4 Conclusion

In this study, an automated scheme has been proposed to estimate the BMD values in the femur regions by the use of digital X-ray images. This method has employed a few simple but efficient tools to get the BMD values, which makes it less time-consuming than the compared methods. As confirmed by the experimental results, this approach is expected to be a useful tool for predicting osteoporosis at the early stage.

Although the proposed work uses only femur images, it can be applied to any other regions of the body, provided that the specific density would be changed. The next step of this research is to build a graphical user interface (GUI) to make it easier to measure the BMD results. If all the necessary operations could be done through the GUI directly, the diagnosis will be quicker and more user-friendly.

## References

1. Donnell PM, McHugh PE, Mahoney DO (2007) Vertebral osteoporosis and trabecular bone quality. *Ann Biomed Eng* 35:170–189
2. Marshall D, Johnell O, Wedel H (1996) Meta-analysis of how well measures of bone mineral density predict occurrence of osteoporotic fractures. *The BMJ* 312(7041):1254–1259
3. Zhou X, Hayashi T, Chen H, Hara T, Yokoyama R, Kanematsu M, Hoshi H, Fujita H (2009) Automated measurement of bone-mineral-density (BMD) values of vertebral bones based on X-ray torso CT images. In: 2009 annual international conference of the IEEE engineering in Medicine and Biology Society. IEEE, USA, pp 3573–3576
4. Sigurdsson G, Aspelund T, Chang M, Jonsdottir B, Sigurdsson S, Eiriksdottir G, Gudmundsson A, Harris TB, Gudnason V, Lang TF (2006) Increasing sex difference in bone strength in old

- age: the age, gene/environment susceptibility-Reykjavik study (AGES-REYKJAVIK). *Bone* 39:644–651
5. Khan SS, Jayan AS, Nageswaran S (2017) An image processing algorithm to estimate bone mineral density using digital X-ray images. In: 2017 second international conference on electrical, computer and communication technologies (ICECCT). IEEE, India, pp 1–4
  6. Promworn Y, Pintavirooj C (2012) Development of bone mineral density and bone mineral content measurements system using a dual energy X-ray. In: The 5th 2012 biomedical engineering international conference. IEEE, Thailand, pp 1–4
  7. Vishnu T, Saranya K, Arunkumar R, Devi MG (2015) Efficient and early detection of osteoporosis using trabecular region. In: 2015 online international conference on green engineering and technologies (IC-GET). IEEE, India, pp 1–5
  8. Zaia A, Eleonori R, Maponi P, Rossi R, Murri R (2006) MR imaging and osteoporosis: fractal lacunarity analysis of trabecular bone. *IEEE Trans Inf Technol Biomed* 10(3):484–489
  9. Akkus Z, Camdeviren H, Celik F, Gur A, Nas K (2005) Determination of osteoporosis risk factors using a multiple logistic regression model in postmenopausal Turkish women. *Saudi Med J* 26(9):1351–1359
  10. Lemineur G, Harba R, Kilic N, Ucan ON, Osman O, Benhamou L (2007) Efficient estimation of osteoporosis using artificial neural networks. In: 33rd annual conference of the IEEE Industrial Electronics Society. IEEE, Taiwan, pp 3039–3044
  11. Fathima SMN, Tamilselvi R, Beham MP (2019) Estimation of t-score and BMD values from X-ray images for detection of osteoporosis. In: Proceedings of the 3rd international conference on cryptography, security and privacy. ACM, USA, pp 220–224
  12. Nam H-S, Shin M-H, Zmuda JM, Leung PC, Barrett-Connor E, Orwoll ES, Cauley JA (2010) Race/ethnic differences in bone mineral densities in older men. *Osteoporos Int* 21:2115–2123
  13. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66
  14. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
  15. Hongbo HR, Yu H, Lan Y (2013) A novel image denoising algorithm based on anisotropic diffusion equation. In: 2013 5th international conference on intelligent human-machine systems and cybernetics. ACM, USA, pp 410–413
  16. Shutterstock, <https://shutterstock.com>. Last accessed 5 June 2022
  17. Ott SM (2002) Osteoporosis and bone physiology. University of Washington, USA



# Chapter 37

## Machine Learning Algorithms for the Prediction of Prostate Cancer



M. M. Imran Molla, Julakha Jahan Jui, Humayan Kabir Rana,  
and Nitun Kumar Podder

**Abstract** Machine learning (ML) algorithms employ a wide variety of methods such as statistical, probabilistic and optimization that allow machines to learn and predict based on historical data. For these capabilities, ML is substantially used in different cancer prediction. The exact prediction is essential for the planning of cancer treatment. This work aims at exploring the idea of using ML algorithms to predict prostate cancer. To increase the probability of survival of prostate cancer patients, it is important to establish suitable prediction models. Therefore, in this study, we applied several ML techniques including support vector machine, k-nearest neighbors, Naive Bayes, random forest and logistic regression algorithms to predict prostate cancer. Among all the five ML techniques, the logistic regression provided better prediction result with 86.21% accuracy. Hence, our achieved results indicate that the logistic regression technique could be utilized for prostate cancer prediction.

**Keywords** Prostate cancer · Support vector machine · K-nearest neighbors · Naive Bayes · Random forest · Logistic regression

---

M. M. I. Molla

Department of Computer Science and Engineering, Pabna University of Science and Technology, Pabna, Bangladesh

J. J. Jui

Department of Computer Science and Engineering, Northern University of Business and Technology, Khulna, Bangladesh

H. K. Rana (✉)

Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh

e-mail: [humayan.pustcse@gmail.com](mailto:humayan.pustcse@gmail.com)

N. . K. Podder

Bangladesh Institute of Governance and Management (BIGM), Dhaka, Bangladesh

## 1 Introduction

Prostate cancer is well known and most common types of cancer that emerges in a gland called prostate in the male reproductive system [1]. Cancer develops when mutation of the prostate cells being out of control and starts to multiply until a tumor develops. The prostate has the function of generating fluid that protects and nourishes semen cells. The seminal vesicles behind the prostate gland contain the majority of the semen fluid. Cancer cells can expand from the prostate to other cells in the body, specifically in the lymph nodes and bones [1]. The decision of treatments for prostate cancer depends on the patient's conditions and cancer stage. There are a variety of treatments for this type of cancer, such as radiation therapy, chemotherapy and radiation and surgical combination.

Prostate cancer is really a critical health concern and needs to be handled carefully. Several studies on cancer detection have been performed using of ML algorithms. Authors in [2] employed logistic regression, decision tree and artificial neural network (ANN) to assess the survival of prostate cancer. Survival rates can be predicted using various predictive techniques such as statistical regression model and machine learning models. Sran Jović et al. proposed three machine learning approaches such as ELM, ANN and GP for the prediction of prostate cancer. They hypothesized that these algorithms can be used to predict prostate cancer relevantly.

A deep learning technique was reported by H. Wen et al. for predicting prostate cancer. They used surveillance, epidemiology and end result program for their estimation to classify mortality rates into two groups: just below and over 60 months. Their results demonstrate that the ANN has the highest accuracy in survivability prediction of prostate cancer, and the rate of accuracy is 85.64%. authors in [3] used convolutional neural networks (CNNs) for the recurrence prediction of prostate cancer. In their proposed method, they trained 80 control-case pairs as independent set and gave the AUC value of 0.81 for a sample of 30 recurrent cases and 30 non-recurring controls but did not demonstrate the classification accuracy in their analysis.

Evaluation of machine learning algorithms for diagnosing prostate cancer and Gleason grading was reported in [4]. They used their own dataset for analysis, with 33 samples of normal prostate tissue and 161 prostate cancer samples. The highest classification accuracy (77.84%) they have been achieved using logistic regression which seems quite low. Extreme logistic regression was used to estimate mortality from prostate cancer. In their measurements, they took online dataset and compared the results with respect to the weighted-accuracy metric but did not report any classification accuracy in the study. In [5], early diagnosis of prostate cancer was reported through the use of artificial neural networks and SVM. Their proposed system was designed with a kernel-based polynomial feature and delivers 79% accuracy. The achieved results were equivalent to SVM, but the training speed was much higher. In [6], random forest regression was used to estimate prostate cancer. They used datasets from online in their research to build an RF model based on age, prostate-specific antigen and ultrasound that predicts prostate cancer with 83.10% accuracy. It is stated in that prostate cancer is predicted using quantitative phase imaging (QPI) [7]. They

used picture data comprising 280 cases of benignity and 141 cases of malignancy. A classifier of SVM was used and gained 0.23 error rate with 0.83 AUC value. Authors in [8] reported that prostate cancer is predicted to recur with chemical imaging. They used SVM classifier and obtained a 0.75 AUC score, which is quite poor.

This paper examined several ML techniques for the classification of prostate cancer. The ML techniques are support vector machine (SVM), k-nearest neighbors (KNN), Naive Bayes, random forest (RF) and logistic regression (LR). The performance accuracy of the mentioned classifiers is evaluated based on the confusion matrix, precision, recall and *F1* score. The ROC curve was also employed to evaluate the performance of these classifiers as because it is another significant performance measurement metric for classification algorithms. The evaluation metrics suggest LR as the best model among all five ML algorithms for the prediction of prostate cancer.

2 Methodology

There are some general steps for building a prostate cancer predictive model. The flow diagram of these steps is shown in Fig. 1. Data collection and preprocessing of datasets for checking and adjusting the missing values are the first work. The correlation analysis also performed on the data and finally the data is being normalized. The dataset is divided into two groups for analysis (training and testing). In the next step, several machine learning classifiers are applied to training and testing datasets to evaluate the performance of them. In the last stage, the best machine learning model is recommended based on the performance.

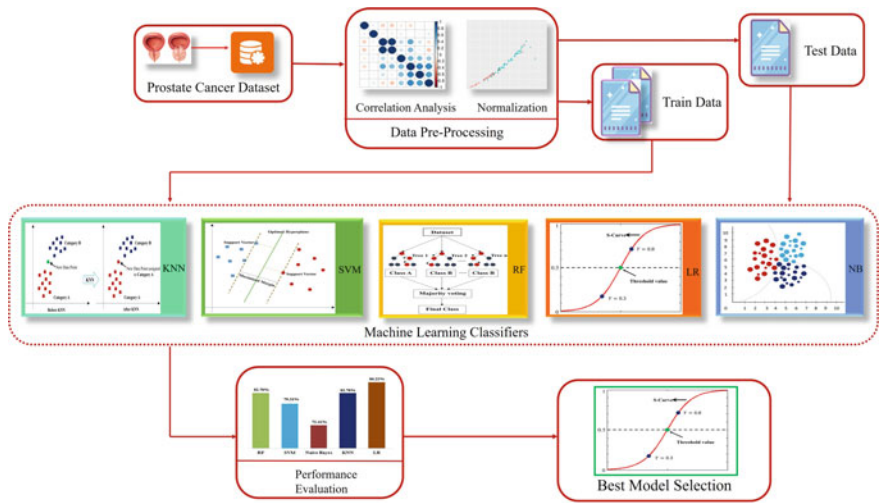


Fig. 1 Flow diagram of the analytical approach used in this study

**Table 1** Attributes and its types in dataset

| Serial No. | Attributes        | Types   |
|------------|-------------------|---------|
| 1          | Id                | Numeric |
| 2          | Diagnosis         | Text    |
| 3          | Radius            | Numeric |
| 4          | Texture           | Numeric |
| 5          | Perimeter         | Numeric |
| 6          | Area              | Numeric |
| 7          | Smoothness        | Numeric |
| 8          | Compactness       | Numeric |
| 9          | Symmetry          | Numeric |
| 10         | Fractal dimension | Numeric |

## 2.1 Datasets Descriptions

The dataset for prostate cancer has been gathered from open source data repository website (Kaggle) [9]. There are 100 observations with 10 variables in this dataset. The attributes and their types are given in Table 1.

Among the 10 variables, 9 variables are the independent and the variable diagnosis is a dependent variable. Diagnosis has two classes to be specific Benign (B) and Malignant (M). 62 observations from 100 are belong to class M and 38 belong to class B.

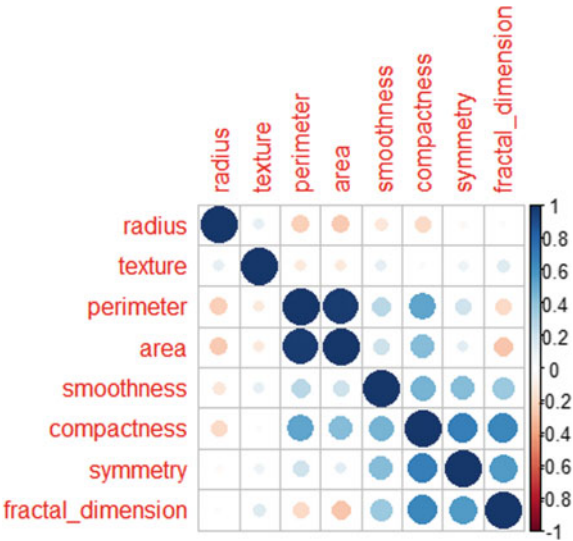
## 2.2 Data Preprocessing

In this phase, various analyses on data have been performed to build a robust prediction model. The analysis of data is described as follows.

**Missing value checking** We will get biased results if we avoid the missing data, so the missing data is very important. Generally, missing information can diminish the power of the model which can increase the rate of miss classification. Data attributes are checked carefully whether there are any missing or NA values in the dataset. Finally, no missing value is found in the dataset.

**Correlation analysis** The datasets are examined to find out the correlation of the attributes. A highly correlated features or highly correlated coefficients have a high impact on performance metrics in the classification method. High level of negative correlation produces a low level of performance. Figure 2 demonstrates the correlation of each attribute for the dataset. Correlations are displayed in blue and red colors where dark blue color represents positive correlation and light red color represents

**Fig. 2** Correlation plot among the attributes of the data



a negative correlation. Correlation coefficients are proportional to the intensity of color and size of the circles. As expected, most of the attributes are significantly correlated with each other. Moreover, the above figure depicts a small amount of negative correlation of attributes and that can be negligible.

**Feature scaling** The prostate cancer dataset consists of attributes which are measured on different scales. This difference ranging may be an obstacle for the superior prediction model. Few ML algorithms such as KNN can be used for feature scaling that is important to unify the features of all attributes. Therefore, the attributes are scaled between 0 and 1 to make it uniform. The scaling plot of two attributes (area and perimeter) is shown in Fig. 3, where Fig. 3a shows the original values of the attributes and Fig. 3b demonstrates the scaled values of the attributes.

2.3 Prediction Techniques

Various supervised ML techniques such as the RF, SVM, Naïve Bayes, KNN and LR have been used for predicting the prostate cancer. These techniques are briefly discussed in this section.

**Support Vector Machine (SVM)** SVM is a set of supervised learning methods that is used for regression, classification and outliers detection. For cancerous cells detection from millions of images or to predict future driving routes with well fitted regression model, we can use SVM [10, 11]. SVM is also known as maximum margin classifier

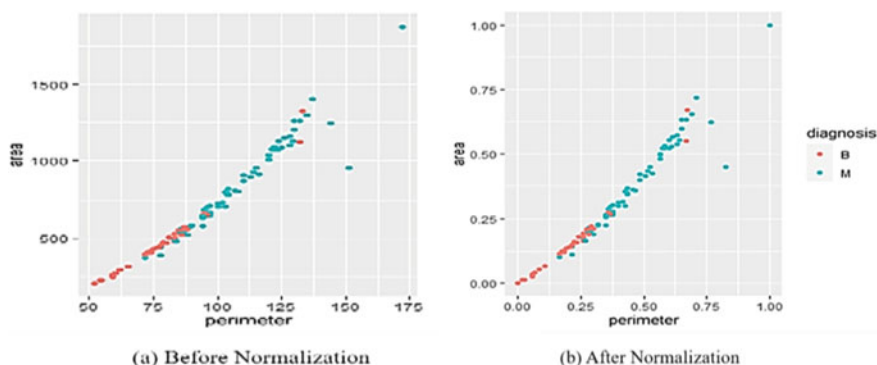


**Fig. 3** Area and perimeter scaling: **a** before scaling and **b** after scaling

because it maximized the distance from nearest data points of all the classes by choosing the decision boundary. This allows a modest number of misclassifications toward the margin to be tolerated. Here,

- Support Vectors—Closest data point to the hyperplane. With the support of these data points, the isolating line can be characterized.
- Optimal Hyperplane—Correctly classify all data points.
- Maximized Margin—If there is a minimal distance among the hyperplane and training point, then it known by margin and it can be maximized by the SVM algorithm.

**K-Nearest Neighbors (KNN)** Another ML classification approach is KNN. The purpose is to assign a new data point to one of the numerous existing classes. As a result, certain neighbors  $k$  are chosen, and the  $k$  closest data points are determined using either Manhattan or Euclidean distances. For a new data point, it is counted that how many neighbors belongs to category A and category B, then new data point is assigned to a category based on majority voting.

**Naïve Bayes algorithm** It is Bayes theorem-based classification technique with a group of classification algorithms where all those algorithms share a common principle. A Naive Bayes classifier presumes that the existence of a particular attribute in a class is not dependent on other features. Naive Bayes model is convenient for the vast amount of data. In addition to simplicity, it is considered as the best even extremely sophisticated methods of classification [12]. Bayes theorem presents a way of calculating posterior probability  $N(a|x)$  from  $N(a)$ ,  $N(x)$  and  $N(x|a)$ . The mathematical formulation of the Bayes theorem is described as follows:

$$N(a|x) = \frac{N(x|a)N(a)}{N(x)} \quad (1)$$

$$N(a|X) = N(x_1|a) \times N(x_2|a) \times \dots \times N(x_n|a) \times N(a)$$

where

- $N(ax)$  represents the posterior probability of class ( $a$ , target)
- given predictor ( $x$ , attributes).
- $N(a)$  represents the prior probability of class.
- $N(x|a)$  is the likelihood of predictor given in class.
- $N(x)$  is the prior probability of predictor.

**Random forest algorithm** Another common approach for classification and regression is random forest (RF) [13]. The RF algorithm creates the basic decision trees for every data set indicated by  $D(P, Q)$  using the equation below.

$$(P, Q) = (p_1, q_1), (p_2, q_2), \dots, (p_n, q_n) \quad (2)$$

where, the training observations number is  $n$  with a set of instances whose class membership is known as  $(p_i, q_i) \in (P, Q)$ .

**Logistic regression (LR)** LR is another classification algorithm, which is particularly used to assign observations to a distinct set of classes. It is derived from the probability theorem and kind of predictive algorithm. The LR hypothesis has a propensity to limit the cost function. The function which maps any real value into a range between 0 and 1 is known as sigmoid function [2]. Sigmoid is used to map predictions to probabilities. The equation of the logistic regression is as follows:

$$\log \left[ \frac{y}{1-y} \right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n \quad (3)$$

**Performance evaluation** Performance evaluation is an important part of a classification technique. Performance metrics assist to identify the best classification model. The performance of a classification technique can be measured based on confusion matrix, accuracy, precision, and recall/sensitivity and  $F1$  values.

**Confusion matrix** A confusion matrix is a structural layout that visualizes and summarizes prediction outcomes on the classification problems. In the confusion matrix, the event row is assigned as “positive” and the no-event row as “negative”. Then the event column of predictions is assigned as “true” and the no-event as “false” and the representation of the confusion matrix is given in Table 2. Here, True Positive (TP) means the predicted outcome is true and the actual outcome is also true. False Positive (FP) means the predicted outcome is true but the actual one is false. When the predicted outcome is false but actual result is true, this scenario is known as False Negative (FN). If the predicted outcome is false and the actual outcome is also false, this is called True Negative (TN).

**Performance evaluation matrix** The performance of the several methods is evaluated based on the classification accuracy, precision, recall/sensitivity,  $F1$  score and specificity. The ratio of correctly predicted class to the total class is called accuracy and Table 3 represents various performance evaluation formulas. The precision

**Table 2** Confusion matrix

| Actual class        | Predicted class     |                     |
|---------------------|---------------------|---------------------|
|                     | Positive prediction | Negative prediction |
| Positive class      | TP                  | FN                  |
| Negative prediction | FP                  | TN                  |

**Table 3** Performance measuring metrics

| Performance metrics | Formula                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------|
| Accuracy            | $\frac{TP+TN}{TP+TN+FP+FN}$                                                               |
| Precision           | $\frac{TP}{TP+FP}$                                                                        |
| Recall/ Sensitivity | $\frac{TP}{TP+FN}$                                                                        |
| <i>F</i> 1-score    | $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ |
| Specificity         | $\frac{TN}{TN+FP}$                                                                        |

represents the ratio of the correctly identified positive outcomes to all positive outcomes. Recall, is also called the sensitivity, is the measure of the ability to identify all actual positive classes as the true positive classes correctly. *F*1-score is calculated by applying the weighted average over precision and recall. If we have an uneven class appropriation, then *F*1-score is generally considered as more important than precision. Specificity is measured as the number of true negative predictions (TN) divided by the total number of negatives (*N*).

### 3 Result and Discussion

In this study, we performed different analyses to examine the five ML classifier for the classification of prostate cancer dataset. We splinted the dataset into two parts, 70% is used for train and 30% for validating the model. Tenfold cross validation is performed on training data when each machine learning classifier is trained up. The confusion matrix which is an important measure for classification is shown in Fig. 4 for RF, SVM, Naïve Bayes, KNN and LR classifiers. Out of 29 observations, 24 are correctly identified by RF and KNN classifiers, 21 are accurately recognized by Naïve Bayes classifier, SVM classifies 23 observations correctly and LR performs excellent among these algorithms with correct identification of 25 observations. The performance of these algorithms has been assessed by different evaluation procedures like accuracy, specificity, precision, sensitivity and *F*1 score. The accuracy of each classifier is shown in Fig. 5. Among the five classifiers, LR achieves the highest accuracy. Other measures such as precision, recall and *F*1 score are given in Table 4. The highest precision, recall and *F*1 score are highlighted in bold.



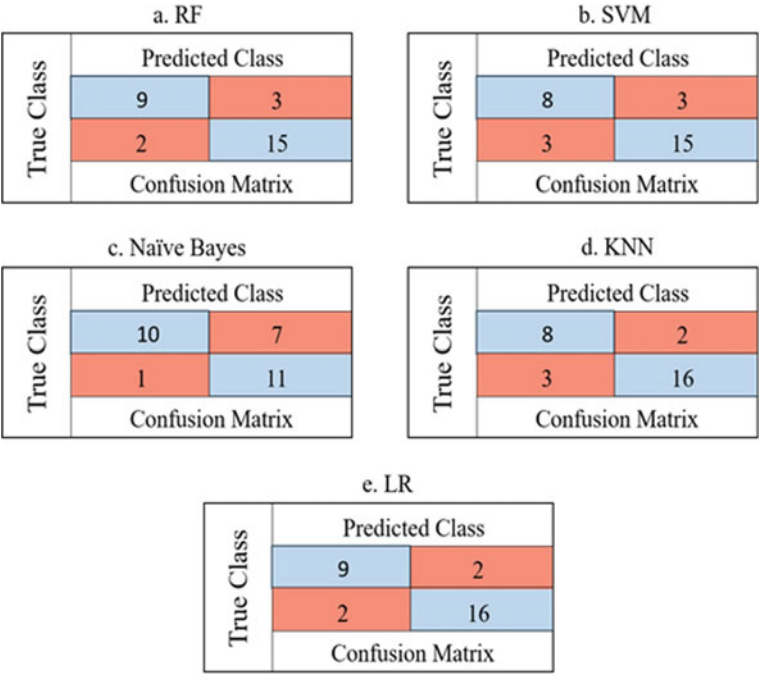


**Fig. 4** Confusion matrix of the classifiers, **a** RF, **b** SVM, **c** Naïve Bayes, **d** KNN and **e** LR

**Fig. 5** Bar chart of the accuracy of all methods

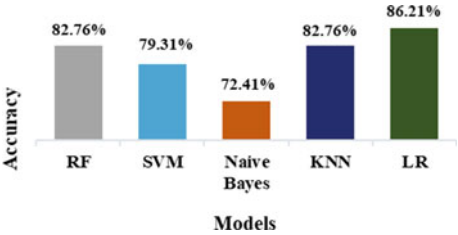


**Table 4** Precision, recall and *F1* score of different classifiers

| Classifier method | Precision    | Recall       | <i>F1</i> score |
|-------------------|--------------|--------------|-----------------|
| RF                | 0.818        | 0.75         | 0.782           |
| SVM               | 0.727        | 0.727        | 0.727           |
| Naive Bayes       | <b>0.909</b> | 0.588        | 0.714           |
| KNN               | 0.727        | 0.80         | 0.761           |
| LR                | 0.818        | <b>0.818</b> | <b>0.818</b>    |

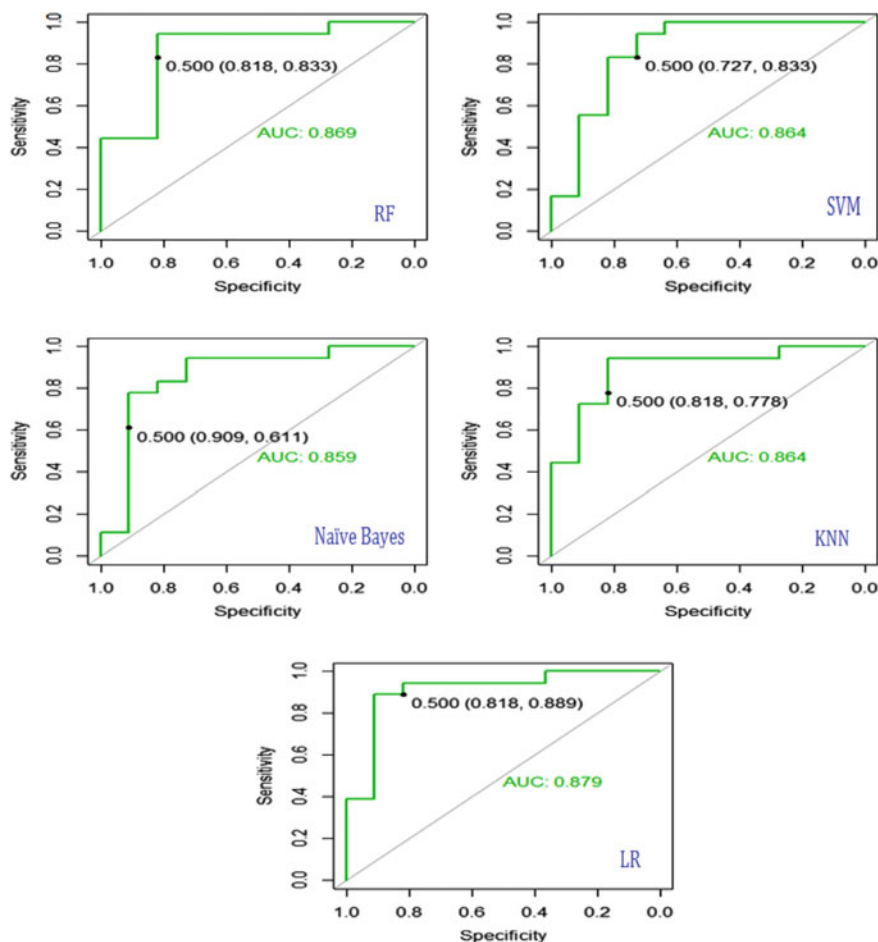


**Fig. 6** ROC curve of RF, SVM, Naïve Bayes, KNN and LR

High precision rate implies the ratio of highly correctly predicted positive observations to the total predicted positive observations. High precision correlated with the low false positive rate. Naïve Bayes demonstrates the highest precision rate. RF and LR both show the precision of 0.818 which is also pretty good. Recall or sensitivity is another measure for the classification algorithm. LR attains the highest recall score of 0.818 and recall of .80 is obtained by KNN classifier which is also pretty good. The  $F1$  score is calculated from the weighted average of precision and recall. Therefore,  $F1$  score considers both FP and FN. A high  $F1$  score is also an indication of a superior model. Receiver operating characteristic (ROC) curve is another important performance measuring metrics for classification problems at various thresholds. ROC is a one types of probability curve, and area under the curve (AUC) is represented by the degree or measure

of separability of classes. Figure 6 depicts the ROC curve in terms of sensitivity and specificity for RF, SVM, Naïve Bayes, KNN and LR classifier with the threshold value of 0.5. In the ROC curve, sensitivity and specificity belong to 0.818 and 0.889 for LR. Moreover, the value of AUC is 0.879 for LR which is the highest value among all classifiers, it means there is 87.9% probability that the model will be able to discriminate between positive and negative class.

From the experimental analysis in Fig. 6, it is clear that LR is the best classification method among RF, SVM, Naïve Bayes and KNN based on different classifier metrics for the prediction of prostate cancer.

## 4 Conclusion

The principal part of this work is to identify an effective machine learning classifier for prostate cancer prediction among five distinctive supervised machine learning classifiers including RF, SVM, Naïve Bayes, KNN and LR. In this study, the dataset was preprocessed and splinted into two parts; 70% of the data is used for training and the rest of the 30% for testing the model. Tenfold cross validation is also performed on training data. The performance of the classifier is performed using different criteria such as accuracy, precision, recall, *F1* score, ROC and AUC value. We have examined the performance of the above-mentioned classifiers on patient's information parameters and got the LR classifier gives the highest accuracy of 86.21%. Beside that RL provides the most elevated *F1* score and high AUC is about 0.8687. Finally, it can be concluded that LR performs better than other machine learning classifiers on prostate cancer prediction.

**Acknowledgement** This research work has been financially supported in part by Green University of Bangladesh.

## References

1. Zupan B, Demšar J, Kattan MW, Beck JR, Bratko I (2000) Machine learning for survival analysis: a case study on recurrence of prostate cancer. *Artif Intell Med* 20(1):59–75
2. Imran Molla M, Jui JJ, Bari BS, Rashid M, Hasan MJ (2021) Cardiotocogram data classification using random forest based machine learning algorithm. In: *Proceedings of the 11th national technical seminar on unmanned system technology* 2019. Springer, pp 357–369
3. Kumar N, Verma R, Arora A, Kumar A, Gupta S, Sethi A, Gann PH (2017) Convolutional neural networks for prostate cancer recurrence prediction. In: *Medical imaging 2017: digital pathology*, vol 10140. International Society for Optics and Photonics, p 101400H
4. Alexandratou E, Atlamazoglou V, Thireou T, Agrogiannis G, Togas D, Kavantzias N, Patsouris E, Yova D (2010) Evaluation of machine learning techniques for prostate cancer diagnosis and Gleason grading. *Int J Comput Intell Bioinf Syst Biol* 1(3):297–315
5. Çınar M, Engin M, Engin EZ, Ateşçi YZ (2009) Early prostate cancer diagnosis by using artificial neural networks and support vector machines. *Expert Syst Appl* 36(3):6357–6361

6. Xiao L-H, Chen P-R, Gou Z-P, Li Y-Z, Li M, Xiang L-C, Feng P (2017) Prostate cancer prediction using the random forest algorithm that takes into account transrectal ultrasound findings, age, and serum levels of prostate-specific antigen. *Asian J Androl* 19(5):586
7. Nguyen TH, Sridharan S, Macias V, Kajdacsy-Balla A, Melamed J, Do MN, Popescu G (2017) Automatic Gleason grading of prostate cancer using quantitative phase imaging and machine learning. *J Biomed Opt* 22(3):036015
8. Kwak JT, Kajdacsy-Balla A, Macias V, Walsh M, Sinha S, Bhargava R (2015) Improving prediction of prostate cancer recurrence using chemical imaging. *Sci Rep* 5(1):1–10
9. Kaggle. <https://www.kaggle.com>. Accessed 15 Jan 2022
10. Rana HK, Azam MS, Akhtar MR, Quinn JM, Moni MA (2019) A fast iris recognition system through optimum feature extraction. *PeerJ Comput Sci* 5:e184
11. Jony MH, Johora FT, Khatun P, Rana HK (2019) Detection of lung cancer from CT scan images using GLCM and SVM. In: 2019 1st International conference on advances in science, engineering and robotics technology (ICASERT). IEEE, pp 1–6
12. Chen S, Webb GI, Liu L, Ma X (2020) A novel selective naïve bayes algorithm. *Knowl-Based Syst* 192:105361
13. Jui JJ, Imran Molla M, Bari BS, Rashid M, Hasan MJ (2020) Flat price prediction using linear and random forest regression based on machine learning techniques. In: *Embracing industry 4.0*. Springer, pp 205–217

# Chapter 38

## Detection of Ventricular Fibrillation from ECG Signal Using Hybrid Scalogram-Based Convolutional Neural Network



Md. Faisal Mina, Amit Dutta Roy, and Md. Bashir Uddin

**Abstract** Ventricular fibrillation (VF) is a shockable ventricular cardiac arrhythmia that causes sudden cardiac death. Numerous studies around the world either utilized hand-engineered features or automatic feature extraction techniques together with different classification algorithms to detect VF from electrocardiogram (ECG) signals. This study introduces a novel hybrid scalogram-based convolutional neural network (CNN) that deploys empirical mode decomposition (EMD) on ECG signals by taking the advantage of the intrinsic mode functions (IMFs). ECG recordings that contained VF and Non-VF episodes were collected from the CU Ventricular Tachyarrhythmia Database. After preprocessing, each ECG recording was segmented at definite intervals (2, 3, and 4 s) and each segment was named either ‘VF’ or ‘Non-VF’ episode, based on the information given in the database. Using EMD, the IMFs of these segments were obtained. IMFs were then converted into hybrid scalogram images using continuous wavelet transform. A CNN model was trained by these hybrid scalogram images for detecting Non-VF and VF segments. To classify VF and Non-VF episodes, this study applied Visual Geometry Group (VGG)-19 that is a novel object-recognition system consisting of 19 layers of deep CNN. The accuracies of VGG-19 for the detection of VF using the segment interval of 2, 3, and 4 s were 95.91, 96.21, and 97.98%, respectively. This method can be employed to identify and categorize VF and Non-VF events from ECG signals where the feature extraction dilemma was mitigated.

**Keywords** Empirical mode decomposition • Continuous wavelet transform • Convolutional neural network

---

Md. F. Mina • A. D. Roy • Md. B. Uddin (✉)

Department of Biomedical Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

e-mail: [mbu@bme.kuet.ac.bd](mailto:mbu@bme.kuet.ac.bd)

Md. F. Mina

e-mail: [mina1615017@stud.kuet.ac.bd](mailto:mina1615017@stud.kuet.ac.bd)

A. D. Roy

e-mail: [amit@bme.kuet.ac.bd](mailto:amit@bme.kuet.ac.bd)

## 1 Introduction

Ventricular fibrillation (VF) is a disorder in the rhythm of the heart which leads to the tremble of the heart's ventricles. Cardiac arrest is caused by VF, which is accompanied by the loss of consciousness and a lack of pulse. Death is the frequent result due to the lack of treatment. Approximately 10% of those who have a cardiac arrest have VF initially. Those who are out of the hospital when the VF is discovered have a survival rate of around 17%, while those who are in the hospital have a survival rate of about 46% [1].

It is necessary to detect and treat VF in the early stage to reduce the probability of death. VF is a difficult task to detect due to the complex and irregular patterns in ECG. By detecting VF events the appropriate kind of treatment can be identified. A common way to detect VF is by observing the ECG signal which is done manually by the physicians. This manual detection process is time-consuming and slow, and also clinical results could be overlapped due to a large number of ongoing examinations. As a result, a fully computerized method is required to detect VF.

Previous attempts have been made for developing the detection systems for VF. Several feature extraction and classification algorithms for the detection and classification of VF have been presented in recent decades. A technique was proposed in [2] for the classification of VF by using a time–frequency Pseudo Wigner-Ville (PWV) operation where denoising and signal alignment was performed, and then the validity was checked by four different classifiers: logistic regression with L2 regularization (L2 RLR), adaptive neural network classifier, support vector machine (SVM), and bagging classifier (BAGG). Assessment of these classifiers for VF detection (including flutter episodes) showed a sensitivity of 95.56% and a specificity of 98.8%. Another method [3] was proposed for the detection of VF episodes where the digital Taylor-Fourier transform was used to divide the ECG signal into distinct oscillatory patterns (DTFT). The least-square support vector machine (LS-SVM) classifier with linear and radial basis function (RBF) kernels was used for the classification of VF. The accuracy, sensitivity, and specificity for the categorization of Non-VF and VF episodes were 89.81%, 86.38%, and 93.97%, respectively. Another study [4] extracted nine-time domain features and seven nonlinear features and classified these features using SVM employing different kernels. The classification achieved an accuracy of 94.7% and a sensitivity of 100% for the detection of VF. A study [5] combined SVM, adaptive boosting (AdaBoost), and differential evolution (DE) algorithms. In the evaluation phase, the proposed methodology showed an accuracy of 98.20%, sensitivity of 98.25%, and specificity of 98.18% using 5 s of the ECG segments. In another study [6], VF/ Ventricular Tachycardia (VT) classification scheme was implemented using a window length of 5 s where twenty-four features were retrieved using EMD and variable mode decomposition (VMD) techniques. The deep neural network (DNN) classifier obtained the highest accuracy of 99.2%, and sensitivity and specificity respectively of 98.8%, and 99.3%. Hand-engineered features or automatic feature extraction techniques were the drawbacks in these studies.

To overcome these drawbacks a technique [7] was proposed for the automatic separation of shockable and non-shockable ventricular arrhythmias from 2 s electrocardiogram (ECG) segments where ventricular fibrillation has been included as shockable ventricular arrhythmias. An eleven-layer CNN model was used to analyze segmented ECGs. The proposed method was cross-validated ten times and scored 93.18% accuracy, 95.32% sensitivity, and 91.04% specificity. A method [8] proposed used short-time Fourier transform (STFT) and continuous wavelet transform (CWT) with the CNN model which obtained a recall of 99% and an accuracy of 97%. Though the accuracies of these approaches were high, minimizing noise with maintaining feature connectivity to the underlying ECG signals was not maintained.

This study introduces a novel hybrid scalogram-based convolutional neural network (HSCNN) that deploys EMD on ECG signals by taking advantage of the IMFs which minimizes noise while maintaining feature connectivity to underlying ECGs.

## 2 Methodology

### 2.1 Data Collection

The data used for this study was acquired from the Creighton University Ventricular Tachyarrhythmia Database (CUDB) [9] which contains 35 eight-minute ECG recordings from human participants. Among these ECG recordings, the number of ventricular fibrillation episodes was 47 which were of different intervals and the rest of the episodes were normal rhythm, ventricular tachycardia, and ventricular flutter. All signals were digitized at 250 Hz having 12-bit resolution across a 10 V range after passing into an active second-order Bessel low-pass filter with a cutoff frequency of 70 Hz. There are approximately 127,232 samples within every record (significantly less than 8.5 min). ECG signal in the mentioned dataset contains 250 samples each second.

### 2.2 Proposed Approach

The proposed approach for detecting ventricular fibrillation (VF) is illustrated in Fig. 1. At first, the raw ECG signals were collected from a public database and preprocessed for removing noise. After preprocessing, the signals were segmented for a definite interval of time. Then, these segments were converted into IMFs utilizing EMD. The IMFs were transformed into hybrid scalograms using CWT. Considering these hybrid scalograms as images, they were firstly resized into a  $224 \times 224$  matrix i.e., the size of the input layer of the CNN model. Then, the CNN model was

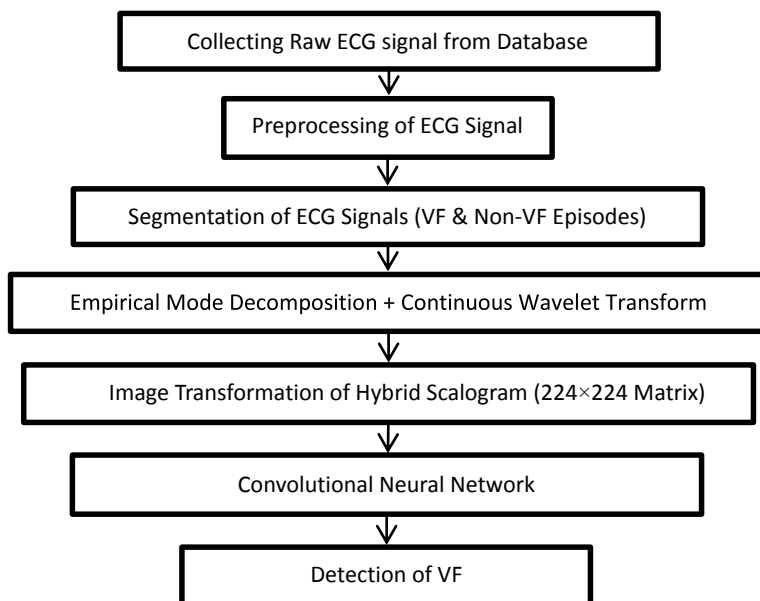


**Fig. 1** Illustration of the proposed methodology

trained using these images. By following this step-by-step procedure, ultimately, the classification of VF and Non-VF episodes were performed in this study.

**Preprocessing and Segmentation** ECGs with or without VF (Fig. 2) were collected from the database according to the given annotation. ECG recording was preprocessed by detrending and normalizing. The preprocessed ECG was segmented at definite intervals (2, 3, and 4 s), and depending on the info in the database, each segment was categorized as ‘VF’ or ‘Non-VF’ episode.

An ‘Equiripple FIR’ low pass filter was used which is the most efficient among the different FIR filters. An efficient approach was developed for the algebraic assessment of the filter’s impulse response based on the differential equation. The passband frequency, stopband attenuation, and passband ripple of the designed filter were 40 Hz, 60 Hz, and 1 dB respectively, whereas the sampling rate of the ECG signal was 250 Hz. The graphical representation of the Equiripple FIR low pass filter is illustrated in Fig. 3. The function ‘detrend’ was used to remove baseline wandering. The filtered data were normalized to avoid unwanted incompatibility during analysis.

At first, the raw ECG signals were collected from a public database and preprocessed for removing noise. After preprocessing, the signals were segmented for definite interval of time. Then, these segments were converted into IMFs utilizing EMD. The IMFs were transformed to hybrid scalograms using CWT. Considering these hybrid scalograms as images, they were firstly resized into a  $224 \times 224$  matrix i.e., the size of the input layer of CNN model. Then, the CNN model was trained using



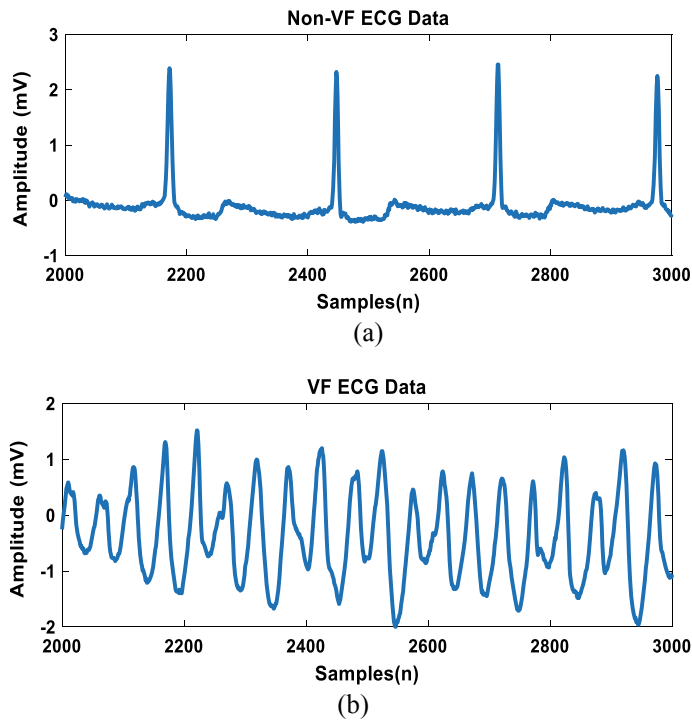


**Fig. 2** ECG with **a** Non-VF and **b** VF episodes

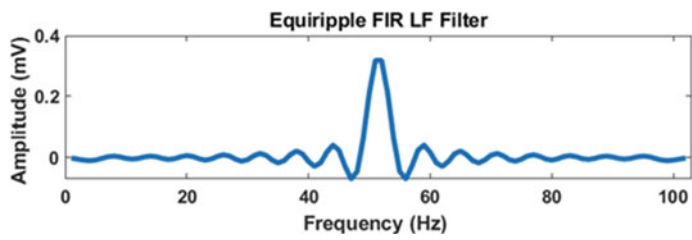


**Fig. 3** Graphical representation of the Equiripple FIR low pass filter

these images. By following this step-by-step procedure, ultimately, the classification of VF and Non-VF episodes were performed in this study.

Each ECG recording comprises a mix of normal ECG signals as well as VF and Non-VF occurrences. Following preprocessing, the ECG signals were segmented into definite time frames. To investigate the influence of the length of ECG frame (i.e., epochs or segments) on the performance of the model, frame lengths of 2 s ( $2 \times 250$  or 500 samples), 3 s ( $3 \times 250$  or 750 samples), and 4 s ( $4 \times 250$  or 1000 samples) were used in this study.

**Empirical Mode Decomposition (EMD)** EMD is especially well suited for studying and analyzing non-stationary natural data within the time domain, such as ECGs. In this study, EMD is pertained to generating intrinsic mode functions (IMFs) such as  $IMF_1(t)$ ,  $IMF_2(t)$ , ..., etc.  $IMF_N(t)$  that depends on the period ( $t$ ) of epochs ( $\varphi$ ) or the segmentation of the ECG data. An IMF is a symmetrical oscillatory function with a comparable amount of zero-crossings and extrema [10]. Epochs can be reconstructed with the sum of all IMFs and a constant known as  $r$  is shown in Eq. (1):

$$\varphi(t) = \sum_{i=1}^N IMF_i(t) + r(t) \quad (1)$$

From the segmented ECG signal  $x(t)$ , the upper envelope and lower envelope are constructed by joining the maxima  $x_u(t)$  and minima  $x_l(t)$ , respectively, of the raw signal. Then the mean of the extrema ( $M(t)$ ) was obtained and afterward, this mean was subtracted from the raw signal to extract the IMF( $I(t)$ ) as represented in Eqs. (2, 3). If the residue was monotonic then the IMF was taken for the further computational process. In this study, only the first IMF was converted into scalograms.

$$M(t) = \frac{x_u(t) + x_l(t)}{2} \quad (2)$$

$$I(t) = x(t) - M(t) \quad (3)$$

**Continuous Wavelet Transform (CWT) and Image Transformation** The CWT of  $X_\varphi^\Psi(s_c, \tau)$  is defined in Eq. (4):

$$X_\varphi^\Psi(s_c, \tau) = \langle \varphi, \psi(s_c, \tau) \rangle = \frac{1}{\sqrt{s_c}} \int_{-\infty}^{\infty} \varphi(t) \cdot \psi \left( \frac{t - \tau}{s_c} \right) dt \quad (4)$$

where  $\varphi$  is the signal segmentation epoch,  $\psi(s_c, \tau)$  is the mother wavelet,  $\tau$  is the translator which shifts along period  $t$ , and scale  $s_c$ . This compresses the wavelet according to the resolution [11]. In this study, Morse, Morlet, and Bump wavelets were considered as the mother wavelets for analysis. The CWT operation was performed on definite interval EMD signal ( $s$ ) by the convolution of the mother wavelet with scale,  $s_c$  by shifting along the period of the signal,  $t$ . The output of the convolution of mother wavelet with EMD signal is termed as wavelet co-efficient ( $x_i, y_f$ ) which is stored as a matrix of 2-Dimensional size as shown in Fig. 4. Then using ‘jet’ as the colormap, RGB images ( $339 \times 1001 \times 3$ ) were obtained from the co-efficient of the matrix [12]. The images were rescaled later to  $224 \times 224 \times 3$ . These images were then used as input in the CNN model.

**Convolutional Neural Network (CNN) and Detection of VF** CNN is a complex algorithm used in deep learning that usually takes images as input and extracts

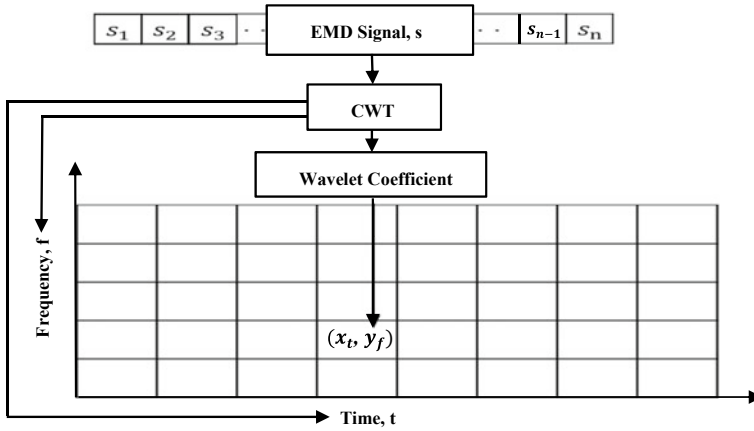


**Fig. 4** The process of construction of image using hybrid scalogram

features by convolving them using filters or kernels. Suppose, an  $N \times N$  image is convolved with an  $f \times f$  filter. By this convolution operation, the same feature is learned on the entire image. In each operation, the window moves, and the feature maps learn the new features. VGGNet is a CNN architecture designed by a group named Visual Geometry Group at Oxford [13]. It has 19 layers and was initially intended for use in an object-recognition system. The dimension of the input layer is  $224 \times 224 \times 3$ . In the architecture, the rectified linear unit (ReLU) function is utilized as the activation function in the convolution layers to improve network learning and prevent dissipating gradients. 40% dropout was used to prevent data from overfitting. Gradient descent with momentum was utilized as the optimal algorithm, which resulted in comparably faster network convergence. To match the number of categorization outputs, the neurons in the hidden layer on the final fully connected layer were increased. SoftMax activation was used on the classification layer, and cross-entropy was used as the cost function. Using binary classification to detect Non-VF and VF events, the segments (2, 3, and 4 s) have been divided randomly into 75% training dataset, 20% validation dataset, and 5% testing dataset.

### 2.3 Evaluation Metrics

The specificity, sensitivity, and accuracy of the learning scheme were used to assess its performance by obtaining true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Accuracy, sensitivity, and specificity were utilized for assessing the performance of the model as shown in Eqs. (5–7).

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (5)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (7)$$

### 3 Results

The typical hybrid scalograms from ECG signals for VF and Non-VF are represented in Figs. 5 and 6, respectively. Samples of ECG data of VF and Non-VF are shown in Figs. 5a and 6a, respectively. Constructed and rescaled hybrid scalograms with dimensions of  $224 \times 224 \times 3$  are shown respectively using Morlet as the mother wavelet in Figs. 5b and 6b, using Bump as the mother wavelet in Figs. 5c and 6c and using Morse as the mother wavelet in Figs. 5d and 6d.

The model was trained using 75% of the whole dataset where the highest training accuracy for each segment using different mother wavelets is approximately 95–98%. Among 20% of the whole dataset, the validation accuracy was approximately around 94–98%. Using binary classification, Non-VF and VF events were detected for the testing dataset for each segment (2, 3, and 4 s) which contains 5% of the whole dataset. The evaluation acquired from the proposed method is represented in Tables 1, 2, and 3 for the testing dataset.

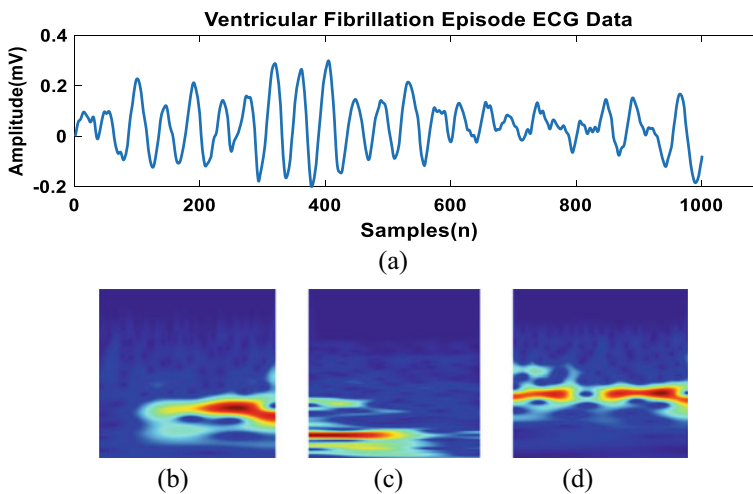


**Fig. 5** Typical scalogram of ECG with VF: **a** ECG signal sample (4 s), **b** constructed scalogram image using Morlet wavelet  $224 \times 224 \times 3$ , **c** constructed scalogram image using Bump wavelet  $224 \times 224 \times 3$ , and **d** constructed scalogram image using Morse wavelet  $224 \times 224 \times 3$

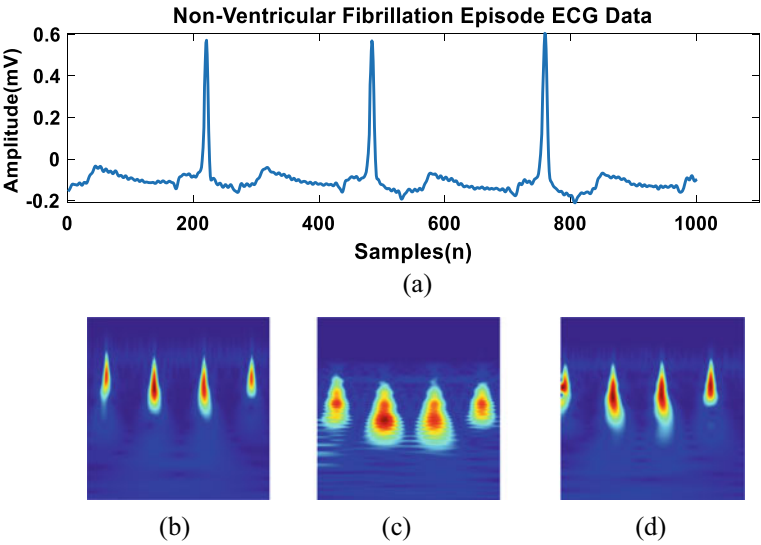


**Fig. 6** Typical scalogram of ECG with Non-VF: **a** sample of ECG signal (4 s), **b** constructed scalogram image using Morlet wavelet  $224 \times 224 \times 3$ , **c** constructed scalogram image using Bump wavelet  $224 \times 224 \times 3$ , and **d** constructed scalogram image using Morse wavelet  $224 \times 224 \times 3$

**Table 1** Evaluation metrics of different segment interval results for Morlet wavelet

| Epoch length (s) | Accuracy (%) | Specificity (%) | Sensitivity (%) |
|------------------|--------------|-----------------|-----------------|
| 2                | 95.91        | 96.86           | 92.95           |
| 3                | 96.21        | 96.85           | 94.82           |
| 4                | 97.98        | 99.00           | 95.91           |

**Table 2** Evaluation metrics of different segment interval results for Bump wavelet

| Epoch length (s) | Accuracy (%) | Specificity (%) | Sensitivity (%) |
|------------------|--------------|-----------------|-----------------|
| 2                | 92.89        | 95.65           | 90.14           |
| 3                | 93.17        | 93.90           | 92.45           |
| 4                | 95.55        | 93.33           | 97.78           |

**Table 3** Evaluation metrics of different segment interval results for Morse wavelet

| Epoch length (s) | Accuracy (%) | Specificity (%) | Sensitivity (%) |
|------------------|--------------|-----------------|-----------------|
| 2                | 95.30        | 81.25           | 76.49           |
| 3                | 95.63        | 98.82           | 92.45           |
| 4                | 95.41        | 93.33           | 97.50           |

In Table 1, by using the Morlet wavelet as the mother wavelet, the accuracies obtained at definite intervals 2, 3, and 4 s are, respectively, 95.91%, 96.21%, and 97.98%. The accuracies using the Bump wavelet were 92.89%, 93.17%, and 95.55% whereas, 95.30%, 95.63%, and 95.41% were the results using the Morse wavelet at definite intervals 2, 3, and 4 s, respectively. In addition, the highest accuracies resulted in the segment with an interval of 4 s.

## 4 Discussion

In this study, a hybrid scalogram-based CNN model was introduced by utilizing IMFs from EMD on ECG signals. This work used VGG-19 to identify VF and Non-VF episodes from the hybrid scalogram images converted from the IMFs. The best accuracy resulted in using the Morlet wavelet, which is 95.91%, 96.21%, and 97.98% respectively with the segment interval of 2, 3, and 4 s.

The most notable fact in this study was detecting VF from ECG data using hybrid scalograms and transfer learning with VGG-19, which is a practically effective and also a unique mechanism for identification. For constructing a hybrid scalogram, the ECG signal was decomposed into empirical mode. The decomposed signals were further converted into scalogram (CWT) images of dimension  $224 \times 224 \times 3$ . Since the EMD signal was converted into a scalogram from an ECG signal, this scalogram conversion technique is termed a hybrid scalogram.

To obtain the lowest segment classification accuracy, specificity and sensitivity for VF and Non-VF episodes from ECG, the signal was segmented for 2, 3, and 4 s which are shown in Tables 1, 2, and 3, respectively. Three mother wavelet analyzes were represented to demonstrate whether the mother wavelet impacts the performance of the CNN model. Moreover, the best accuracy obtained is at 4 s which is 97.98% using the Morlet wavelet. But in comparison to other segments the accuracy was the least for 2 s ECG segment. This might be because the total period VF episode was not properly represented in the 2 s ECG. Hence, the mother wavelet could not fit the data accurately.

In the CWT approach for wavelet analysis, the selection of mother wavelet is essential. Improved outcomes can be obtained when the mother wavelet suits the data in a better way through correlation formed by scaling and shifting during wavelet transform. In this context, the Morlet wavelet in comparison to other wavelets produced better performance results for different segments of EMD signals. Therefore, it can be generalized that the choice of the best mother wavelet depends on the pattern of data. If the mother wavelet goes appropriately with the pattern of the data, the performance enhances. A comparison of the proposed approach with the previous similar works is represented in Table 4.

In Table 4, it can be observed that the proposed approach used the CUDDB database which was also utilized in [3, 5–8] to detect VF using different approaches. The proposed approach showed better accuracy than logistic regression, SVM and adaptive neural network classifiers [2], LS-SVM classifier with RBF kernel [3], SVM

**Table 4** Comparison of results with similar works

| Authors               | Database           | Proposed method                                                                                       | Accuracy (%) |
|-----------------------|--------------------|-------------------------------------------------------------------------------------------------------|--------------|
| Acharya et. al. [7]   | MITDB, CUDB, VFDB  | Eleven-layer CNN model                                                                                | 93.18        |
| Tripathy et al. [3]   | CUDB, VFDB         | LS-SVM classifier with linear and RBF kernels                                                         | 89.81        |
| Panigrahy et. al. [5] | CUDB, VFDB, MITDB  | Combined SVM, adaptive boosting (AdaBoost), and DE algorithms                                         | 98.2         |
| Sabut et. al. [6]     | CUDB, VFDB         | Twenty-four features extracted using EMD and VMD and classified using DNN                             | 99.2         |
| Heng et. al. [4]      | SDDB, NSRDB        | Nine-time domain features and seven nonlinear features and classified using SVM for different kernels | 94.7         |
| Mjihad et. al. [2]    | VFDB, Standard AHA | Logistic regression, SVM, adaptive neural network classifier                                          | 96.1         |
| Tseng, et. al. [8]    | CUDB               | 2-D STFT and 2-D CWT with CNN                                                                         | 97           |
| Proposed method       | CUDB               | Hybrid scalogram (EMD with CWT) and deep CNN coupled                                                  | 97.98        |

*MITDB* MIT-BIH Arrhythmia Database, *VFDB* MIT-BIH Malignant Ventricular Arrhythmia Database, *SDDB* Sudden Cardiac Death Holter Database, *NSRDB* MIT-BIH Normal Sinus Rhythm Database, *AHA* American Heart Association Database

classifier for different kernels [4], STFT and CWT coupled with CNN [8] using CUDB database and other datasets as observed in Table 4. The advantageous part of this study in comparison to [2–6] is that the explicit feature extraction problem was avoided. Moreover, although the studies in [7, 8] implemented CNN models, their accuracies were lower compared to the proposed approach.

In this study, the IMFs minimized noise while keeping the features relevant and connected with the original ECG signals. The learning performance may be increased even more by raising the quantity of data and then tweaking the network's hyperparameters. Overall, the performance of a deep neural network improves dramatically as the amount of data grows.

There are some possible limitations to the work. Firstly, VF occurrences are still uncommon, although the database had extensive ECG recordings of numerous individuals. Many portions prior to each VF episode were extracted to alleviate this problem. Secondly, only 35 subjects were used to develop the model. So, the VF episodes were quite rare. The morphology of the ECG signal can also be affected by poor ECG lead location, power-line interference, and other artifacts. Nevertheless, these problems were alleviated to a certain extent by using preprocessing methods. The results showcased a considerable increase in quality in terms of sensitivity,

specificity, and accuracy compared to most related works in Table 4. All in all, this study implemented a novel technique and established its importance in VF detection.

## 5 Conclusion

In this work, a system was developed where ECG signal was converted into 2D images. The categorization of ventricular fibrillation and non-ventricular fibrillation events was done using a deep CNN model with the images of ECG signals from hybrid scalograms. The approach is efficient in extracting sophisticated data derived from ECG signals and yielded a promising classification accuracy. The best accuracy obtained is 97.98% for VF and Non-VF event detection tasks. With appropriate settings, the method generates 2D input with suitable dimensions. To improve the robustness of the algorithm of the proposed approach, in the later period more databases that are available publicly may be used other than CUDb database. This robust method might be advantageous for creating wearable device systems for real-time VF prediction.

## References

1. Baldzizhar A, Manuylova E, Marchenko R, Kryvalap Y, Carey MG (2016) Ventricular Tachycardias: characteristics and management. *Crit Care Nurs Clin* 28(3):317–329
2. Mjahad A, Rosado-Muñoz A, Bataller-Mompeán M, Francés-Víllora JV, Guerrero-Martínez JF (2017) Ventricular Fibrillation and Tachycardia detection from surface ECG using time-frequency representation images as input dataset for machine learning. *Comput Methods Programs Biomed* 141:119–127
3. Tripathy RK, Zamora-Mendez A, de la O Serna JA, Paternina MRA, Arrieta JG, Naik GR (2018) Detection of life-threatening ventricular arrhythmia using digital Taylor Fourier transform. *Front Physiol* 9:722
4. Heng WW, Ming ESL, Jamaluddin ANB, Harun FKC, Abdul-Kadir NA, Yeong CF (2020) Prediction of ventricular fibrillation using support vector machine. In: *IOP conference series: materials science and engineering*, p. 012008. IOP Publishing, United Kingdom
5. Panigrahy D, Sahu PK, Albu F (2021) Detection of ventricular fibrillation rhythm by using boosted support vector machine with an optimal variable combination. *Comput Electr Eng* 91:107035
6. Sabut S, Pandey O, Mishra BSP, Mohanty M (2021) Detection of ventricular arrhythmia using hybrid time–frequency-based features and deep neural network. *Phys Eng Sci Med* 44(1):135–145
7. Acharya UR, Fujita H, Oh SL, Raghavendra U, Tan JH, Adam M, Gertych A, Hagiwara Y (2018) Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network. *Futur Gener Comput Syst* 79:952–959
8. Tseng LM, Tseng VS (2020) Predicting ventricular fibrillation through deep learning. *IEEE Access* 8:221886–221896
9. Nolle FM, Badura FK, Catlett JM, Bowser RW, Sketch MH (1986) CREI-GARD, a new concept in computerized arrhythmia monitoring systems. *Comput Cardiol* 13(1):515–518
10. Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Yen N, Tung CC, Liu HH (1971) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time



- series analysis. In: Proceedings of the royal society of London. Series A: mathematical, physical and engineering sciences 454:903–995
11. Mashrur FR, Islam MS, Saha DK, Islam SMR, Moni MA (2021) SCNN: scalogram-based convolutional neural network to detect obstructive sleep apnea using single-lead electrocardiogram signals. *Comput Biol Med* 134:104532
  12. Roy AD, Islam MM (2020) Detection of epileptic seizures from wavelet scalogram of EEG signal using transfer learning with AlexNet convolutional neural network. In: 23rd International conference on computer and information technology, pp 1–5. IEEE, Dhaka, Bangladesh
  13. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 1–14. San Diego, USA

## Chapter 39

# Lung Cancer Detection Using Ensemble Technique of CNN



**Zebel-E-Noor Akhand, Afridi Ibn Rahman, Anirudh Sarda, Md. Zubayer Ahmed Fahim, Lubaba Tasnia Tushi, Katha Azad, and Hiya Tasfia Tahiat**

**Abstract** Lung cancer is one of the leading contributing factors to the mortality rate. The prevailing types of non-small cell lung cancer (NSCLC) include adenocarcinoma, large cell carcinoma, and squamous cell carcinoma. Studies have shown that 18% of the mortality rate stems from this disease. This is attributable to sub-standard diagnosis techniques and inefficient treatments available to cure metastasis. Hence, this paper opts to employ transfer learning techniques by using different state-of-the-art, pre-trained models to detect lung cancer and classify it into four groups, namely adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal using chest CT scan images, followed by conducting a comparative analysis of their performances. The models implemented are ResNet101, VGG16, InceptionV3 and DenseNet169. Moreover, the paper proposes a new CNN model using the ensemble technique which is an amalgamation of ResNet101 and InceptionV3 that are employed initially. In addition to that, it introduces another 11 layered CNN model built from the outset. The dataset used for this study is called chest CT scan images which is retrieved from Kaggle. The models' performances are analyzed utilizing different metrics like accuracy, precision, recall, F1-score, and AUC. The ensemble of ResNet101 and InceptionV3 models has achieved the highest accuracy of 93.7%, on this particular dataset, based on which a Web app is deployed to demonstrate real-world application.

**Keywords** NSCLC · CNN · RNN · ANN · KNN · AI

## 1 Introduction

Lung cancer is a fatal disease that can potentially terminate lives. In the United States, lung cancer accounts for roughly 225,000 cases, 150,000 deaths, and 12 billion dollars in healthcare costs per year [1]. It occurs when lung cells begin to mutate and

---

Z.-E.-N. Akhand (✉) · A. I. Rahman · A. Sarda · Md. Z. A. Fahim · L. T. Tushi · K. Azad · H. T. Tahiat  
Department of Computer Science and Engineering, Brac University, 66 Mohakhali, Dhaka 1212, Bangladesh  
e-mail: [zeb.akhand@gmail.com](mailto:zeb.akhand@gmail.com)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023  
M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, [https://doi.org/10.1007/978-981-19-7528-8\\_39](https://doi.org/10.1007/978-981-19-7528-8_39)

497

uncontrollably divide, killing the surrounding healthy cells. If the cancer is metastatic, then it will disseminate to other parts of the body. This paper specializes in detecting non-small cell lung cancer (NSCLC) which can be divided into three categories, namely adenocarcinoma, large cell carcinoma, and squamous cell carcinoma. Adenocarcinoma is the most common form of NSCLC, contributing to about 40% of lung cancer deaths [2]. It begins in cells that ordinarily exude substances like mucus [3]. Accordingly, adenocarcinoma forms in the alveoli, hampering the gas exchange process. Large cell carcinoma is known for its atypical, large cells, infesting almost every inch of the lungs. It can grow and spread faster than the other two types of NSCLC. Lastly, squamous cell carcinoma can be relatively more malignant as most cases manifest in the final stages [4]. It infects the bronchi, causing about a quarter of deaths from lung cancer.

Technical advancements like artificial intelligence and deep learning play pivotal roles in the detection of lung cancer. Pulmonary nodule screening is tedious and requires a great deal of attention to avoid missed diagnosis. Combining CAD and manual diagnosis significantly improves the screening process and allows very little space for error. In fact, it has been proven that using CAD for detection brings about better results than manual diagnosis, owing to its high sensitivity and specificity. Therefore, deep learning not only develops the lung cancer detection process but also sets a better picture for prognosis, leading to reduced mortality rates [5]. The use of artificial intelligence is widespread, mainly in medical image segmentation, pathological diagnosis of lung cancer, extraction of lung nodules, and searching the tumour marker in lung cancer detection.

Convolutional neural network (CNN), a domain of deep learning, is the most common algorithm used for image recognition and classification. It has been dominating the machine vision space for many years. In fact, the best performances for many databases such as MNIST database (99.77%), CIFAR10 dataset (97.6%), and the NORB database (97.47%) were drastically improved with research conducted using CNN [6]. A CNN model typically has an input layer, output layer, convolution layers, pooling layers, fully connected layers, and normalization layers (ReLU) [7]. For more complicated models, additional layers can be employed. Having this array of well-trained networks, CNN is able to efficiently utilize millions of images being fed to it and bring about outstanding results that surpass the previous success [8]. Furthermore, research has proven that CNN works better than its competitors; recurrent neural network (RNN), K-nearest neighbours (KNNs), and artificial neural network (ANN), as CNN has the ability to detect important features of images automatically, without the need for manual supervision [9].

## 2 Goals

This paper attempts to pursue the following objectives to conduct a thorough comparative analysis.

i. **Employ five pre-trained CNN models and one self-constructed model**

The previous researches simply used at most two model for the diagnosis. This paper employs five pre-trained models and a self-constructed model that can automatically classify and detect important features of images without the need of human supervision for efficient lung cancer diagnosis.

ii. **Conduct a comparative analysis on the performances of the models**

The previous researches did not perform a comparative analysis on the single model used with other existing models. Therefore, after employing the six models, the paper seeks to perform a comparative analysis on the results that the models yield.

iii. **Determine the Best Working Models**

After doing a comparative analysis, the best working model is to be determined based on accuracy and other performance metrics. This can aid in hospitals deciding which model to use for lung cancer diagnosis.

iv. **Design an Ensemble Model with the Best Working Models**

Using the ensemble technique, the best working models are to be merged in order to create a new powerful model that can detect and classify lung cancer more accurately.

v. **Deploy a Web Application**

Subsequent to building the ensemble model, a Web application is to be deployed in order to demonstrate real-world application. This can be helpful as hospitals or doctors will receive more clarity on the model and hence be more willing to adopt CNN-based detection techniques.

### 3 Related Work

The first paper written by Basha et al. [10] applied an enhanced neural network-based algorithm for predicting lung cancer more accurately in the initial stage. They used the chest CT scan images dataset from Kaggle. For lung cancer image prediction, PSO is used to combine CNN and ELM. Moreover, to improve the accuracy of detecting early stage diagnosis, the authors suggested the optimized extreme learning machine (OELM) model. Compared to existing models such as CNN, ELM and ANN, it is clear that the proposed OELM offers higher accuracy, specificity, sensitivity, and precision.

In the next study [11], Han et al. utilized a common CNN model known as VGG16, to detect lung cancer. This paper also made use of the chest CT scan images dataset from Kaggle. The proposed model was trained to categorize CT scan images into four sectors to identify different types of lung cancers. The trained model was then coupled with a chatbox and a graphical user interface. Additionally, the scientists employed a variety of algorithms with various batch sizes to increase the accuracy of lung cancer detection. As a result, it was demonstrated that the proposed 64 batch-size self-designed CNN algorithms outperformed the competition by displaying higher accuracy. The researchers proclaimed to create a Website for early lung cancer detection in the near future.

The chest CT scan images dataset from Kaggle was then classified into four forms of lung cancer by Sari et al. [12] using a modified ResNet50 architecture and transfer learning technique. After the modification of ResNet50 architecture, it showed high accuracy of 93.33% and a sensitivity, precision, F1-score of 92.75%, 93.75%, and 93.25%, respectively. Furthermore, EfficientB1 and AlexNet were being compared with the modified ResNet50 architecture, and it showed that the latter worked best during the mathematical analysis. However, they believe that if deep learning method is implemented, they can get better results in identifying lung cancer and other multivariate lung diseases.

The final research [13], Sasikala et al. proposed a method that employs CNN to detect malignant or benign tumours in the lung using CT images. The proposed framework is able to distinguish the presence and non-appearance of cancerous cells with an exactness of about 96%. In addition, 100% specificity was acquired in the study which depicts zero false positive detection. However, by comparing with prior CNN-based systems, their proposed model performed better in terms of accuracy, specificity, and sensitivity. For future research, large datasets will be used to train their proposed model for identifying the size and shape of lung cancer. Besides, 3D CNN including the hidden neurons with deep networks helps to enhance the overall accuracy of their suggested method.

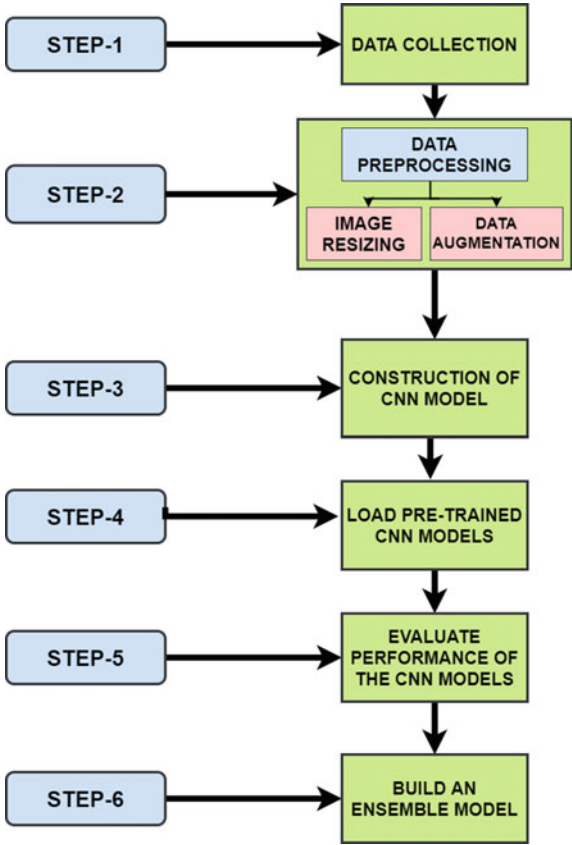
## 4 Methodology

This paper utilizes five pre-trained models, namely VGG16, ResNet101, InceptionV3, and DenseNet169 and conducts transfer learning to observe their functioning. Moreover, it proposes a new CNN model built from scratch and a model made using the ensemble technique that combines the two best pre-trained models. The dataset contains images of adenocarcinoma, large cell carcinoma, and squamous cell carcinoma. All of these are fed into each model so they can determine and allocate the images into their respective categories. Using the results obtained, a comparative analysis is made to determine the best performing model in terms of accuracy and other performance metrics. Figure 1 depicts the full workflow of the methodology.

### 4.1 Data Collection

The chest CT scan images dataset in PNG format from Kaggle [14] is used for this paper. The owner of the dataset collected these images from various sources for a cancer detection project of their own. It consists of four types of cancer images, namely adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal CT scan images. The data is already split into three categories that are training (70%), testing (20%), and validation (10%).

**Fig. 1** Workflow of the methodology



**4.2 Data Preprocessing**

Image transformation tasks, in which the input and output are both images, are important in a wide range of applications, including edge detection, semantic segmentation, and black-and-white image colorization [15]. Hence, this method is used to eliminate unnecessary details of images that do not play a part in enhancing the accuracy of models. In fact, it enriches the valuable aspects of the raw data, productively contributing to the performance of the CNN models [16]. This paper refines data in two parts: image resizing and data augmentation. Images found in the dataset are of random sizes [17]. This inconsistency can potentially bring down the models’ accuracy. Hence, all the images are resized to  $224 \times 224$  pixels before being fed to the models. In addition, data augmentation is carried out on the dataset to help remove sparse data [18]. This technique manually modifies the dimensions of the training dataset, producing multiple altered copies of the selected images. The following geometric modifications are applied: Rescale, Range, Rotation, WidthShiftRange, HeightShiftRange, HorizontalFlip, and VerticalFlip.

**Table 1** Summary of the 11 layered CNN model

| Layer (type)                  | Output shape         | Param #   |
|-------------------------------|----------------------|-----------|
| conv2d (Conv2D)               | (none, 222, 222, 32) | 896       |
| conv2d_1 (Conv2D)             | (none, 220, 220, 32) | 9248      |
| max_pooling2d (MaxPooling2D)  | (none, 110, 110, 32) | 0         |
| conv2d_2 (Conv2D)             | (none, 108, 108, 64) | 18,496    |
| max_pooling2d_1 (MaxPooling2) | (none, 54, 54, 64)   | 0         |
| conv2d_3 (Conv2D)             | (none, 52, 52, 128)  | 73,856    |
| max_pooling2d_2 (MaxPooling2) | (none, 26, 26, 128)  | 0         |
| dropout (Dropout)             | (none, 26, 26, 128)  | 0         |
| flatten (Flatten)             | (none, 86,528)       | 0         |
| dense (Dense)                 | (none, 64)           | 5,537,856 |
| dense_1 (Dense)               | (none, 4)            | 260       |

**4.3 Construction of CNN Model**

A new CNN model of 11 layers is constructed for further comparison. This comprises of convolution layer, pooling layer, and fully connected layer. Apart from that, dropout layer is also added to enhance the efficacy of the model. The layers are described in more details below:

1. **Convolutional Layer:** Four layers of Conv2D are selected for this layer.
2. **Pooling Layer:** This consists of three Maxpooling2D layers that calculates the maximum value of each feature map.
3. **Fully Connected Layer:** This layer is composed of two additional layers. The flatten layer is used after selecting the pooling layer to level out the entire network. After this, two dense layers are applied so that the outputs from the previous layers are fed as inputs to all the model’s neurons. This helps to reduce computational complexity, which, in turn, accelerates the training time.

Table 1 depicts the summary of the newly constructed CNN model.

**4.4 Load Pre-trained Models**

The Keras library is employed to load the pre-trained models for transfer learning. This is a robust and user-friendly, deep learning application programming interface that is written in Python [19]. It facilitates simple training of neural network models in merely a few lines of code by encasing the structured numerical computational

libraries, Theano, and TensorFlow. The following is a list of the models that are used in this study:

- I. **ResNet101**
- II. **VGG16**
- III. **InceptionV3**
- IV. **DenseNet169.**

The selection of these pre-trained models was predicated from the potential to achieve consequential accuracy in classifying lung cancer from CT scan images. It has been proven that VGGNet and ResNet are able to identify modalities of medical images more accurately as they have different depths of pre-training on ImageNet [20]. In contrast, other experiments substantiate that training deep learning models with transfer learning from ImageNet pre-trained CNN models tend to yield significant accuracy at classifying medical images [21]. These CNN models include DenseNet and Inception. Hence, this study resorts to take at least one variant from each of these CNN models to make the selection as diverse and versatile as possible. This, in turn, can help put forward a good comparative analysis of their performances in detecting lung cancer from chest CT scan images.

#### 4.5 Evaluate Performance of the CNN Models

The paper tries to completely examine the performance of each model by calculating F1-score, precision, accuracy, recall, and AUC curve in order to give a well-informed comparative evaluation of the pre-trained, ensemble, and newly developed CNN models. The equations for each of these metrics are as follows [22, 23]:

$$\textbf{Precision: } PV = \frac{TPV}{TPV + FPV}$$

$$\textbf{F1-score: } F1 = 2 \times \frac{PV \times RV}{PV + RV} = 1 - FDRV$$

$$\textbf{Accuracy} = \frac{TPV + TNV}{CP + CN} = \frac{TPV + TNV}{TPV + TNV + FPV + FNV}$$

$$\textbf{Recall: } PV = \frac{TPV}{TPV + FNV} = 1 - FDRV$$

The abbreviations for the symbols are as follows: PV = Precision value, RV = Recall value, TPV = True positive value, FPV = False positive value, FDRV = False discovery rate value, TNV = True negative value, CP = Condition positive, CN = Condition negative, FNV = False negative value.



**AUC:** AUC or area under the curve is a metric that [24] calculates how efficiently a classifier is able to differentiate amongst the sets. Since AUC is unvarying with respect to scale and classification threshold, it delivers a composite measure of functioning across all viable classification thresholds. Furthermore, it can also depict the level of separability. A high AUC score denotes that the model is very good at classifying positive and negative sets.

#### ***4.6 Build an Ensemble Model***

After training and testing the pre-trained models individually, it was found that ResNet101 and InceptionV3 perform well for detecting lung cancer on this dataset. These models have achieved an accuracy of 90.95% and 91.74%, respectively. With a view to achieve better performance than the pre-trained models, this study has opted to build an ensemble model of the best two performing models. In order to do so, the outputs of both ResNet101 and InceptionV3 have been put in a concatenation layer. Following that, an extra dense layer was added, followed by another dense layer with a single output and an activation function, ‘softmax’, as this study involves multi-class categorization. The accuracy of this new ensemble model is 93.71%, which is an increase of 1.97% precisely.

### **5 Analyze and Compare All the Models’ Performance Results**

The accuracy of DenseNet169, InceptionV3, ensemble (ResNet101+InceptionV3), ResNet101, and VGG16 is 88.96%, 91.74%, 93.70%, 90.95%, and 82.77%, respectively, which is shown in Table 2, along with other evaluation metrics like precision, recall, AUC, and F1-score. We found out that models with more layers tend to have better accuracy even though InceptionV3 is an exception but the trend we see is that the more the number of layers and the denser the architecture, the better the accuracy. Amongst all the transfer learning models, the combined model of ResNet101 and InceptionV3 works best with a considerable amount of parameters. The accuracy reached by the combined model is a state-of-the-art accuracy for this particular dataset. However, our proposed ensemble model also has an accuracy of 93.7% which is a competitive accuracy compared to the state-of-the-art models that are used. Table 2 shows a comparison of all the models’ performances.

Figure 2 depicts both train and validation accuracy and train and validation loss of the ensemble model. Here, it can be seen that the training and validation accuracy are converging with one another at a good rate, indicating that there is neither any overfitting nor any underfitting. It further signifies how powerful the ensemble model is at this particular classification. On the other hand, the training and validation loss

**Table 2** Comparison of performances in terms of the performance matrices

| Model                             | Accuracy (%) | Precision (%) | Recall (%) | AUC (%) | F1-score (%) |
|-----------------------------------|--------------|---------------|------------|---------|--------------|
| Ensemble (ResNet101+ Inceptionv3) | 93.70        | 83.85         | 82.27      | 95.90   | 82.45        |
| InceptionV3                       | 91.74        | 83.92         | 82.83      | 95.72   | 83.39        |
| ResNet101                         | 90.95        | 84.30         | 78.41      | 95.49   | 81.24        |
| DenseNet169                       | 88.97        | 78.20         | 77.46      | 93.88   | 77.83        |
| VGG16                             | 82.78        | 89.52         | 35.24      | 86.74   | 50.57        |
| New CNN model                     | 72.94        | 45.78         | 44.76      | 78.72   | 45.26        |

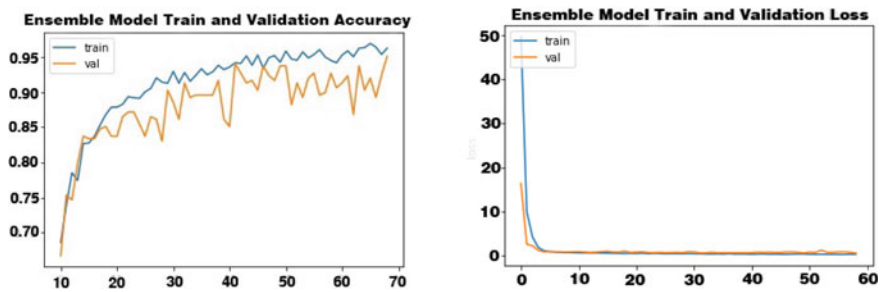


**Fig. 2** Train and validation accuracy and train and validation loss of the ensemble model

graph represent that loss is getting lower as the model is moving forward with the training, and it stops at almost zero (0) which again implies the strength of the model.

## 6 Deployment of Web Application

A modest Web application based on the ensemble model from this study has been developed and hosted on Heroku to demonstrate a real-world implementation. The user selects a lung CT image and uploads it to the Website to see if the CT scan indicates any evidence of lung cancer. The Website supports image formats such as JPG, PNG, and JPEG, which are the most often used. After analyzing the features of the input image, the application is able to determine and classify the image as squamous cell carcinoma, adenocarcinoma, large cell carcinoma, or normal. The Website is designed to be run from any device that has a Web browser. Therefore, the Website’s user interface has been kept responsive and lightweight. In addition, the application does not keep track of user history, which helps to maintain patient confidentiality. The successful Web deployment is a testament of how easily the ensemble CNN model can be implemented. It further validates the model’s potential

of having a good business value. The source code for our proposed ensemble model's Web application is available at [25].

## 7 Conclusion

Lung cancer is a leading cause of death worldwide. In order to reduce the severity of cases, a meticulous diagnosis is required. But the present day detection methods are quite ineffective and tedious. This paper aims to improve the detection of cancer using a powerful CNN model. The research also employs the ensemble technique to combine two CNN models into one and evaluate their performance. The secondary aim of the paper was to do a comparative study on the effectiveness of different CNN models. Since the previous researchers aimed to only analyze the best-suited model, this paper aims to find the reason behind the model's yield performance. The ensemble model proves to have the highest accuracy of 93.7%. The accuracies of the other CNN models were lower than the ensemble model. DenseNet169, InceptionV3, and ResNet101 were 88.97%, 91.74%, and 90.95%, respectively. In the near future, the paper seeks to apply meta learning and design not only a more powerful CNN model that can yield a higher accuracy, but also a model that has the potential to be employed commercially in the detection of lung cancer.

## References

1. Choi W-J, Choi T-S (2013) Automated pulmonary nodule detection system in computed tomography images: a hierarchical block classification approach. *Entropy* 15(2):507–523
2. Legato MJ, Bilezikian JP (2004) Principles of gender-specific medicine, vol 2. Gulf Professional Publishing
3. Araujo LH, Horn L, Merritt RE, Shilo K, Xu-Welliver M, Carbone DP (2020) Cancer of the lung: non-small cell lung cancer and small cell lung cancer. In: Abelloff's clinical oncology. Elsevier, pp 1108–1158
4. Lonardo F, Rusch V, Langenfeld J, Dmitrovsky E, Klimstra DS (1999) Overexpression of cyclins D1 and E is frequent in bronchial preneoplasia and precedes squamous cell carcinoma development. *Cancer Res* 59(10):2470–2476
5. Perla F, Richman R, Scognamiglio S, Wüthrich MV (2021) Time-series forecasting of mortality rates using deep learning. *Scand Actuar J* 7:572–598
6. Li Q, Cai W, Wang X, Zhou Y, Feng DD, Chen M (2014) Medical image classification with convolutional neural network. In: 2014 13th international conference on control automation robotics & vision (ICARCV). IEEE, pp 844–848
7. Hussain M, Bird JJ, Faria DR (2018) A study on CNN transfer learning for image classification. In: UK workshop on computational intelligence. Springer, pp 191–202
8. Sharma N, Jain V, Mishra A (2018) An analysis of convolutional neural networks for image classification. *Procedia Comput Sci* 132:377–384
9. Dertat A (2017) Applied deep learning—Part 4: Convolutional neural networks. *Towards Data Sci*
10. Basha BMF, Surputheen DMM (2020) Optimized extreme learning machine based classification model for prediction of lung cancer. *Int J Electr Eng Technol (IJEET)* 323–332

11. Han Z, Liu L, Liu Y, Mao H, Diagnose lung cancer and human-machine interaction based on CNN and NLP. In: 2021 2nd international seminar on artificial intelligence, networking and information technology (AINIT). IEEE, pp 175–179
12. Sari S, Soesanti I, Setiawan NA (2021) Best performance comparative analysis of architecture deep learning on CT images for lung nodules classification. In: 2021 IEEE 5th international conference on information technology, information systems and electrical engineering (ICI-TISEE). IEEE, pp 138–143
13. Sasikala S, Bharathi M, Sowmiya BR (2018) Lung cancer detection and classification using deep CNN. *Int J Innov Technol Expl Eng (IJITEE)* 259–262
14. Hany M (2020) Chest CT-scan images dataset [online]. Available at: <https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscanimages>
15. Gomes VJ, Alavee KA, Sarda A, Akhand Z-E et al (2021) Early detection of diabetic retinopathy using deep learning techniques. Ph.D. dissertation, Brac University
16. Tang S, Yuan S, Zhu Y (2020) Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery. *IEEE Access* 8:149487–149496
17. Saponara S, Elhanashi A (2022) Impact of image resizing on deep learning detectors for training time and model performance. In: International conference on applications in electronics pervading industry, environment and society. Springer, pp 10–17
18. Rahman AI, Bhuiyan S, Reza ZH, Zaheen J, Khan TAN (2021) Detection of intracranial hemorrhage on CT scan images using convolutional neural network. Ph.D. dissertation, Brac University
19. J. Brownlee (2021) Your first deep learning project in python with keras step-bystep
20. Yu Y, Lin H, Meng J, Wei X, Guo H, Zhao Z (2017) Deep transfer learning for modality classification of medical images. *Information* 8(3):91
21. An G, Akiba M, Omodaka K, Nakazawa T, Yokota H (2021) Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images. *Sci Rep* 11(1):1–9
22. Hussain E, Hasan M, Hassan SZ, Azmi TH, Rahman MA, Parvez MZ (2020) Deep learning based binary classification for Alzheimer's disease detection using brain MRI images. In: 2020 15th IEEE conference on industrial electronics and applications (ICIEA). IEEE, pp 1115–1120
23. Das S, Aranya ORR, Labiba NN (2019) Brain tumor classification using convolutional neural network. In: 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT). IEEE, pp 1–5
24. Narkhede S (2018) Understanding AUC-ROC curve. *Towards Data Sci* 26(1):220–227
25. afridirahman97. Afridirahman97/lung-cancer-detector [online]. Available at: <https://github.com/afridirahman97/Lung-Cancer-Detector>

## Chapter 40

# Identification of the Similarity of Bangla Words Using Different Word Embedding Techniques



Aroni Saha Prapty and K. M. Azharul Hasan

**Abstract** From the browser search bar to the voice command in a computer system, everywhere Natural Language Processing (NLP) is broadly used nowadays. NLP is mainly used for the purpose of converting user data into compatible search materials. Some specific applications of NLP are sentiment analysis, text mining, news classification and many more. Those concepts are also very important in case of Bangla language processing. A number of researches have been conducted to process Bangla language to produce potential output from the user data of Bangla-speaking people. Word embedding is a major part of NLP and bears a vast importance in processing languages. In this work the word embedding for Bangla language is mainly focused. In doing the embedding words in Bangla language we have used Word2Vec model and FastText model with Gensim library. The Word2Vec model produces vector of words and similarly does the FastText model while FastText breaks the words into small blocks to train into machine. There are a very countable number of researches regarding the word embedding in Bangla language. In the proposed word FastText produces promising result over the Word2Vec model although no numerical conclusion was possible to derive at this phase of proposed implementation.

**Keywords** Natural language processing · Word embedding · Word2Vec · FastText · Word clustering

## 1 Introduction

From among thousands of application Natural Language Processing (NLP) some are textual entitlement and finding similar words. The word proposed in this paper finds the similar words of Bangla language from a dataset of Bangla newspaper.

---

A. S. Prapty (✉) · K. M. A. Hasan

Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh  
e-mail: [aronisahaprapty@gmail.com](mailto:aronisahaprapty@gmail.com)

K. M. A. Hasan

e-mail: [az@cse.kuet.ac.bd](mailto:az@cse.kuet.ac.bd)

The concept of finding similar word can be obtained using word embedding. A word embedding is a learnt representation for text in which words with related meanings are represented [1]. In NLP, we sometimes need to convert our words formatted data in numerical forms which holds numerically pointed values against words understandable by the machine. The word embedding technique groups the words according to the similarity of the meaning of those words for representing in similar group [2]. However the conducted research work used a pair of embedding techniques, one is Word2Vec and another is FastText with the Gensim methodology. The one-hot vector is a traditional way of representing words, it has the element '1' as target and other elements to be '0'. Chain length of such vector is the count of unique entries in the database. Later on individual words are maintained according to position of alphabetic letters [3]. The word representation in one-hot vector is much easier and implementation complexities is less, but as usual few issues to be considered. The relationship between two words can't be inferred in this vector presentation. As an example, the words "encounter" and "undergo", bear relevant meaning, thus have target value of '1', differ one from another. Again redundancy of word must be handled sincerely as several redundant '0' might be spotted inside the presently active vectors. Some other ways to be thought for word representation to solve these issues.

The Word2Vec model perfectly solves this type of anomalies by returning the index numbers for the words which are our target. It is necessary to analyze words in neighbor to describe the actual target for feeding a neural network. All the layers hidden part mainly encodes the represented target word. Variations of Word2Vec model are discussed as Skip-Gram and Continuous Bag of Words (CBOW). In Skip-Gram, we consider the input as target word, the words surrounding the target (input) word are the outputs. Take a sentence "I need a cup of coffee", think that the input is "a", then the outputs are "I", "need", "cup", "of", and "coffee", when the size of the window is 5. Every input/output are one-hot encoded with equal dimension. An equal embedding sized hidden layer is existing in the network and it is smaller in size compared to the vector size of input/ output. The finishing layer of the output contains an activation logic to describe each of the output entity in similar sense in the presented scenario. The whole concept and structure is pictured in Fig. 1.

The embedding of the target word can be founded when the word vector has been fed inside network and layers hidden from outside is extractable of such datum. The size of the represented vector shrinks and varies from  $V$ , the size of the vocabulary under consideration toward  $N$ , which indicates the hidden layer length in Skip-Gram presentation. The produced vectors are capable enough in detailing the dependencies of words with other words. Two neighboring words indicate a vector that tells about part of the speech or mode of time. Its concept has been made clear in Fig. 2.

Another variation of Word2Vec model is Continuous Bag of Words (CBOW) which is very near with the Skip-Gram method. The basic difference between the two division of Word2Vec model is interchanging of inputs/output in CBOW what was done in Ship-Gram implementation. The core idea behind CBOW is to find out which words are most likely to appear in a dictionary. The word vectors generated from CBOW has a different representation from the word vector generated from Ship-Gram model. The networks are fed with targeted words in CBOW and the hidden

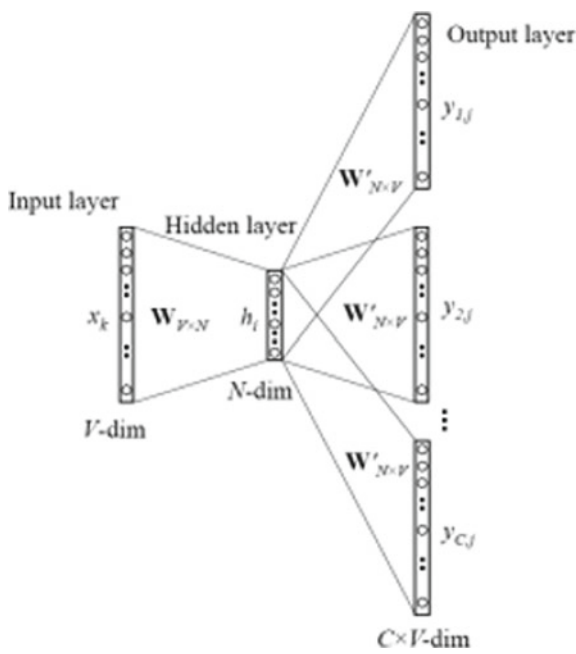


Fig. 1 Skip-Gram network structure [4]

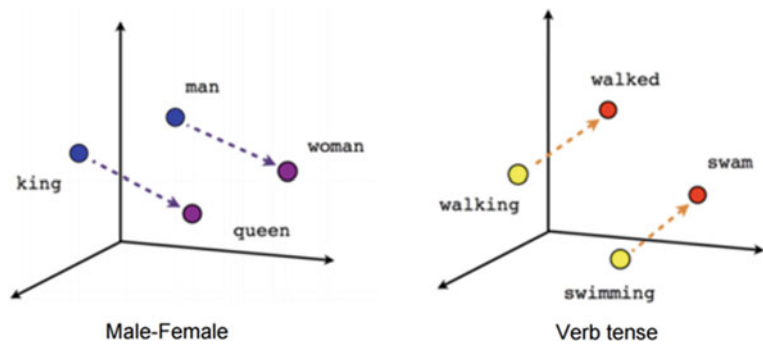
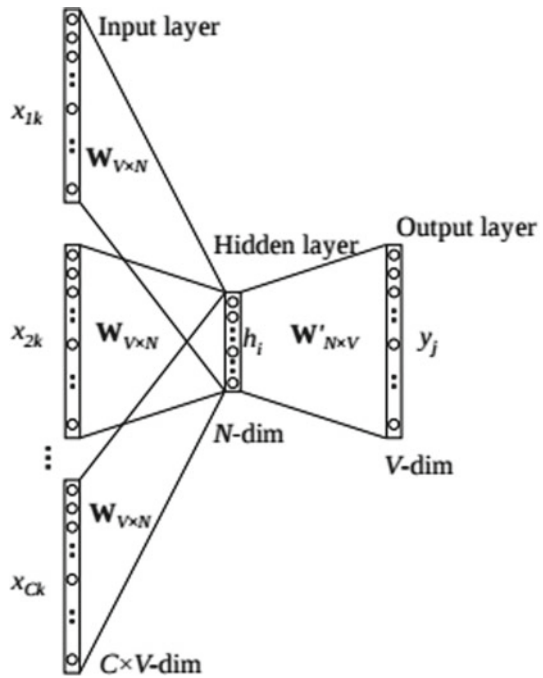


Fig. 2 Visualization of word vector from Skip-Gram model [5]

layer extracted average information are taken. For the purpose of clarification, “He holds an excellent portrait” and “He talked a gentle woman” these two sentences are assumed. The word vector for “a”, feed “He holds excellent portrait”, and “He talked a gentle woman” into the neural network and take the average of the value in the hidden layer. Skip-Gram feeds the target word one-hot vector as input while CBOW feeds all the words rather than the target in network. The Skip-Gram shows better result for rare words. After all Skip-Gram and CBOW perform similar. The main mechanism of CBOW is shown in the following Fig. 3.

**Fig. 3** CBOW network visualization [4]



Apart from the concept of Word2Vec we need the FastText model of word embedding in this work. This model is an updated version of Word2Vec model which was developed by the Facebook authority in 2016 [6]. Earlier word embedding methods, the neural network was fed with every words, but FastText has changed the concept completely. The words broken and clustered in few sub-words. To illustrate the concept tri-grams segmentation of a word football are foo, oot, otb, tba, bal and all. All the boundary points around words are ignored. The vector that is embedded for the word football using FastText technique is the sum of all the  $n$ -grams which were produced. When training into the neural network is completed, it will generate word embedding for each  $n$ -grams obtained in dataset for training purpose. The problem regarding the rare words now can be solved. The rare words can be represented properly as there is high probability that some of the  $n$ -grams of the rare words may appears in other words.

## 2 Related Works

The basic goal of Natural Language Processing (NLP) is to analyze any language for preparing machine representable format of the language pattern. Many aspects are available for processing for different types of languages. Many techniques have been



developed over time for such processing. The word embedding technique is one of them whose major job is to find the vector presentation of words in a sentence. The meaning of any word within a sentence greatly depends on the surrounding words as well as the position of the word in the sentence. The word embedding technique does the job of generating numeric vector for words and feed them into machine to produce some meaningful result. Different types of word embedding techniques are in action today. Window-based methods in processing of word embedding have been introduced by many researchers. Some of them are Word2Vec, GloVe, FastText, etc. among which Word2Vec is most well-known developed by a team researcher including Mikolov et al. [7]. Authors in this work have proposed two novel architectures: Continuous Bag of Words (CBOW) and Skip-Gram (SG). This approach was used in many of the document classification [8] and sentiment analysis [9]-based NLP projects. Another research work conducted by a team headed by Lund K. has discussed about a method of factorization of matrix [10]. Their job was to represent the word with similar meaning and showed the relationship between words in a form of matrix. Further several of the normalization techniques were adopted to improve the performance of word semantic analysis [11]. Bojanowski et al. were pioneer in considering the concept of morphological data for finding relationships among words [12]. Some word embedding methods like FastText and Adaptive models were designed especially for Hindi language [13]. This word embedding implemented in many researches and projects to represent information and utilized as language features in NLP. Another research showed word clustering using the  $N$ -gram technique [14]. In the work semantic and contextual sense of words was analyzed with the  $N$ -gram model. The findings were that similar context and similar meaning words are in the same group. For classification of Bangla language documents, different neural networks have been implemented by Mandal and Sen [15]. This research had five classes to be classified using tf-idf on a mini local dataset and obtained an accuracy of 89.14% with SVM classifier. Alam et al. [16] have worked on a Bangla news article dataset where five labels of the news were considered. The work implemented machine learning algorithms showed it performs better with Word2Vec and tf-idf methods. Word embedding technique is very important for NLP of Bangla language. From the research works of different authors we have been inspired to work with a word embedding technique with Word2Vec and FastText method. The FastText method was developed by a team of Facebook researchers. This technique produces a considerable result over Word2Vec method in our research.

### 3 Methodology

In previous times many techniques are applied for clustering of word in different methods and they produce high accuracy for English word a Bangla language has variety of structures and difficult to handle the complexity. Here we apply some methods that represent vectors of word to find the similarity of words using the context.

### ***3.1 Data Collection***

The performance of any model depends importantly on dataset which has been used in train, validate and test phase of implementation. An embedding model in any language processing requires a validated set of data without any sort of excess information. In this proposed work we had to collect Bangla news articles from different sources. The articles article contains 111111 articles regarding different social and trending issues, editorial, interviews of many persons, sports and entertainment news and many more which contained the news articles from a spread areas of life in Bangla language.

### ***3.2 Word, Emoticons and Punctuation Remove***

The proposed methodology works on Bangla language word embedding technique. This is the reason the dataset we would provide to train and test in the machine must be pure Bangla language. As the dataset we had considered to use news articles which might contain word of other language. In news, people may use other language words, sometime some complete sentence to express their feelings, commenting an article or tagging purpose. Basically the Bangla words can be separated by the alphabet. The words with letter of non-Bangla alphabet are removed to clean the dataset. Another way to remove certain overlapping alphabets is Bangla Regular Expressions (BREs). The articles of multiple newspaper we collected contain different types of emoticons symbol that people used to use in their writings. Those symbols bear no meaningful standard in language processing. Those also meant different emoticons for different expression by different person. To remove them we had to traverse through the whole dataset and replace the ASCII values of those symbol. Similarly the dataset contains different punctuations. The fact is only word makes sense in word embedding techniques the punctuation marks are valueless. So the punctuations within the sentences of the dataset have to be removed. In the similar way we removed the emoticons from our database, traversing the news we had replace punctuation marks like , ‘ “ ? etc.

### ***3.3 Data Preprocessing***

To produce effective result Bangla word embedding technique normalization of characters is important in managing Bengali text. Sometime the alternative structures of sentences are used with parallel words or symbols. For example, the word টাকা” (cash) is replaced by ¢” symbol, মিনিট (minute) is written in a short form as মিঃ” and কাম্পানি” (company) as কাং” and so on. To get rid of such situations dataset normalization is needed. The normalized dataset go through the data and converts

those short forms into complete format to eliminate the recognition anomalies. In word embedding machines mainly rely on words not sentences. Our dataset contains sentences from different news portals which are paragraphs with different sentences. We need to convert the sentences into words. This is done by tokenization technique. The following is an example of tokenization of Bangla sentence. “রাশিয়া ইউক্রেন যুদ্ধের ভয়াবহতা” this sentence is tokenized as “রাশিয়া”, “ইউক্রেন”, “যুদ্ধের” and “ভয়াবহতা”. In a sentence several words are found which has less significance of meaning. We had removed such junction words “এবং” (and), “কাথায়” (where), “অথবা” etc. are few of the examples. As the words have no impact on emotions or relevant situations those can easily be removed without impacting the total database meaning.

### 3.4 Building Models

**Word2Vec Word Embedding using Gensim Package** Word embedding is defined as a one of the most leading techniques to represent a word in a document capturing the context of documentation vocabulary maintaining the similarity with the association with alternative words retaining syntactic and semantic relationships. Remarkably word embedding can be defined as representation as a vector of any specific word. Among the most significant and widely used methods for learning word embeddings is Word2Vec. The main motive of Word2Vec is to assign a group of particular words based similarity based on the context that occupies adjacent spatial points. In one-hot encoding individual word holds one dimension regarding the absolute dimensions of the entire sentence and in consequence rest of the dimensions for a particular word remain having no projection. In distributed representation it appraised that there exist dependencies among the words analyzing the context that share higher dependencies unless one-hot encoding where each words are unconventional to other. Suppose we consider a sentence where window size is one (1) like Fig. 4.

Calculating the cosine similarity, the correlation of words is measured by the vector is close to 1 whereas the angle is adjacent to 0. Similarity  $(w_1, w_2) = \cos(\theta) = (w_1 w_2) / \|w_1\| \|w_2\|$ . For example, the word “গ্রামীণ” calculates the following cosine distances among the words based on similarity (Table 1).

Word2Vec compares words with the other words in the training corpus that are adjacent to them. In CBOW the context is used to anticipate the given target word

**Fig. 4** Visualization of window size partition in Word2Vector



**Table 1** Cosine distance of words

| Rank | Word              | Cosine distance |
|------|-------------------|-----------------|
| 1    | নারী ,স্ত্রী      | 0.914013        |
| 2    | মহিলা ,স্ত্রী     | 0.883214        |
| 3    | মেয়ে ,স্ত্রী     | 0.750165        |
| 4    | রমণী ,স্ত্রী      | 0.620022        |
| 5    | মেয়েছেলে ,স্ত্রী | 0.531408        |
| 6    | ছেলে ,স্ত্রী      | 0.231045        |

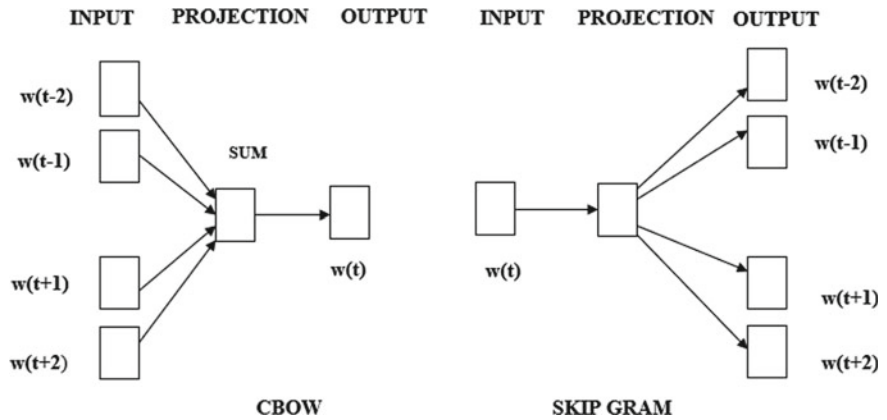


**Fig. 5** Visualization of Skip-Gram and CBOW

whereas the Skip-Gram uses a word to generate the similarity of the target word which works precisely with generous amount of data (Fig. 5).

According to Fig. 2  $w(t)$  represents the weight that is predicted by given context words which are considered as input and it is attained by the neurons. To calculate the target word the value  $P$  is estimated which is derived from the given input in the neural network. Here the shallow neural network is applied where the word embedding technique relies on the disposal quantity of the neuron. Regardless the size of the context word the size of the embedding will be used to emblemize it. The performance of the CBOW enhances its performance with more frequent data with low consumption of memory. In the Skip-Gram the contextual word is predicted, whereas target word is served as output and hidden layer is acted for the concern to attain the weights. According to the size of the window randomly one word is picked up from the contextual word and further  $P$  value is estimated which is tightly nearby the input word.

To work with word embedding technique the Word2Vec class is provided by Python library named Gensim. This library is applied for both CBOW and Skip-Gram.

### Basic Steps

**Corpus:** The dataset contains Bangla sentences that are reserved in a text file and those sentences are evaluated as strings. To generate out target output those strings are gone through some preprocessing steps as the strings are not suitable to feed in the proposed neural network model.

**Tokenizing:** As the dataset values are treated as strings and they need preprocessing so in order to proper utilization of input they are broken into tokens to perfectly fit in our model. Example is given as followed:

Sentence: বাংলা আমার মাতৃভাষা” Tokenization: বাংলা”, আমার”, মাতৃভাষা”.

**Training:** The training process is done by the tokenized dataset. The output is differentiated from individual model by applying distinct size of window, iterations and vectors utilizing the dataset.

**Sentences:** Each sentence is broken into tokens and for CBOW we used the enormous list of documents.

**Size:** The size defines how many neurons are integrated in hidden layer for applying word embedding. Here we set the size 100.

**Window:** Here the window indicates how many words are around the target word. For implementation of word embedding the size is considered 3 that means 3 words from both side of the target word is contemplated.

**Mincount:** The mincount indicates minimum count where the infrequent words are not considerable. Here the threshold value of mincount is 1.

**Workers:** The workers can be defined as the CPU Threads that affect positively for decreasing the training time.

**Word Embedding with FastText model** Though Word2Vec word embedding technique plays a vital role in the field of NLP but it has some issues that's why new word embedding technique FastText is introduced. In Word2Vec, embedding process is generalized for each individual word, whereas there is an issue arising named out of dictionary. In such case words that are unseen in the training but related to training word fail to produce appropriate output. For example, if there is a word like কথা” and কার্ত” in the training set then it can generate output of these two words but when it comes to the word কথাবার্ত” then it cannot manage this as it is missing in the training data. To deal with the challenge FastText is introduced where improvisation is taken for the Skip-Gram.

**Generate Sub words:** To make each word identical a bracket is set to the starting and ending of that word. The word মাতৃভাষা” after adding bracket is like [মাতৃভাষা]” and then  $n$ -gram process is applied based on the size of the  $n$  until it will reach to the ending

**Table 2** Cosine distance of words

| Center word | Main context  | Negative sample |               |               |
|-------------|---------------|-----------------|---------------|---------------|
| আমার        | বাংলা         | ঢাকা            | রাজধানী       | মাতৃভাষা      |
|             | মাতৃভাষা      | মাতৃভাষা        | মাতৃভাষা      | মাতৃভাষা      |
|             | dot আমার      | dot বাংলা       | dot ঢাকা      | dot রাজধানী   |
|             | 0.7 (sigmoid) | 0.5 (sigmoid)   | 0.3 (sigmoid) | 0.4 (sigmoid) |
|             | Near to 1     |                 | Near to 0     |               |

bracket. If the value of  $n$  is 3 then the processed word is divided into following parts. মাতৃভাষা, মাতৃ, মাতৃভা, ভাষা, ভাষা, ভাষা. But this process contains immense number of  $n$  gram and for proper memory utilization hashing is applied. Negative sampling using Skip-Gram: For pretraining process we consider a sentence আমার মাতৃভাষা বাংলা where the context words are আমার and বাংলা that need to be predicted by the model and the center word is মাতৃভাষা. The contextual words আমার and বাংলা are taken without processing  $n$ -gram technique and then generate word vectors. Then randomly accumulate the negative words as sample like ঢাকা and রাজধানী. After this dot product is applied with respect the center word comparing with contextual and actual word and sigmoid function is applied to reach the outcome within 0 and 1 (Table 2).

**Word2Vec using BNL** For applying word embedding technique BNL toolkit is used for processing Bangla language. For this process tokenization is done for each sentence and then vector of each word is generated by importing BengaliWord2Vec. After pretraining process testing is done with test data that will generate 10 most similar word with the input data.

## 4 Result Analysis

The concept behind the word embedding techniques is mostly similar but has some remarkable differences that differentiate the performance. Each and every approach with same test value has generated near about satisfactory results also having individual verities which are notable. The combination of vector and window size is implemented for tuning to get most satisfactory and optimum results. For Skip-Gram and CBOW window size and vector size is 3 and 100 and for another test it is 5 and 400 respectively (Fig. 6).

Analyzing the result based on the similarity of a word the performance is quite impressive where the model is provided by the BNL. The Skip-Gram model that is based on Gensim accelerates the result by giving almost similar word regarding the context but the performance drops when the word is unseen. But when it comes to the result analysis of CBOW the performance deviates because of the noisy output that it produced. The FastText model performs well as it produces good cluster and

| Random Word | Cluster Word                             |                                     |                            |                                          |
|-------------|------------------------------------------|-------------------------------------|----------------------------|------------------------------------------|
|             | BNLP                                     | SKIP GRAM                           | CBOW                       | FastText                                 |
| মেয়ে       | কন্যা, মেয়েমানুষ, পুত্রী, পাত্রী, মানুষ | মেয়েছেলে, মহিলা বোন, সন্তান, মানুষ | নারী, মহিলা লোক, ছেলে, ভাই | মেয়েমানুষ, নারী, প্রেমিকা, মানুষ, পুরুষ |
| আমরা        | আমাদের, আমার, আমি, নিজ, তারা             | আমি, আমার, আমারতো, লোক, আমারই       | নিজেরা, আপন, আমি, দল, গর্ত | আমার, আমরাও, আপনি, সবাই, তুমি            |

Fig. 6 Outcomes of different word embedding models

Table 3 Performance analysis

| Rank | Model              | Accuracy (%) | Description                                                                                                                                                                                                |
|------|--------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Word2Vec Skip-Gram | 82           | In this process the target word is used that is mainly represented for generating to predict the context                                                                                                   |
| 2    | Word2Vec CBOW      | 78           | The approach holds the context of every word in the input and attempts to predict the target word relatively similar to the context                                                                        |
| 3    | Word2Vec BNLП      | 88           | For Bangla language processing, BNLП toolkit is used for Natural Language Processing like Bangla word embedding technique                                                                                  |
| 4    | FastText           | 89           | FastText is introduced to overcome the lacking of Word2Vec embedding as it can produce result for any unseen word in training set because of using <i>n</i> -gram technique for each token of the sentence |

from evaluating the performance Word2Vec using BNLП and Skip-Gram performs well in terms on training data but in terms of unknown data FastText provides related contextual word as it generates vectors by applying addition of *n*-gram of words. Although the result of unseen word is not satisfactory but it produces cluster rather than other models. If we used well prepared dataset then the outcome of FastText is more realistic (Table 3).

### 5 Conclusion

Bangla language is one of the most spoken languages in Indo Asian groups that consist of different varieties and one word contains different meanings and that is why context of a sentence easily varies from one another which make it more complex. For this similarity of word needs to generate to understand the context with good number of cluster data. Here we attempt different perspective of dynamic models for good numbers of Bangla word clustering based on the similarity maintaining the context. By observing the performance, we can conclude that Word2Vec in BNLП and FastText perform better but if it comes to the point of unseen data FastText has some contextual outcome where Word2Vec fails to generate any output. This work will help us in different ways like understanding the context of a sentence replacing it

using similar words, language modeling, syntax and semantics analysis of a language and so on. We can conclude that the more data is used to train more we can get most satisfactory and accurate result.

## References

1. Ritu ZS, Nowshin N, Nahid MMH, Ismail S (2018) Performance analysis of different word embedding models on Bangla language. In: International conference on Bangla speech and language processing (ICBSLP), pp 1–5
2. Thavareesan S, Mahesan S (2020) Sentiment Lexicon expansion using Word2vec and Fast-Text for sentiment prediction in Tamil texts. In: Moratuwa engineering research conference (MERCon), pp 272–276
3. Pham D-H, Le A-C (2018) Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis. *Int J Approximate Reasoning* 103:1–10
4. Introduction to word embedding and Word2Vec [Online]. Available at: [www.towardsdatascience.com](http://www.towardsdatascience.com). Accessed on: 5th July 2022
5. Mojumder P, Hasan M, Hossain F, Hasan KM (2020) A study of fastText word embedding effects in document classification in Bangla language. In: International conference on cyber security and computer science. LNICST, vol 325, pp 1–13
6. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016) FastText.zip: compressing text classification models. arXiv preprint [arXiv:1612.03651](https://arxiv.org/abs/1612.03651)
7. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
8. Lilleberg J, Zhu Y, Zhang Y (2015) Support vector machines and Word2vec for text classification with semantic features. In: IEEE 14th international conference on cognitive informatics and cognitive computing (ICCI\*CC). IEEE, pp 136–140
9. Zhang D, Xu H, Su Z, Xu Y (2015) Chinese comments sentiment classification based on Word2vec and SVMperf. *Expert Syst Appl* 42(4):1857–1863
10. Lund K, Burgess C (1996) Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, and computers* 28(2):203–208
11. Rohde DL, Gonnerman LM, Plaut DC (2006) An improved model of semantic similarity based on lexical co-occurrence. *Commun ACM* 8(627–633):116
12. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. *Trans Assoc Comput Linguist* 5:135–146
13. Gaikwad V, Haribhakta Y (2020) Adaptive glove and FastText model for hindi word embeddings. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp 175–179
14. Ismail S, Rahman MS (2014) Bangla word clustering based on *n*-gram language model. In: International conference on electrical engineering and information and communication technology. IEEE, pp 1–5
15. Mandal AK, Sen R (2014) Supervised learning methods for Bangla web document categorization. arXiv preprint [arXiv:1410.2045](https://arxiv.org/abs/1410.2045)
16. Alam MT, Islam MM (2018) Bard: Bangla article classification using a new comprehensive dataset. In: 2018 international conference on Bangla speech and language processing (ICBSLP). IEEE, pp 1–5



# Chapter 41

## Energy Consumption Optimization of Zigbee Communication: An Experimental Approach with XBee S2C Module



Rifat Zabin  and Khandaker Foysal Haque 

**Abstract** Zigbee is a short-range wireless communication standard that is based on IEEE 802.15.4 and is vastly used in both indoor and outdoor Internet of Things (IoT) applications. One of the basic constraints of Zigbee and similar wireless sensor networks (WSN) standards is limited power source as in most of the cases they are battery powered. Thus, it is very important to optimize the energy consumption to have a good network lifetime. Even though tuning the power transmission level to a lower value might make the network more energy efficient, it also hampers the network performances very badly. This work aims to optimize the energy consumption by finding the right balance and trade-off between the transmission power level and network performance through extensive experimental analysis. Packet delivery ratio (PDR) is taken into account for evaluating the network performance. This work also presents a performance analysis of both the encrypted and unencrypted Zigbee with the stated metrics in a real-world testbed, deployed in both indoor and outdoor scenarios. The major contribution of this work includes (i) to optimize the energy consumption by evaluating the most optimized transmission power level of Zigbee where the network performance is also good in terms of PDR (ii) identifying and quantizing the trade-offs of PDR, transmission power levels, current and energy consumption (iii) creating an indoor and outdoor Zigbee testbed based on commercially available Zigbee module XBee S2C to perform any sort of extensive performance analysis.

**Keywords** Zigbee · Energy consumption optimization · Zigbee testbed · Current · Energy consumption measurement · Wireless sensor network · PDR

---

R. Zabin (✉)

Chittagong University of Engineering and Technology, Chittagong, Bangladesh  
e-mail: [u1602105@student.cuet.ac.bd](mailto:u1602105@student.cuet.ac.bd)

K. F. Haque

Northeastern University, Boston, MA, USA  
e-mail: [haque.k@northeastern.edu](mailto:haque.k@northeastern.edu)

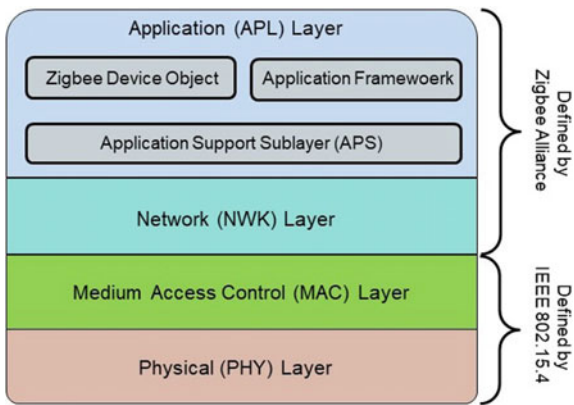
# 1 Introduction

The evolution of wireless communication has evolved IoT in the recent years. It now allows multifarious applications which definitely improves the quality of every sphere of life [1]. The wireless sensor network (WSN) consists of sensor nodes being deployed in remote locations and performs wireless communication among themselves. While considering wireless sensor network, the critical metrics might be constrained power source and hence lower transmission power level might be preferred. As a result, the strength of the signal is reduced and consequently the PDR is compromised. Our objective is to find a balance between the two concerns as in determining an optimized power level to obtain decent rate of packet delivery, i.e., good PDR with considerably lower energy consumption, as this has not been addressed by the research community yet.

Zigbee is one of the most popular low-powered wireless sensor networks which is based on IEEE 802.15.4 standards. The physical (PHY) and medium access control (MAC) layers are associated with transmission, channel selection and operating frequency, while on the contrary, the network (NWK) and application (APL) layers involve the network infrastructure, encryption and decryption of network and others [2]. Figure 1 presents the Zigbee protocol stack. The lower two layers of Zigbee PHY layer and MAC layer of Zigbee are exactly same as the IEEE 802.15.4. However, Zigbee Alliance improvises the upper three layers to introduce multi-hop and mesh network and to improve the application and usability of WSN.

Though Zigbee is one of the most energy efficient WSNs, further optimization of the energy consumption is necessary due to the constraint power supply to the Zigbee end nodes. It would also help to better understand its trade-offs with performance metrics like PDR and energy efficiency. Setting up the transceiver to highest possible transmission power level might improve the PDR, but it also hampers the energy efficiency drastically which might cause early death of nodes creating holes in the mesh network. On the other hand, tuning the transmission power level to the minimum

Fig. 1 Zigbee protocol stack



might result in drastic drop of PDR and link quality. Moreover, different applications have various priorities in terms of performance metrics. Thus, to optimize the energy consumption and network performance at the same time, a comparative study of the transmission power levels, energy consumption and corresponding PDR is necessary. For this, an experimental study with actual transceivers deployed in practical environment is much needed. Even though, Zigbee has been the center of attraction of a large research community for a decade, such experimental study has not been addressed yet. Thus, this work intends to optimize the energy consumption of Zigbee communication with analytical experiment on Xbee S2C module. The contribution of the work includes: (i) setting up the indoor and experimental test beds for measuring the PDR and energy consumption (indoor only) at different transmission power levels: 1, 3, 5 dBm (ii) to study the variation of the PDR with the change of transmission distance and power levels at both indoor and outdoor scenario (iii) optimize the energy consumption by analyzing the per packet transmission energy at different transmission power levels and determine most optimized transmission power level which would also perform with decent PDR (iv) to study the trade-offs of the transmission power level, energy consumption and PDR.

The rest part of the paper is organized as following: Sect. 2 presents the necessary background and relevant literature on the research field, Sect. 3 describes the experimental test beds. Study on PDR, current and energy consumption and optimization of the energy consumption are conducted in Sect. 4 and the work is concluded in Sect. 5.

## 2 Background and Relevant Literature

Even though there is not enough literature on this aspect based on experimental study, Pathak et al. proposed an optimization technique of Zigbee in patient monitoring by fine tuning the transmission duty cycle [3]. Wang et al. propose an energy efficient routing algorithm by switching the cluster head in a simulation-based study [4]. Li et al. propose an load balancing-based routing protocol which optimizes the energy consumption by directing the propagation of RREQ [5]. Zhang et al. propose an adaptive MAC layer-based energy saving algorithm based on both simulation and experimental study [6]. Essa et al. propose an power sensitive ad hoc on demand routing protocol by managing the operations and routes [7]. Gocal et al. propose an algorithm based on timing channels for different data priorities to optimize the energy efficiency of Zigbee [8].

Most of the previous analysis are based on software simulations and optimizing the routing protocol where the variation between indoor and outdoor configuration and optimization of transmission power has been merely discussed. Thus, this work presents a unique analysis based on performance metrics like current consumption and packet delivery ratio in both indoor and outdoor environment to optimize the energy efficiency of the Zigbee network. Zigbee node can be set up with three different modes, namely (i) coordinator, (ii) router and (iii) end node. Coordinator being

the central node of the network performs the data transfer from central node to any other nodes by allowing them to join the network. The router allows two-way communication between coordinator and end nodes, while the end nodes interacts with environment by means of sensors.

Zigbee mainly operates on three types of topology, namely (i) Tree, (ii) Star and (iii) Mesh topology as depicted by Fig. 2 [9, 10]. In tree topology, coordinators are connected to router or end node in a one to one fashion, while the star topology only consists of coordinator and end nodes in direct contacts without involving any router as inter medium. The mesh topology is the most efficient one and known to be self healing as it finds the best route for two-way data communication between coordinator and end nodes by means of mesh arrangement of routers. However, our study focuses only on the end nodes as the router and coordinator nodes always remain awake and are assumed to un-constrained power supply. Zigbee indoor application involves home automation, smart surveillance, energy management and others [11]. All of them face hindrance because of indoor obstacles due to its Non-Line-of-Sight (NLoS) arrangement. Radio frequency interference affects the performance severely as the operating frequency of Zigbee overlaps the identical operating frequencies of Wi-Fi, Bluetooth, Z-wave and create cross technology interference to drastically deteriorate the performance of Zigbee network [12, 13]. It is evident from the literature that a real life experiment with testbed deployment would provide much better estimation of the performance parameters like PDR and energy consumption. Thus, this work is based on both indoor and outdoor testbed deployment with real data measurements which takes into account of the practical world interference and environment.

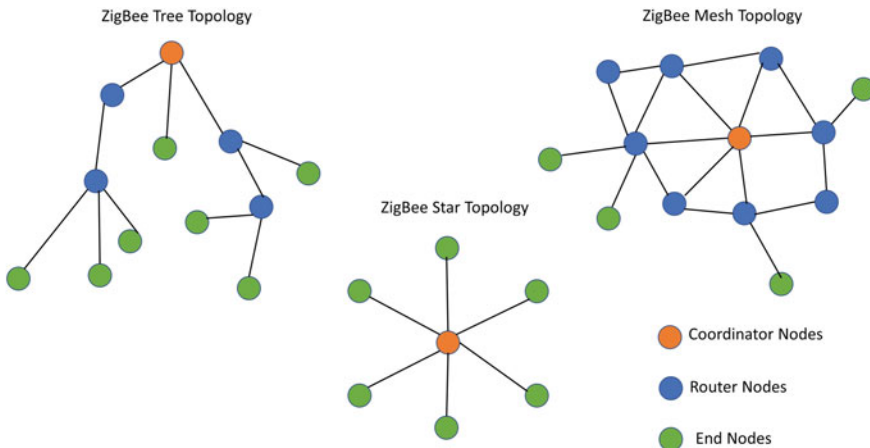


**Fig. 2** Different topology of Zigbee protocol

### 3 Experimental Setup

The hardware setup has been carried out for both indoor and outdoor experiment. The Xbee S2C module has been used at 3 different Tx power levels—1, 3 and 5 dBm with an operating frequency of 2.4 GHz. The outdoor setup is conducted in an open space by placing the coordinator node at one end and the end nodes at a distance ranging 0–40 m from the coordinator. A big parking lot is taken into consideration for outdoor testbed allowing minimum noise and interference. For indoor testing, the lab space is taken into consideration as indicated in Fig. 3. The placements of the end nodes are annotated by  $E_0$ ,  $E_{10}$ ,  $E_{20}$ ,  $E_{30}$ ,  $E_{40}$ ,  $E_{50}$  which are, respectively, at a distance of 0, 10, 20, 30, 40 and 50 m distance from the coordinator which is placed at  $C_0$ . The end nodes are placed at level one and at random location with NLoS. It is also to mention that the whole building is under Wi-Fi coverage with moderate amount of Bluetooth devices in use which would surely affect the Zigbee performances in comparison with the outdoor scenario [1].

Different PDR measurements are taken by placing the end nodes at various locations indicated by Fig. 3. This would facilitate to understand the trend of PDR in terms of transmission distance and transmission power levels. On the other hand, to analyze the energy consumption better, it is important to understand how the communication is performed, thus a data packet is transferred from a coordinator to an end node or vice versa. Zigbee protocol is based on IEEE 802.15.4 and Zigbee follows this standard completely for medium access control (MAC) and physical (PHY) layer. However, it is modified and different than that of IEEE 802.15.4 in the network (NWK) and application (APS) layer where it allows the Zigbee protocol to form a mesh network and enable multi-hop communication from the end node to the coordinator [2]. However, the energy consumption of a sensor network heavily depends on PHY and MAC layer which are the same as the IEEE 802.15.4. Zigbee

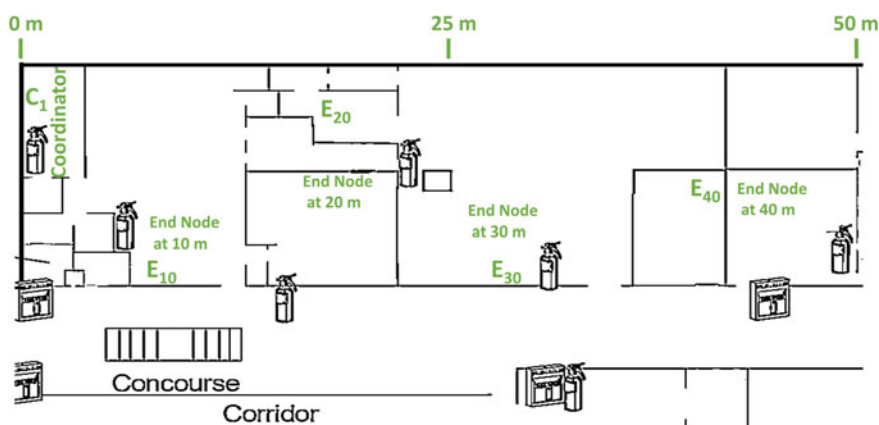


Fig. 3 Experimental testbed in the indoor lab environment

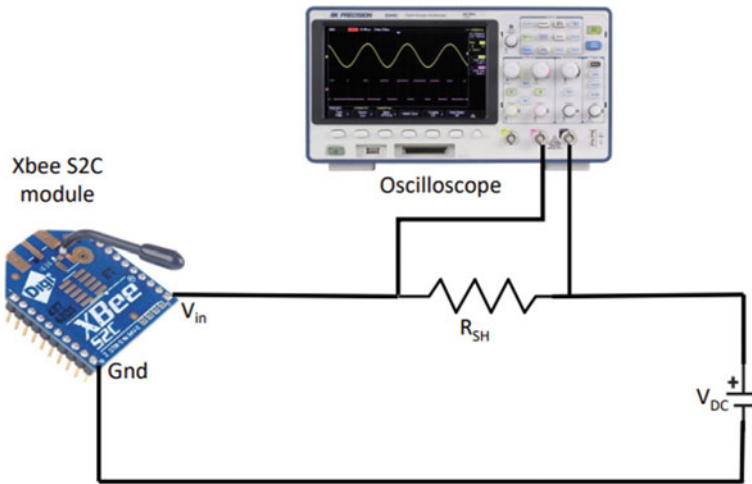


**Fig. 4** Real-time current consumption measurement setup

follows the CSMA/CA to send data from one node to another. In this experiment, data packets of 30 bytes are sent from the coordinator to an end node. For measuring the current consumption, the end node is powered up from a 3.5 DC supply and connected to an oscilloscope across a shunt resistor of  $9\ \Omega$ . This setup for real-time current consumption measurement during data transmission is presented by Fig. 4.

The current consumption measurement is done for various  $P_{Trans}$  level from which the power and energy consumption for a successful transmission of a packet can also be derived.

## 4 Energy Consumption Optimization by Analyzing ( $P_{Trans}$ ) Levels with Corresponding PDR

Different Tx power levels are considered in this experimental study with a view to comparing and determining the optimized energy consumption with a fair enough PDR. We will now look in details on this analysis procedure.

### 4.1 Variation of Current and Energy Consumption with Various $P_{Trans}$ Levels

The coordinator and the end nodes exactly follow the principle of IEEE 802.15.4 protocol in PHY and MAC layer. A current consumption capture is presented in Fig. 5 which would help in better realizing the different steps of a packet transmission and

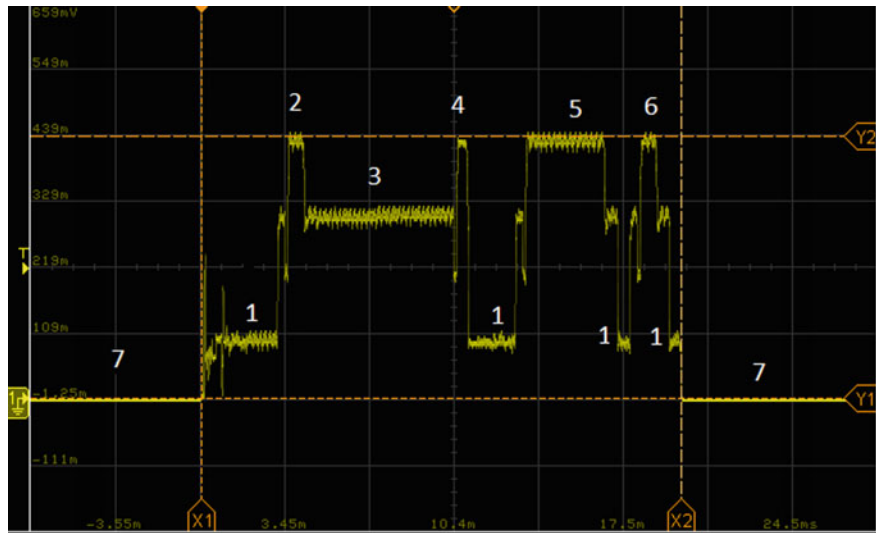


Fig. 5 Real-time current consumption capture of a successful packet reception

corresponding current and energy consumption. According to the CSMA/CA, the sender node (end node/coordinator) initiates the communication which is coordinator in this case study.

The coordinator broadcasts a beacon message of length 30 bytes in its network [14]. The end nodes are configured in cyclic sleep mode. They wake up after the predefined sleep time and listen for the beacon message from the sender which is the coordinator in this case. To wake up from sleep and receive the broadcasted beacon from the coordinator, the receiver takes some time. During this wake-up time, the receiver stays in radio idle mode as presented by annotation 1, whereas annotation 2 denotes the reception of this beacon message. After that, the receiver sends a data request to the coordinator. After receiving a broadcast beacon, the receiver stays in radio standby mode as annotation 3 until it sends the data request to the coordinator as presented by annotation 4. Then the receiver radio goes to idle mode and stays in idle mode until it receives the acknowledgment of the data request. After receiving the data request from the receiver, the sender (coordinator) waits for a backoff time according to the predefined contention window and then performs the clear channel assessment (CCA) before sending the data. The receiver goes to the receiving mode after the reception of this acknowledgment from the coordinator and remains in that mode until the data transfer is over as denoted by annotation 5. It processes the received data and upon successful reception of the data it sends an acknowledgment back to the coordinator which is denoted by annotation 6 and then goes to sleep mode again. These steps keep repeating for every packet transmission and all these stages of packet transmission are summarized in Table 1.

**Table 1** Different stages of data reception corresponding to Fig. 5

| Annotation of Fig. 5 | Stages of data transmission | Brief explanation                                                                                                                      |
|----------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 1                    | Idle time (tidle)           | Node is active, but radio is not active. Node tends to stay in idle mode to save energy                                                |
| 2                    | Data reception time Trx     | Reception of the beacon message broadcasted from a coordinator                                                                         |
| 3                    | Radio standby time Tsb      | Radio stays in standby mode before sending a data request to a coordinator or any sender as it waits for backoff time and performs CCA |
| 4                    | Data transmit time Ttx      | End node sends the data request to the coordinator/sender                                                                              |
| 5                    | Data reception time Trx     | End node receives the ACK of the data request and goes to receiving mode and waits until data transmission is over                     |
| 6                    | Data transmit time Ttx      | ACK is sent upon successful reception of the packet                                                                                    |
| 7                    | Sleep time Tsleep           | End node remains in the sleep mode before and after the reception of the data packet as defined by the experiment configuration        |

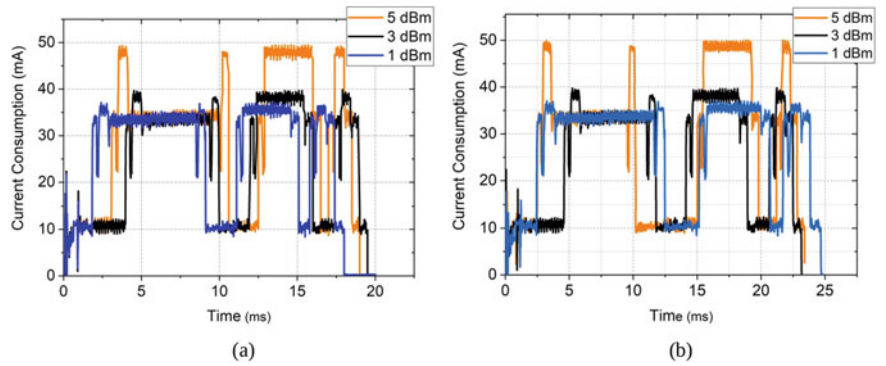


**Fig. 6** Current consumption of a single packet **a** without and **b** with AES encryption

The current consumption captures of the packet reception at different  $P_{\text{trans}}$  levels with both 128-bit AES encryption and without encryption are presented in Fig. 6.

It is to be mentioned that the end node can receive the broadcast beacon from the coordinator at any time during its wake up duration. Thus, the duration of the first idle time (before receiving the beacon broadcast) as depicted by annotation 1 (first one before annotation 2) of Fig. 5 may vary randomly. To normalize this effect for each of the instances, a mean of 20 readings is taken into consideration.



The figure shows that the current consumption during transmitting, and reception of a packet is highest for  $P_{\text{trans}}$  level of 5 dBm which is followed by 3 dBm, and the current consumption with a transmission power of 1 dBm is the lowest. Moreover, it is evident from Fig. 6 that, the difference of the current consumption with different transmission power levels mostly differs in transmit and reception mode. The current consumption during radio standby time, idle time and sleep time is almost the same for all the  $P_{\text{trans}}$  levels. Moreover, with AES encryption, the current consumption during transmit and reception is slightly higher than that of unencrypted data due to the higher time and resource requirement of the encryption procedure. The exact current and energy consumption during each stage of the data reception for all the  $P_{\text{trans}}$  levels with both unencrypted and encrypted communication are depicted in Tables 2 and 3, respectively. The mean of the 20 readings is taken into consideration for the data collection. The energy consumption for each stage of the transmission is calculated with Eq. (1) where  $I$  is the average current consumption,  $R_{sh}$  is the shunt resistor of  $9\ \Omega$  across which the XBee module is connected, and  $T$  is the duration of that stage.

$$\text{Energy Consumption} = I^2 R_{sh} T \quad (1)$$

From Tables 2 and 3, we can see that for both the encrypted and unencrypted communication, the energy consumption is highest with the transmission power level of 5 dBm and lowest with 1 dBm. As encrypted communication performs functions like encrypting the data before transmission and decrypting the data after reception, it consumes more energy for all the transmission levels in comparison with unencrypted communication. However, provided the security the AES encryption provides to the communication, this increased energy consumption for the encryption is considerable and is recommended for most indoor and outdoor IoT applications.

## 4.2 PDR Performances at Different $P_{\text{Trans}}$

The packet delivery ratio can be represented by Eq. (2) which signifies what percentage of the sent data packet is received at the other end, a vital metric in WSN. If the PDR is bad, the network is prone to losing important data and it also increases the number of re-transmissions which eventually increases the power consumption, network traffic and data overhead. For the indoor scenario, the PDR is measured by placing the coordinator at C1 and end nodes at E25, E30, E35 and E40, at a distance of 25 m, 30 m, 35 m and 40 m from coordinator, respectively, as presented in Fig. 4. 1000 data packets are sent from the coordinator to the end node at each location and from the number of received packets, and PDR has been calculated. To equalize the effect of interference of other 2.4 GHz wireless technologies, three different tests at a different time of the day have been carried out with the same setup and the mean of these three tests is taken into consideration. The transmission power is kept unchanged throughout the experiment and the whole process is repeated for each

**Table 2** Current and energy consumption during each stage with unencrypted communication

| Annotation of<br>Fig. 6       | Stages of data<br>transmis-<br>sion/reception | $P_{trans}$   |                                           |                               |               |                                           |                               |               |                                           |                               |        |
|-------------------------------|-----------------------------------------------|---------------|-------------------------------------------|-------------------------------|---------------|-------------------------------------------|-------------------------------|---------------|-------------------------------------------|-------------------------------|--------|
|                               |                                               | 1 dBm         |                                           |                               | 3 dBm         |                                           |                               | 5 dBm         |                                           |                               |        |
|                               |                                               | Duration (ms) | Average<br>current<br>consumption<br>(mA) | Energy<br>consumption<br>(mJ) | Duration (ms) | Average<br>current<br>consumption<br>(mA) | Energy<br>consumption<br>(mJ) | Duration (ms) | Average<br>current<br>consumption<br>(mA) | Energy<br>consumption<br>(mJ) |        |
| 1                             | Idle time tidle                               | 6.1           | 10.5                                      | 6.05                          | 8.4           | 10.5                                      | 8.33                          | 6.8           | 10.5                                      | 6.74                          |        |
| 2                             | Data reception<br>time trx1                   | 0.9           | 36.0                                      | 10.49                         | 0.9           | 38.2                                      | 11.81                         | 0.9           | 47.9                                      | 18.58                         |        |
| 3                             | Radio standby<br>time Tsb                     | 6.0           | 33.8                                      | 61.69                         | 5.5           | 33.7                                      | 56.21                         | 5.5           | 33.9                                      | 56.88                         |        |
| 4                             | Data transmit<br>time Ttx1                    | 0.7           | 35.1                                      | 7.76                          | 0.7           | 36.0                                      | 8.16                          | 0.7           | 47.5                                      | 14.21                         |        |
| 5                             | Data reception<br>time trx2                   | 4.0           | 35.0                                      | 44.10                         | 4.0           | 36.0                                      | 46.65                         | 4.8           | 46.0                                      | 91.41                         |        |
| 6                             | Data transmit<br>time Ttx2                    | 2.1           | 33.9                                      | 22.23                         | 2.1           | 36.0                                      | 25.07                         | 2.1           | 40.0                                      | 30.24                         |        |
| Total energy consumption (mJ) |                                               |               |                                           |                               | 152.32        |                                           |                               | 156.23        |                                           |                               | 218.06 |

**Table 3** Current and energy consumption during each stage with encrypted communication

| Annotation of<br>Fig. 6       | Stages of data<br>transmis-<br>sion/reception | $P_{trans}$   |                                           |                               |               |                                           |                               |               |                                           |                               |                               |
|-------------------------------|-----------------------------------------------|---------------|-------------------------------------------|-------------------------------|---------------|-------------------------------------------|-------------------------------|---------------|-------------------------------------------|-------------------------------|-------------------------------|
|                               |                                               | 1 dBm         |                                           |                               | 3 dBm         |                                           |                               | 5 dBm         |                                           |                               | Energy<br>consumption<br>(mJ) |
|                               |                                               | Duration (ms) | Average<br>current<br>consumption<br>(mA) | Energy<br>consumption<br>(mJ) | Duration (ms) | Average<br>current<br>consumption<br>(mA) | Energy<br>consumption<br>(mJ) | Duration (ms) | Average<br>current<br>consumption<br>(mA) | Energy<br>consumption<br>(mJ) |                               |
| 1                             | Idle time tidle                               | 7.3           | 10.5                                      | 7.29                          | 10.0          | 10.75                                     | 10.4                          | 10.75         | 10.75                                     | 11.18                         |                               |
| 2                             | Data reception<br>time trx1                   | 1             | 36.0                                      | 11.66                         | 1.05          | 38.8                                      | 14.22                         | 1.0           | 49.0                                      | 21.60                         |                               |
| 3                             | Radio standby<br>time Tsb                     | 5.4           | 34.0                                      | 56.18                         | 5.2           | 34.0                                      | 54.1                          | 6.25          | 34.0                                      | 65.02                         |                               |
| 4                             | Data transmit<br>time Ttx1                    | 0.8           | 35.3                                      | 8.97                          | 0.8           | 38.4                                      | 10.61                         | 0.8           | 49.8                                      | 17.85                         |                               |
| 5                             | Data reception<br>time trx2                   | 6.0           | 35.0                                      | 66.15                         | 5.0           | 37.5                                      | 63.28                         | 4.8           | 47.5                                      | 97.47                         |                               |
| 6                             | Data transmit<br>time Ttx2                    | 2.5           | 34.5                                      | 26.78                         | 2.3           | 36.0                                      | 26.82                         | 2.3           | 41.0                                      | 34.79                         |                               |
| Total energy consumption (mJ) |                                               |               |                                           | 177.03                        |               |                                           | 179.43                        |               |                                           | 247.91                        |                               |

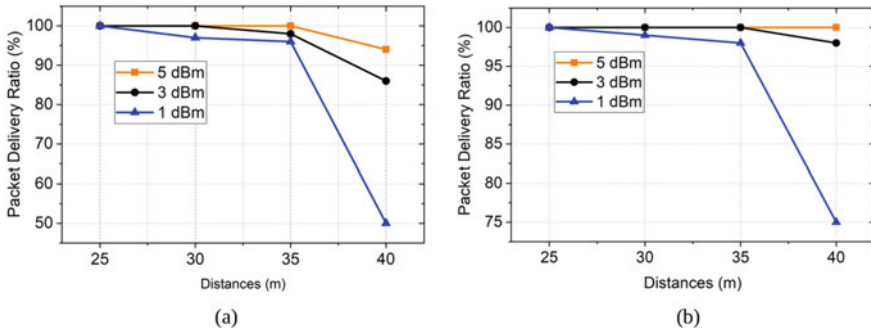


**Fig. 7** PDR versus distances with three different transmission power levels for **a** indoor and **b** outdoor scenario

of the transmission power 1, 3 and 5 dBm. For the test in the outdoor scenario, the coordinator node is placed in an open parking lot and end nodes around it with direct LoS where the interference is also minimum. Figure 7 presents the variation of PDR at different distances with three different transmission power.

$$\text{PDR} = \frac{\text{Total Packets Received}}{\text{Total Packets Sent}} \quad (2)$$

With 1 dBm, at 30 m of transmission distance, the PDR decreased to 96% and 97%, respectively, for indoor and outdoor scenarios, which are still quite reliable. The PDR with 3 dBm starts decreasing after 30 m and reaches to 98% at 35 m with indoor scenario, whereas it still performs with 100% PDR in outdoor. So, up to a transmission distance of 35 m, transmission with all the  $P_{\text{Trans}}$  performs reliably. However, the PDR performances degrade drastically at 40 m in the indoor scenario for the transmissions of all the power levels which are 94%, 86% and 50% for 5 dBm, 3 dBm and 1 dBm, respectively. For the outdoor scenario, the PDR performances did not degrade so drastically: 96% and 75% with  $P_{\text{Trans}}$  of 3 dBm and 1 dBm, respectively, whereas the transmission with 5 dBm still performs with 100% PDR at a transmission distance of 40 m even. It is evident from the analysis that, the decrease of the PDR at 3 dBm is not as drastic as at 1 dBm. The transmission link with 1 dBm becomes quite unreliable at 40 m for both indoor and outdoor scenarios, whereas for 5 dBm and 3 dBm, it remains reasonably reliable with decent PDR in indoor and performs with even better PDR in outdoor. To optimize the energy consumption finding the optimized  $P_{\text{Trans}}$  level with decent PDR and energy consumption is necessary through comparative analysis. This would also help to realize the trade-offs between  $P_{\text{trans}}$ , PDR and energy consumption and to find an optimized  $P_{\text{Trans}}$  level where it can perform both reliably and energy efficiently.

### 4.3 Evaluation of $P_{\text{Trans}}$ Levels for Energy Optimization

One of the interesting facts of both encrypted and unencrypted communication is that the difference of the total energy consumption between 1 dBm and 3 dBm  $P_{\text{Trans}}$  level is very little in comparison with the energy consumption difference between 3 dBm and 5 dBm. In fact, with encrypted communication, this difference of energy consumption between  $P_{\text{Trans}}$  levels of 1 dBm and 3 dBm is almost inconsiderable. On the contrary, if PDR for different  $P_{\text{Trans}}$  levels are analyzed, it is noticeable that PDR improved significantly from 1 dBm to 3 dBm. As with 1 dBm, at 40 m, the PDR drops down drastically to 75% and 50%, respectively, for outdoor and indoor environments, it is no longer reliable and beyond consideration. With 3 dBm, the PDR is 100% up to a transmission distance of 35 m and 30 m, respectively, for outdoor and indoor environments. And this drops down to 96% and 86% for outdoor and indoor environments, respectively, at a transmission distance of 40 m which is decent for most IoT applications. As with very little increase of energy consumption from 1 dBm to 3 dBm, the PDR improves radically, at 3 dBm, transmission power level optimized energy consumption and performance are achieved. To improve the PDR further with a 5 dBm  $P_{\text{Trans}}$  level, the energy consumption increases significantly which will decrease the overall network life dramatically. Thus, considering the trade-off between PDR and energy efficiency,  $P_{\text{Trans}}$  level of 3 dBm achieves optimized energy consumption with decent transmission performance. The further detail works are presented in [15].

## 5 Conclusion

In this work, a case study is performed based on the real-world indoor and outdoor testbeds and measurements where the XBee S2C module is used as the transceiver. Aim of this work is to optimize the energy consumption of Zigbee communication with a comparative analysis of current, energy and power consumption with PDR at two different scenario: indoor and outdoor. The performance evaluation of PDR, current and energy consumption is also performed based on the practical data collected from the deployed indoor and outdoor scenario. This work draws out the trade-offs and performance limitations of these metrics which would benefit both academia and industry for designing and deploying Zigbee in both indoor and outdoor applications. The study shows the transmission power level of 3 dBm is the most optimized one in terms of PDR and energy consumption, i.e., the node lifetime. It is to be noted that the experiment that is conducted in this study is based on the XBee S2C module. However, the results can be generalized as most of the commercially available modules share the similar performance trend.

## References

1. Haque KF, Abdelgawad A, Yanambaka VP, Yelamarthi K (2020) LoRa architecture for V2X communication: an experimental evaluation with vehicles on the move. *Sensors* 20(23):6876
2. Ergen SC (2004) Zigbee/IEEE 802.15.4 summary. *UC Berkeley* 10(17):11
3. Pathak S, Kumar M, Mohan A, Kumar B (2015) Energy optimization of Zigbee based WBAN for patient monitoring. *Proc Comput Sci* 70:414–420
4. Wang Q, Wang D, Qi X (2019) An energy-efficient routing protocol for Zigbee networks. In: *IOP conference series: earth and environmental science*, vol 295. IOP Publishing, p 052040
5. Li B, Li Q, Wang Y, Shentu N (2019) Zigbee energy optimized routing algorithm based on load balancing. In: *AIP conference proceedings*, vol 2122. AIP Publishing LLC, p 020057
6. Zhang Y, Yang K, Chen H (2019) An adaptive MAC layer energy-saving algorithm for Zigbee-enabled IoT networks. In: *International conference on smart city and informatization*. Springer, pp 365–378
7. Essa EI, Asker MA, Sedeeq FT (2020) Investigation and performance optimization of mesh networking in Zigbee. *Period Eng Nat Sci* 8(2):790–801
8. Gočál P, Macko D (2019) EEMIP: energy-efficient communication using timing channels and prioritization in Zigbee. *Sensors* 19(10):2246
9. Varghese SG, Kurian CP, George V, John A, Nayak V, Upadhyay A (2019) Comparative study of Zigbee topologies for IoT-based lighting automation. *IET Wirel Sens Syst* 9(4):201–207
10. Moridi MA, Kawamura Y, Sharifzadeh M, Chanda EK, Wagner M, Okawa H (2018) Performance analysis of Zigbee network topologies for underground space monitoring and communication systems. *Tunn Undergr Space Technol* 71:201–209
11. Ali AI, Partal SZ, Kepke S, Partal HP (2019) Zigbee and LoRa based wireless sensors for smart environment and IoT applications. In: *2019 1st global power, energy and communication conference (GPECOM)*. IEEE, pp 19–23
12. Chen G, Dong W, Zhao Z, Gu T (2018) Accurate corruption estimation in Zigbee under cross-technology interference. *IEEE Trans Mob Comput* 18(10):2243–2256
13. Chen Y, Li M, Chen P, Xia S (2019) Survey of cross-technology communication for IoT heterogeneous devices. *IET Commun* 13(12):1709–1720
14. Tseng HW, Pang AC, Chen J, Kuo CF (2009) An adaptive contention control strategy for IEEE 802.15.4-based wireless sensor networks. *IEEE Trans Veh Technol* 58(9):5164–5173
15. Haque KF, Abdelgawad A, Yelamarthi K (2022) Comprehensive performance analysis of Zigbee communication: an experimental approach with XBee S2C module. *Sensors* 22(9):3245

## Chapter 42

# PredXGBR: A Machine Learning Based Short-Term Electrical Load Forecasting Architecture



Rifat Zabin , Labanya Barua, and Tofael Ahmed

**Abstract** The increase of consumer end load demand is leading to a path to the smart handling of power sector utility. In recent era, the civilization has reached to such a pinnacle of technology that there is no scope of energy wastage. Consequently, questions arise on power generation sector. To prevent both electricity shortage and wastage, electrical load forecasting becomes the most convenient way out. Artificial Intelligent, Conventional and Probabilistic methods are employed in load forecasting. However the conventional and probabilistic methods are less adaptive to the acute, micro and unusual change of the demand trend. With the recent development of Artificial intelligence, machine learning has become the most popular choice due to its higher accuracy based on time, demand and trend based feature extractions. Even though machine learning based models have got the potential, most of the contemporary research works lack in precise and factual feature extractions which results in lower accuracy and higher convergence time. Thus the proposed model takes into account the extensive features derived from both long and short time lag based auto-correlation. Also, for an accurate prediction from these extracted features two Extreme Gradient Boosting (XGBoost) Regression based models: (i) PredXGBR-1 and (ii) PredXGBR-2 have been proposed with definite short time lag feature to predict hourly load demand. The proposed model is validated with five different historical data record of various zonal area over a twenty years of-2 time span. The average accuracy ( $R^2$ ) of PredXGBR-1 and PredXGBR-2 are 61.721% and 99.0982% with an average MAPE (error) of 8.095% and 0.9101% respectively.

---

R. Zabin (✉) · L. Barua

Chittagong University of Engineering and Technology, Chittagong, Bangladesh  
e-mail: [u1602105@student.cuet.ac.bd](mailto:u1602105@student.cuet.ac.bd)

L. Barua

e-mail: [u1602106@student.cuet.ac.bd](mailto:u1602106@student.cuet.ac.bd)

T. Ahmed

Department of EEE, Chittagong University of Engineering and Technology, Chittagong, Bangladesh  
e-mail: [tofael@cuet.ac.bd](mailto:tofael@cuet.ac.bd)

**Keywords** Electrical load forecasting · Load prediction · XGBoost · Definite time lag · Regression

## 1 Introduction

Electricity generation according to demand has always been a matter of great concern in the power sector of a country. Generation must ensure the fulfillment of industrial and domestic demand of all over the particular region at the same time restrain the excess generation to prevent power wastage. Development of technology and necessity of green energy utilization have discovered many probability and prospects in the power sector. PV system, wind energy and other renewable sources are being utilized in building up decentralized stand-alone grid stations [1]. These progresses are in vein if system loss is not reduced significantly. Hence naturally, demand prediction draws attention of researchers. Load prediction is not a whole new concept. It has been implemented for a long time in the grid network. Both qualitative and quantitative methods including curve fitting, decomposition, regression, exponential smoothing etc have been studied and applied conventionally over the time. Eventually the statistical techniques turned into probabilistic and heuristic forms involving Auto Regression (AR), Auto Regression Moving Average (ARMA), Auto Regression Integrated Moving Average (ARIMA) model, Support Vector Machine (SVM) and other computer algorithms [2]. All these probabilistic algorithms stated complex multi-variable mathematical models for solution. The more networks added to the central grid, the more non-deterministic polynomial (NP-hard) problems arises which increases complexity. Further studies approaches to the reduction of the complexity and introduces data driven neural networks. Historical data from past two months to two years built forecasting model and reduced complexity until the data-sets increased enormously with the period of time [3].

In the last decade, machine learning approach has reached to the apex in time series prediction technique. Machine Learning is a branch of Artificial Intelligence. It involves the process of accessing some data sets and learn from them in a way similar to which, human brain learns [4]. Machine learning begins with the processing of data set, learning data, extracting feature and finally gain knowledge. The method has been accepted worldwide because of its computational speed, error-less calculation, and feature adaption. Artificial Intelligence is the main objective of ML [3]. In the passage of time, machine learning has been deployed within many practical applications. Image recognition, hand-writing and language recognition, home automation, IOT based smart waste collection presented great application of machine learning concept [5]. Load forecasting involves labelled data sets for training and hence classified into supervised learning type. Deep learning approaches later served the purpose involving depth of layers. Not only these methods simplified the total process of load prediction, but also speed up the operation and establish robustness of architecture.

In this paper we propose an approach of Extreme Gradient Boosting (XGBoost) regression technique to develop a robust electric load forecasting model with great



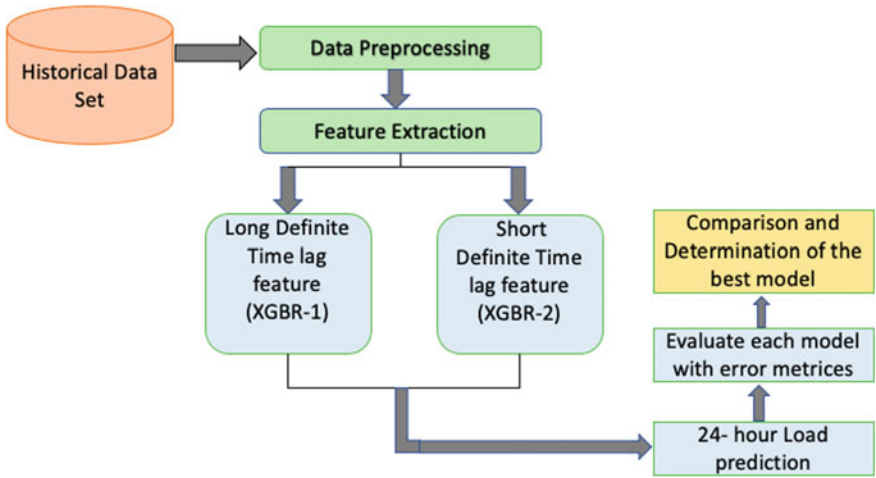


**Fig. 1** Load forecasting mechanism

accuracy. Our goal is to design feature based XGBoost models and to evaluate and cross validate them with 5 different data sets. Make a comparative study on how the different features effect the model performance (Fig. 1).

The architecture of this paper shows in Sect. 2, works relevant to load prediction using machine learning hence, the main contribution of this study, Sect. 3 presents an elaborated framework on the model, The following sections presents the data preparation, feature extraction and result analysis.

## 2 Contemporary Research and Authors Contributions

Different approaches of Short-Term Load Forecasting (STLF) have been evidently considered lately as the most effective method for electric load prediction. Many machine learning and deep learning models are derived throughout the passage of time. They have ease the efficient management, economic dispatch and scheduling of generated electrical load [6].

Artificial Neural Network techniques are come out to be most reasonable method of load forecasting. H. Aly proposed six clustering hybrid models [7] consisting of Artificial Neural Network (ANN), Wavelet and Neural Network (WNN) and Kalman Filtering (KF) combination. A regional load forecasting was studied by Singh [8] of NEPOOL region of ISO New England where hourly temperature, humidity and historical electric load data were taken into account. Marino [9] in one of his studies investigated on conventional LSTM and Sequence to Sequence architecture (S2S) base on LSTM for individual building level forecasting. Ageng [10] proposed an hourly load forecasting structure for domestic household merging LSTM and data

preparation strategies. A hybrid model was proposed (2021) by Bashir and Haoyong [11] namely Back Propagation Neural network (BPNN) consisting of the combination of Prophet and LSTM. DPSO-LSTM approach was proposed using Discrete Particle Swarm Optimization (DPSO) algorithm by Yang [12]. K. Amarasinghe studied on (2017) classical CNN and bench-marked it against result obtained from other ideal models like LSTM (S2S) [13]. Alhussein [14] proposed a hybrid model namely CNN-LSTM where CNN layers used for feature extraction with LSTM layers for sequence learning.

XGBoost is a recent addition to regression model. Wang [15] applied line regression for trend series and Extreme Gradient Boosting (XGBoost) for fluctuating sub-series data decomposed my VMD and SVMD method. Zheng [16] where a hybrid model was built up involving Similar Day (SD), Empirical Mode Decomposition (EMD) and LSTM combination.

However, most of the contemporary research works related to electric load forecasting are associated with LSTM, RNN, CNN and other statistical algorithms like ARIMA, SVM and so on. To author's knowledge, none of them executed short term definite time lag features and so the characteristics of the trained data record might be random and very much non-linear.

On this aspect our main contributions are:

1. Designed feature based XGBoost models and evaluated as well as cross validated them with five different data sets.
2. Performed a comprehensive comparative study on how different features might impact the model performances.
3. Introduced time lag features which improves short term load prediction provided previous 24 h data available.
4. The proposed model performed with an error rate of 1.05% on an average on all the dataset.
5. The operational code has been made publicly available for the ease of further research work.

### 3 Model Design

Electric load data are unbalanced, non linear and difficult to build up relationships. As stated before, conventional statistical models are insufficient in case of forecasting these type of historical data. XGBoost regression allows a scalable tree boosting algorithm. Our work is mainly associated with this model. The impact of the model drew attention in the kaggle ML competition where most of the winning projects were associated with XGBoost regression and classifier model [17].

XGBoost employs CART (Classification and Regression Tree) to be modified from the residual of each iteration. The principle of CART is a generalized binomial variable called GINI Index. CART facilitate the procedure by allowing splits utiliza-

tion on the aspect of missing values [18]. Let  $N$  be the number of CART, the score by  $i$ th sample represented by  $f_k(x_i)$ .

$$\hat{y}_{xi} = \sum_{k=1}^N f_k(x_i), \quad f_k \in \zeta \quad (1)$$

where,  $\hat{y}_{xi}$  stands for the final output function,  $\zeta = (f_x = w_{q(x)})$  and  $q$  represents structure of each tree. Regression tree provides a continuum score associated with each leaf. This is considered as one of the basic difference between decision and regression tree [17].

By summing up, the minimized objective function hence obtained,

$$\mathcal{L}\{\phi\} = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2)$$

where,

$$\Omega(f) = \gamma T + \frac{1}{2} \alpha \|\omega\|^2 \quad (3)$$

In the above Eqs. (2) and (3), the term ' $l$ ' belongs to loss function that determine the difference between the prediction  $\hat{y}_i$  and the actual data  $y_i$ . The error hence obtained is penalized by the function  $\Omega(f_k)$ . This procedure ultimately smooth the weight of the prediction curve and therefore quality is enhanced by eliminating over-fitting.

The formal objective function we obtain,

$$\tilde{\mathcal{L}}^{(t)} = \sum_{j=1}^T \left[ \left( \sum_{i \in I_j} g_i \right) \omega_j + \sum_{i \in I_j} (h_i + \alpha) \omega_j^2 \right] + \gamma T \quad (4)$$

$I_j$  stands for all leaf nodes  $j$ ,  $g_i$  and  $h_i$  represent the first and second order derivatives. In the equation,  $\omega_j^*$  is defined as the weight function of  $j$  leaf

$$\omega_j^* = -\frac{\sum_i g_i}{\sum_i h_i + \alpha}, \quad \text{where } i \in I_j \quad (5)$$

The quality of the regression tree can obtained by the following equation. This is determined over a wide range of objective function, starting from a single leaf and therefore adding branches cumulatively using a greedy algorithm.

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^T \frac{\left( \sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \alpha} + \gamma T \quad (6)$$

Finally, Eq. (7) presents the practical formula for evaluation of split candidates.

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left[ \frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \alpha} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \alpha} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \alpha} \right] - \gamma \quad (7)$$

This study is based on XGBoost regression tree algorithm which we make sure to perform extra-ordinarily on time series type forecast. We have two models, one of them works on long term features defined by PredXGBR-1 and the second and the best one works on short term features defined by PredXGBR-2. These characteristics are later discussed in the following sections. However, The two XGBoost models continue iteration until there is presence of residual. It stops at least after 200 runs as soon as the residual goes to null. The model performs a full iteration for given dataset and thus the parameters are tuned for the subset. The learning process is similar to transfer learning of old parameters. The active function seasonal decomposition is employed here to avoid over-fitting in PredXGBR-1. Each dataset is trained and tested in a particular ratio which is described in later sections.

## 4 Data Preparation and Feature Extraction

**Data selection and preparation** Our proposed model has been validated and verified with bunch of dataset. The list of data we employed:

A regional transmission organization which supervise the distribution of wholesale electricity all over the states of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia and the District of Columbia. Long term planning of broad, reliable, efficient and cost-effective interstate electricity wholesale market to ensure coverage for 65 million people. From the historical record, electric load demand from 1998 to 2002 has been extracted. PJM east (PJME) data covers electric load demand within the timeline of 2002–2018 in the eastern region of USA. Likewise PJM West (PJMWest) serves the load demand data from 2002 to 2018 over the western region of USA. AEP is known as one of the great investors supplying electricity to almost 11 states over USA. AEP activities involve making strategies and planning through engineering, construction, handling raw materials and renewable energy conversion. Owing almost 38,000 MW generation capacity and over 750 kV ultra HV lines, AEP coverage considered as the largest electrical generation company. A complete historical data record between 2004 to 2018 has been employed for the purpose of this study. The DAYTON, Ohio (DP&L) power plant is mostly coal-fired electricity generation plants placed in Ohio, meeting the power demand over the state. The plant provides a complete demand record within the timeline of 2004–2018 before it had been reached in an agreement with AES Corp, a global power company in Arlington (Table 1).

The earlier stated data set are the key to our proposed model validation. The records are pre-processed and then trained and tested at a ratio of 80/20 percentage. Figure 2 can clarify the process.

**Table 1** Dataset scheduling

| Dataset  | Starting date | Ending date | Split ratio | Split date |
|----------|---------------|-------------|-------------|------------|
| PJM      | 31-12-1998    | 02-01-2001  | 80%         | 07-08-2000 |
| PJM East | 31-12-2002    | 02-01-2018  |             | 02-01-2015 |
| PJM West | 31-12-2002    | 02-01-2018  |             | 02-01-2015 |
| AEP      | 31-12-2004    | 02-01-2018  |             | 28-05-2015 |
| DAYTON   | 31-12-2004    | 02-01-2018  |             | 28-05-2015 |

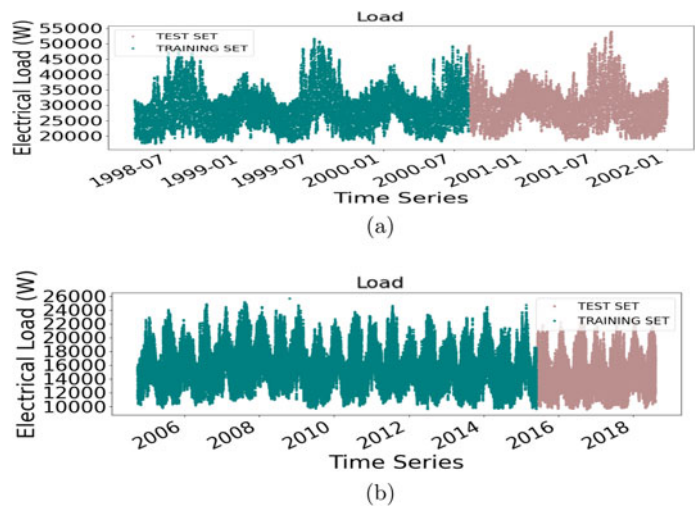


**Fig. 2** Split up the dataset by pre-processing

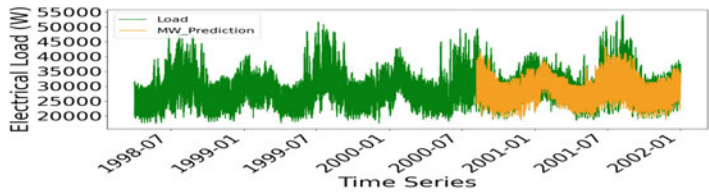
**Feature Extraction** Both of the models are enhanced with unique features. It can be described as below:

**Long Time Lag Feature** Moving forward to training the model, we initially compose a long term lag feature which can be noted as conventional one. The year, month, day, month of the year, week of the year are the key properties of this feature.

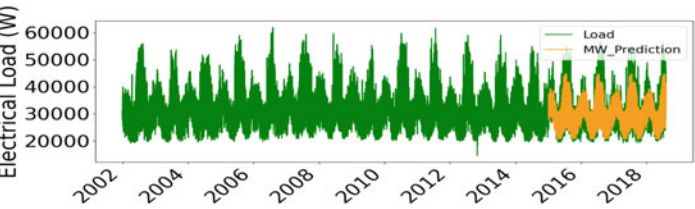
**Very Short Term Definite Time Lag Feature** When considering this feature, a second model is composed considering the mean and standard deviation value of six hours, twelve hours and 24 h immediate previous values of load demand.

5    Result Analysis

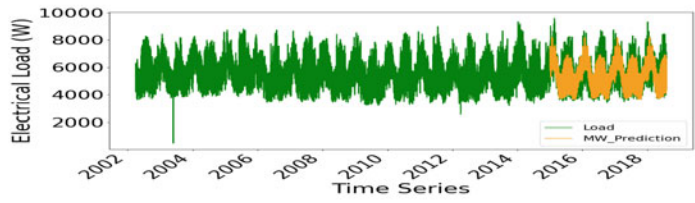
The collected data set were pre processed and trained by the proposed XGBoost machine learning model. The obtained results are represented and compared in the form of three precision metrics.



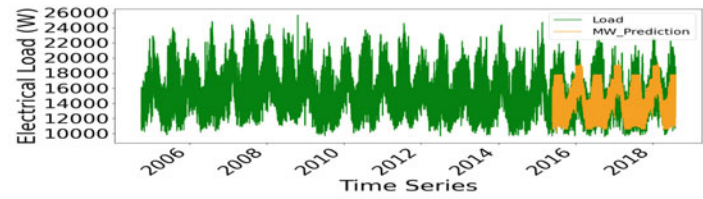
(a) PJM Load



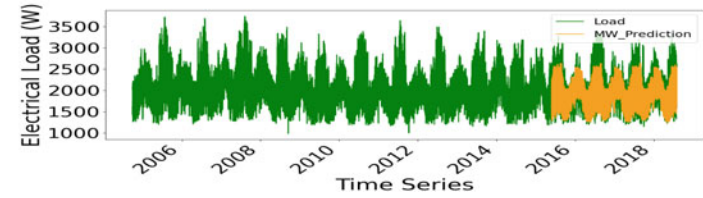
(b) PJME Load



(c) PJMW Load



(d) AEP Load



(e) DAYTON Load

**Fig. 3** Electric load prediction result with PredXGBR-1

**Table 2** Precision metrics obtained from model PredXGBR-1

| Model      | Dataset | Metrics  |           |         |
|------------|---------|----------|-----------|---------|
|            |         | $R^2$    | MSE       | MAPE    |
| PredXGBR-1 | PJM     | 0.71209  | 9515542.0 | 6.87851 |
|            | PJME    | 0.581409 | 443292.8  | 8.59412 |
|            | PJMW    | 0.59473  | 335754.53 | 8.42565 |
|            | AEP     | 0.5762   | 2575187.3 | 8.08461 |
|            | DAYTON  | 0.62165  | 54881.0   | 8.49580 |

$R^2$  value is the measurement of curve fitness indicating the determination of dependant variable by the independent variable. An ideal  $R^2$  value in the scale of 1 should be closest to 1.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} \quad (8)$$

*Mean Squared Error (MSE)* is the mostly used error function used in machine learning algorithms. Basically MSE is determined by the difference between the predicted value and the ground truth by squaring and followed by taking the average from it.

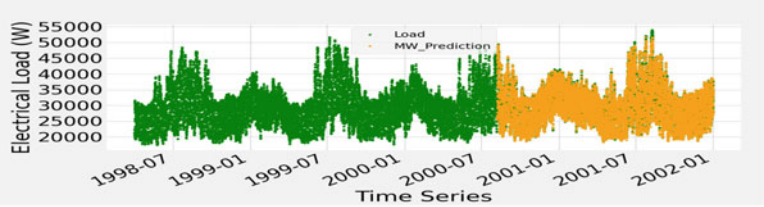
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (9)$$

*Mean Absolute Percentage Error (MAPE)* is slightly different from MSE as it is determined by the absolute value of the difference between predicted value and ground truth, take average and followed by normalized into percentage value. The mathematical expression can be stated as

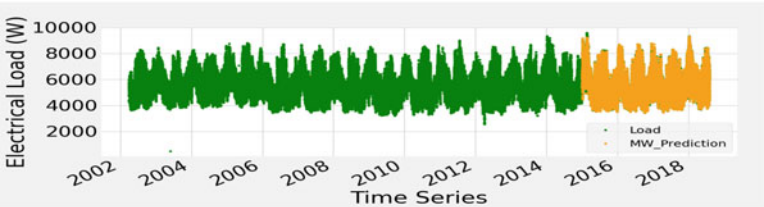
$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{(y_i - \hat{y}_i)}{y_i} \right| \quad (10)$$

Initially model 1 was trained was and tested where PJM interconnection dataset resulted in reasonable  $R^2$  value that is 0.71209. Though the historical record of AEP company showed a very poor value compared to others (0.5762). However the Mean Absolute Percentage Error (MAPE) was calculated worst under the application of PJM East dataset (8.59412) but best under PJM dataset (6.87851). The results of PredXGBR-1 are depicted in Fig. 3. The following Table 2 presents the corresponding performance metrics results from model 1.

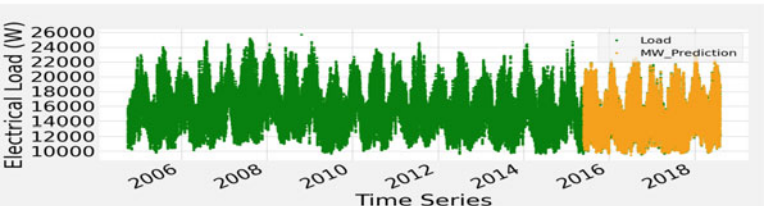
The second model is featured with definite short term lag property. The mean and standard deviation of six hours, twelve hours and twenty-four hours previous value is taken into consideration as feature importance. Consequently the lack of efficiency due to weather condition, peak and off peak hour dependency, seasonal



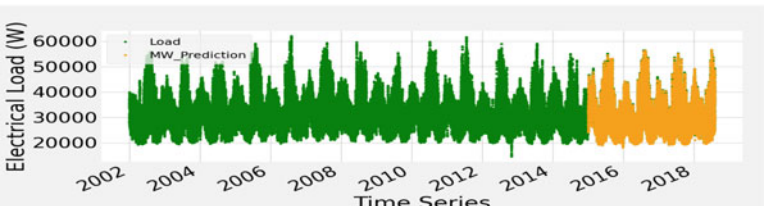
(a) PJM Load



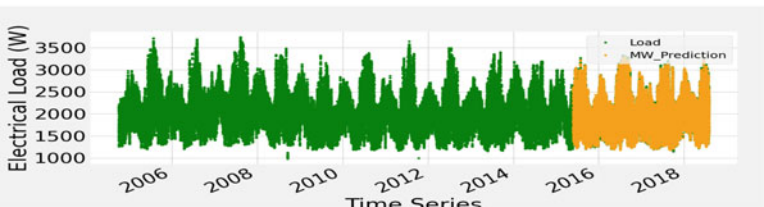
(b) PJME Load



(c) PJMW Load



(d) AEP Load



(e) DAYTON Load

**Fig. 4** Electric load prediction result with PredXGBR-2



**Table 3** Precision metrics obtained from model PredXGBR-2

| Model      | Dataset | Metrics    |            |           |
|------------|---------|------------|------------|-----------|
|            |         | $R^2$      | MSE        | MAPE      |
| PredXGBR-2 | PJM     | 0.991547   | 279381.752 | 1.0776227 |
|            | PJME    | 0.99114491 | 368633.416 | 1.289423  |
|            | PJMW    | 0.98963    | 10979.769  | 1.07894   |
|            | AEP     | 0.9912     | 53421.370  | 0.98304   |
|            | DAYTON  | 0.99134    | 1255.176   | 1.12189   |

load demand and other factors is minimized by this model. The PJM interconnection organization dataset gives the best outcome in the view of  $R^2$  value (0.991547), whereas PJM West zonal dataset gives comparatively lower outcome though the difference is negligible. On the other hand, AEP generation provides the least MAP error (0.98304) hence considered the effective most dataset, whereas PJM East zone provides comparatively higher error. Again the deviation between these two datasets are negligible. The results obtained from PredXGBR-2 model are depicted in Fig. 4.

The above figures represents the load demand prediction with in the last two or three years of the total record. It is evident that prediction status for model 2 with definite time lag feature provides accurate most output. The following Table 3 shows the results of feature 2 model.

It is evident from the two sets of result that PredXGBR-2 provides far better outcome as it is working based on the very short time lag characteristics. In this way the environmental data including temperature, rain, wind etc are unnecessary, and it is able to adapt with any kind of unexpected change in the demand trend.

## 6 Conclusion

This paper works on seeking one of the most accurate short-term electric load forecasting model based on Extreme Gradient Boosting (XGBoost) Regression algorithm. Initially the model extracts a long definite time lag feature which contains days, weeks, years, week of the year, month of the year etc type of data learning characteristics. On the contrary, the proposed model later extracts a “short definite time lag feature” containing the mean and standard deviation of previous few hours demand data. This feature enables the model an explicit training on previous data set. This feature alone can reduce the requirement of many other subsidiary features to be employed. The models PredXGBR-1 and PredXGBR-2 were validated with a wide range of previous data set obtained from five private and regional grid stations of USA with a twenty-years time span. This feature based proposed model is capable of forecasting demand load in hourly basis with about 1.05–1.15% error rate that is, almost 98–99% accuracy.

## References

1. Saqib N, Haque KF, Zabin R, Preonto SN (2019) Analysis of grid integrated PV system as home RES with net metering scheme. In: 2019 international conference on robotics, electrical and signal processing techniques (ICREST), pp 395–399. <https://doi.org/10.1109/ICREST.2019.8644098>
2. Singh AK, Khatoon S, Muazzam M, Chaturvedi D et al (2012) Load forecasting techniques and methodologies: a review. In: 2012 2nd international conference on power, control and embedded systems. IEEE, pp 1–10
3. Lusis P, Khalilpour KR, Andrew L, Liebman A (2017) Short-term residential load forecasting: impact of calendar effects and forecast granularity. *Appl Energy* 205:654–669
4. Zhang L, Wen J, Li Y, Chen J, Ye Y, Fu Y, Livingood W (2021) A review of machine learning in building load prediction. *Appl Energy* 285:116452
5. Haque KF, Zabin R, Yelamarthi K, Yanambaka P, Abdelgawad A (2020) An IoT based efficient waste collection system with smart bins. In: 2020 IEEE 6th world forum on internet of things (WF-IoT), pp 1–5. <https://doi.org/10.1109/WF-IoT48130.2020.9221251>
6. Chen K, Chen K, Wang Q, He Z, Hu J, He J (2018) Short-term load forecasting with deep residual networks. *IEEE Trans Smart Grid* 10(4):3943–3952
7. Aly HH (2020) A proposed intelligent short-term load forecasting hybrid models of ANN, WNN and KF based on clustering techniques for smart grid. *Electr Power Syst Res* 182:106191
8. Singh S, Hussain S, Bazaz MA (2017) Short term load forecasting using artificial neural network. In: 2017 fourth international conference on image information processing (ICIIP). IEEE, pp 1–5
9. Marino DL, Amarasinghe K, Manic M (2016) Building energy load forecasting using deep neural networks. In: IECON 2016-42nd annual conference of the IEEE Industrial Electronics Society. IEEE, pp 7046–7051
10. Ageng D, Huang CY, Cheng RG (2021) A short-term household load forecasting framework using LSTM and data preparation. *IEEE Access* 9:167911–167919
11. Bashir T, Haoyong C, Tahir MF, Liqiang Z (2022) Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Rep* 8:1678–1686
12. Yang J, Zhang X, Bao Y (2021) Short-term load forecasting of central China based on DPSO-LSTM. In: 2021 IEEE 4th international electrical and energy conference (CIEEC). IEEE, pp. 1–5
13. Amarasinghe K, Marino DL, Manic M (2017) Deep neural networks for energy load forecasting. In: 2017 IEEE 26th international symposium on industrial electronics (ISIE). IEEE, pp 1483–1488
14. Alhussein M, Aurangzeb K, Haider SI (2020) Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access* 8:180544–180557
15. Wang Y, Sun S, Chen X, Zeng X, Kong Y, Chen J, Guo Y, Wang T (2021) Short-term load forecasting of industrial customers based on SVM and XGBoost. *Int J Electr Power Energy Syst* 129:106830
16. Zheng H, Yuan J, Chen L (2017) Short-term load forecasting using EMD-LSTM neural networks with a XGBoost algorithm for feature importance evaluation. *Energies* 10(8). <https://www.mdpi.com/1996-1073/10/8/1168>
17. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 785–794
18. Loh WY (2011) Classification and regression trees. *Wiley Interdisc Rev Data Min Knowl Discov* 1(1):14–23

## Chapter 43

# Sn Doped $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$ Films for Uncooled Infrared Detections



Jaime Cardona, Femina Vadakepurathu, and Mukti Rana

**Abstract** We have prepared thin films of Sn doped  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  thin films using a combination of direct current (DC) and radio frequency (RF) magnetron sputtering. We bonded pieces of Si substrate on a Ge target by silver paste to obtain a compound Si-Ge target. Si-Ge and Sn were sputtered simultaneously in a Kurt J. Leskar Pro Line PVD-75 sputtering system in  $\text{Ar} + \text{O}_2$  environment to prepare Sn doped  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  thin films. The sputtering power and process gases' concentrations were varied to obtain appropriate film compositions. Samples were fabricated in a single batch for different tests so that uniform composition and film properties were ensured throughout the samples being analyzed. A room temperature TCR of  $-3.66\%/K$  and resistivity of  $1.885 \times 10^5 \Omega\text{-cm}$  was obtained from the thin film. During our investigation, we found that as the Sn concentration in  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  increased, the absorption also increased. This increased absorption led to decrease the optical bandgap of Sn doped  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  thin films. Higher Sn concentrations in Ge-Si-Sn-O tend to decrease the resistivity as well TCR. Higher  $\text{O}_2$  concentration in Ge-Si-Sn-O, increased the optical bandgap, resistivity, and TCR.

**Keywords** Microbolometer · Germanium-silicon-oxide · Uncooled detector · Infrared detector · TCR · Thin film

---

J. Cardona

Department of Nuclear Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

F. Vadakepurathu (✉) · M. Rana

Division of Engineering, Physics, Mathematics and Computer Sciences and Optical Science Center for Applied Research, 1200 N Dupont Highway, Dover, DE 19901, USA  
e-mail: [fvadakepurathu19@students.desu.edu](mailto:fvadakepurathu19@students.desu.edu)

## 1 Introduction

Infrared detectors can be classified as either photon or thermal detectors. Photon detectors operate through the optical generation of photocarriers or electron–hole pairs. They require that the photon energy be greater than the band gap energy of the material. Thermal generation of carriers results in a dark current and noise that limit the sensitivity of the photon detector.

For photon detectors, the room temperature thermal generation of carriers is significant. To suppress thermal generation of carriers, cryogenic cooling is employed which increases the cost of the detector. On the other hand, thermal detectors are heated by the incident radiation and provide detection through the change in a measurable parameter. The electrical resistance changes with temperature in the case of microbolometers. Microbolometers offer many advantages as compared to other infrared detectors. These include among others, their room temperature operation and superior performance, light weight, compact and smaller in size, low power consumption and cost. Because of these advantages, they are utilized in night vision systems and scientific instruments such as spectrometers and radiometers. Although a-Si and  $\text{VO}_x$  are two of the most widely used sensing layer materials for microbolometers, the microbolometers fabricated with these materials suffer from lower figures of merit because of their low change in resistance with temperature (i.e., TCR) and lower absorption in the bands of interest. The Ge-Si-O based materials exhibit higher TCR but has not been studied extensively [1–3]. In this work, we investigated some of the optical and electrical properties of Ge-Si-O thin films mixed with Sn to form an alloy of Ge-Si-Sn-O for uncooled infrared detection. The thin films were deposited by co-sputtering of Sn and Ge-Si in the Ar + O<sub>2</sub> environment using a combination of direct current (DC) and radio frequency (RF) sputtering system. Investigations of atomic composition, resistivity, TCR, activation energy, transmittance, absorptance and reflectance, optical bandgap, and thermal conductivity of Sn doped  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  thin films were done for using them as the sensing layers of microbolometers.

## 2 Experimental

A Kurt J. Lesker PVD-75 Pro Line series sputtering machine was used for the thin film deposition on various substrates of different sizes. The sputtering system has the capability of using four targets of four-inch diameter simultaneously. We prepared Ge-Si target by bonding lightly doped pieces of p-type Si wafer on top of 99.999% pure Ge target using Ag paste. The area occupied by the Si pieces determined the atomic percentage of Si in Ge which was typically 15–20%. A 99.999% pure Sn was used as the second target. We used Ar + O<sub>2</sub> environment to sputter. A RF power of 195 Watts was used to sputter Si-Ge, while for Sn we used a direct current (DC) power of 30 Watts. The process parameters (Ar + O<sub>2</sub> flow in the chamber, sputtering power and chamber pressure) were varied and optimized to obtain the desired atomic

compositions of thin films. Prior to sputtering, the sputtering chamber was evacuated to a base pressure of  $6 \times 10^{-6}$  Torr. During sputtering, the chamber pressure was kept at 3 mTorr. The film deposition cycle was preceded by an RF etch clean for 600–900 s) to remove any impurities from the target. The sputtering time was varied between 1800 and 3600 s.

To measure the film thickness, we used a Bruker Dektak XT surface profiler. Prior to deposition, we attached a portion of adhesive tape on the cover glass substrate which was removed at the end of the deposition process to create a step for measuring the film thickness.

The atomic composition of thin film was measured by using energy dispersive spectroscopy (EDS) function of a FEI Quanta field emission gun 250 scanning electron microscope. An accelerating voltage of 5 keV was applied from the field emission gun and corresponding elemental compositions were obtained. The EDS system was supplied from Oxford Instrument.

Using a Jandel 3000 four-point probe, we measured the resistivity of the thin films at different temperatures. The sample deposited on cover glass substrate was placed on the metallic chuck of a probe station from micromanipulator Inc. The temperature of the chuck was varied from 17 to 37 °C using a Lakeshore temperature controller and the sheet resistance was measured at the different temperatures. Resistivity ( $\rho$ ) of the thin film was calculated using (1)

$$\rho = R_{\text{sheet}} * t \quad (1)$$

where  $R_{\text{sheet}}$  is the sheet resistance and  $t$  is the thickness of the sample.

Activation energy is related to resistance by (2)

$$R(T) = R_o e^{-\frac{E_a}{k_B T}} \quad (2)$$

where  $R(T)$  is the resistance at temperature  $T$ ,  $R_o$  is the pre-exponential factor, and  $E_a$  is the activation energy. From the variation of resistivity with temperature, an Arrhenius plot of  $\ln(\rho)$  versus  $1/kT$  was developed and slope of the curve was used to calculate the activation energy. The activation energy was used to calculate the TCR.

For the optical measurements we used the setup depicted in Fig. 1. By using a silicon carbide light source and monochromator (MS257 from Newport), different wavelengths of light were passed through the sample, this enabled the testing of the transmittance and reflectance of the samples. First, the measurement was made for light passing through air. This became the total amount of light, from which the transmittance and reflectance were determined as a percentage of that total light. Depending on the wavelengths, we used thin films deposited on various substrates like glass for the range 1–3  $\mu\text{m}$ , sapphire for 2–5  $\mu\text{m}$ , and germanium for 6–10  $\mu\text{m}$ . The absorption can then be calculated by using (3)

$$1 = T + R + A \quad (3)$$

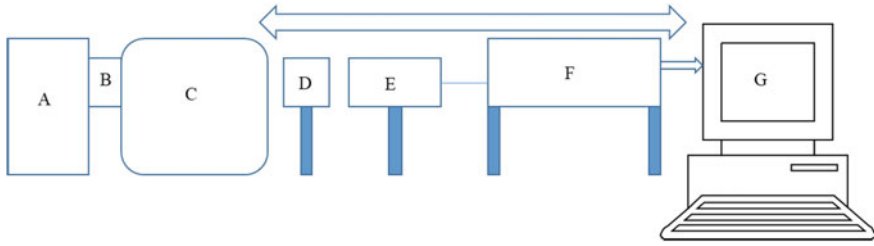


**Fig. 1** Transmittance and reflectance measurement setup. **A** Blackbody (Silicon carbide bar) light source, **B** Newport optical chopper, **C** Monochromator, **D** Thin Film, **E** Newport Pyroelectric detector, **F** SRS SR810 DSP Lock-In Amplifier, **G** Computer

From the transmittance and reflectance data, the absorption coefficient ( $\alpha$ ) was determined. The absorption coefficient is the value that describes how different wavelengths of light pass through a material before it is absorbed. A small absorption coefficient would mean that the material is transparent to the light. Absorption coefficient was determined from the relationship mentioned in (4)

$$T = \frac{(1 - r)^2 \exp(-\alpha x)}{1 - r^2 \exp(-2\alpha x)} \quad (4)$$

where  $T$  is transmittance,  $r$  is reflectance, and  $x$  is film thickness.

The optical band gap,  $E_g$ , was determined by fitting  $\alpha$ , to Tauc's equation for semiconductors mentioned in (5).

$$(\alpha h\nu)^{\frac{1}{n}} = B(h\nu - E_g) \quad (5)$$

where,  $\nu$  is the frequency,  $B$  is a constant, and  $n$  can be  $\frac{1}{2}$  or 2 depending on whether the material is a direct or indirect bandgap semiconductor. The optical band gap was found by drawing a tangent in the linear region of the  $(h\alpha\nu)^2$  versus the photon energy ( $h\nu$ ) and extrapolating it to the  $x$ -axis (Fig. 2).

### 3 Results and Discussions

Figure 3 shows the  $\ln(\rho)$  versus  $1/kT$  plot for sample  $\text{Ge}_{0.36}\text{Si}_{0.04}\text{Sn}_{0.11}\text{O}_{0.43}$ . From this plot, we determined an activation energy of 0.2718 eV. The room temperature TCR from this activation energy was found to be  $-3.66\%/K$ . We also found the resistivity of the sample as  $1.885 \times 10^5 \Omega\text{-cm}$ . This value of TCR is lower than the a- $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  film reported by Cheng et al. [2]. The resistivity value is of three orders higher as compared to amorphous  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  films. It was reported that higher noise is associated with higher resistivity value for a- $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  films [1]. Clearly, we need more investigations on this.

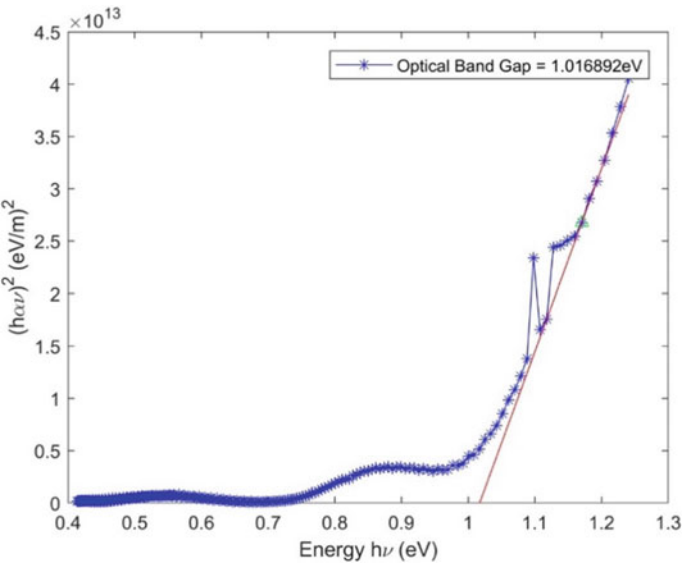


Fig. 2 Optical bandgap determination of thin films

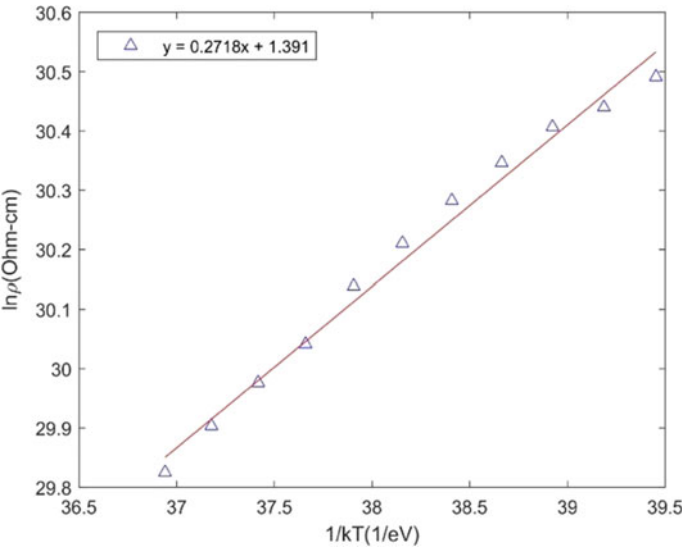


Fig. 3 Plot of  $\ln(\rho)$  versus  $1/k_B T$  for  $\text{Ge}_{0.36}\text{Si}_{0.04}\text{Sn}_{0.11}\text{O}_{0.43}$

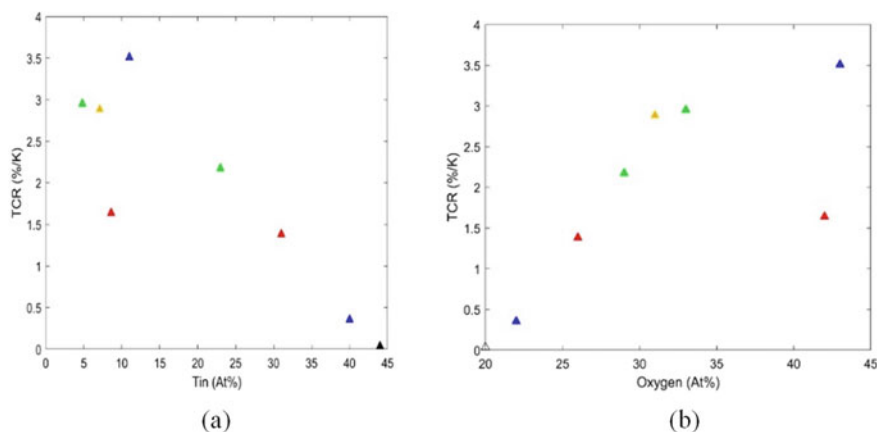


**Fig. 4** Variation of TCR with **a** Sn, **b**  $O_2$  concentrations in Ge-Si-Sn-O

Addition of Sn in  $a\text{-Ge}_x\text{Si}_{1-x}\text{O}_y$  decreased the activation energy and TCR. This can be seen from Fig. 4a where the Sn concentration in  $a\text{-Ge}_x\text{Si}_{1-x}\text{O}_y$  increased from 11% to 40% and the TCR decreased from  $-3.55\%/K$  to  $-0.4\%/K$ . Increasing the  $O_2$  concentration in  $a\text{-Ge}_x\text{Si}_{1-x}\text{O}_y$  increased the TCR as it can be seen from Fig. 4b. This result is consistent with previously reported value for  $a\text{-Ge}_x\text{Si}_{1-x}\text{O}_y$  thin films [1–3], although at some atomic concentrations of Sn and  $O_2$ , we see deviations from the original trend. The reasons behind these are currently unknown and need to be investigated further.

Figure 5 shows the transmittance, reflectance and absorption for  $\text{Ge}_{38}\text{Si}_{106}\text{Sn}_{23}\text{O}_{29}$  thin films. During our investigation, we also noticed that, as the Sn concentration increased in Ge-Si-Sn-O, absorption increased as well. This increased absorption led to decrease in the optical bandgap of Ge-Si-Sn-O samples. Decrease in optical bandgap with higher Sn concentrations, tend to decrease the resistivity as well as TCR. Higher  $O_2$  concentration in Ge-Si-Sn-O increased the optical bandgap as well as the resistivity and TCR.

From the optical transmittance data, we found that when Sn concentration was lower than 30%, the transmittance was much higher than when it was more than 30%. At Sn concentrations more than 40%, all IR radiation was absorbed.

Figure 6a and b show the variation in band gap energy with variations of  $O_2$  and Sn concentrations respectively. From these plots, it can be seen that a higher Sn concentration led to a lower band gap energy and vice-versa for higher  $O_2$  concentrations.

Table 1 compares the results of  $\text{Ge}_{0.36}\text{Si}_{0.04}\text{Sn}_{0.11}\text{O}_{0.43}$  with other well-known sensing layer materials.



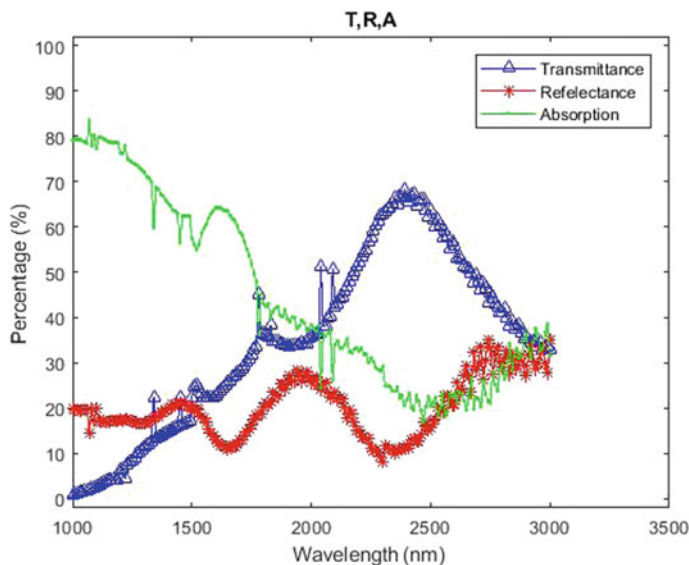


**Fig. 5** Variations of the transmittance, reflectance and absorption for  $\text{Ge}_{38}\text{Si}_{06}\text{Sn}_{23}\text{O}_{29}$  thin film

## 4 Conclusions

$\text{Ge}_x\text{Si}_y\text{Sn}_z\text{O}_{1-x-y-z}$  thin films were deposited by combining RF magnetron and DC sputtering and characterized to determine some of their electrical and optical properties for using them as the sensing layer in microbolometer. We optimized the atomic composition of  $\text{Ge}_x\text{Si}_y\text{Sn}_z\text{O}_{1-x-y-z}$  thin films to obtain a higher TCR value along with moderate resistivity. We found that  $\text{Ge}_{0.36}\text{Si}_{0.04}\text{Sn}_{0.11}\text{O}_{0.43}$  thin film had a room temperature TCR  $-3.66\%/K$  and resistivity of  $1.885 \times 10^5 \Omega\text{-cm}$ . From the optical transmittance, reflectance and absorption properties, we determined the optical band gap energy of  $\sim 1 \text{ eV}$  in  $\text{Ge}_{0.36}\text{Si}_{0.04}\text{Sn}_{0.11}\text{O}_{0.43}$ . The TCR value obtained here, found to have similar value like that of two of the most widely used sensing layer materials of microbolometers-vanadium oxide and amorphous silicon. The results show that the optical and electrical properties of  $\text{Ge}_x\text{Si}_y\text{Sn}_z\text{O}_{1-x-y-z}$  alloys depend greatly on the amount of Sn that is present in that compound. When large amounts of Sn are present, the film will absorb more IR radiation. Higher absorption in the low and medium IR region will help to increase the performance of microbolometer although it was observed that greater Sn concentration leads to a lower TCR. Thus, we need more investigation to find a more optimum atomic composition which will show higher TCR, moderate resistivity and lower noise.

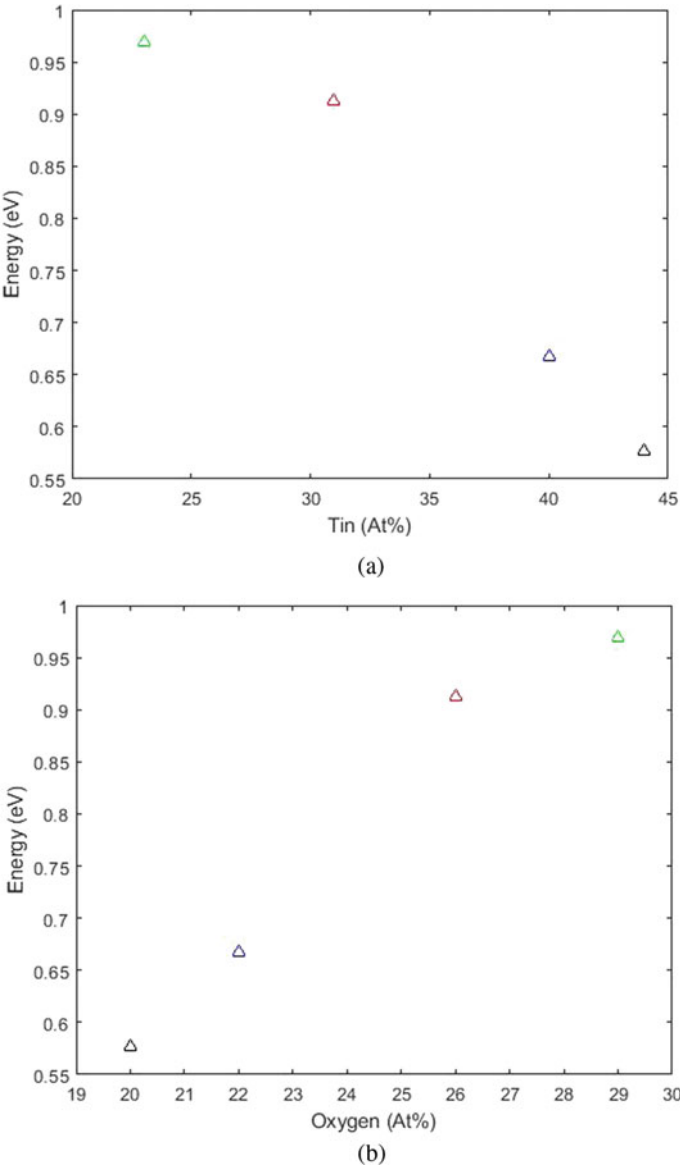


**Fig. 6** **a** Variation of optical bandgap with Sn concentrations in Ge-Si-Sn-O, **b** Variation of optical bandgap with O<sub>2</sub> concentrations in Ge-Si-Sn-O

**Table 1** Comparison of different sensing layer materials

| Material                                                          | Resistivity, $\rho$<br>( $\Omega\text{-cm}$ ) | Activation energy<br>(eV) | TCR (%/K)         | References          |
|-------------------------------------------------------------------|-----------------------------------------------|---------------------------|-------------------|---------------------|
| $\text{Ge}_{0.36}\text{Si}_{0.04}\text{Sn}_{0.11}\text{O}_{0.43}$ | $1.885 \times 10^5$                           | 0.2718                    | – 3.66            | Current work        |
| $\text{V}_2\text{O}_5$                                            | 1.7                                           | 0.214                     | – 2.8             | Kumar et al. [4]    |
| a-SiGe                                                            | 40                                            | 0.1532                    | – 2               | Yon et al. [5]      |
| Poly-SiGe                                                         | N/A                                           | 0.1463                    | – 1.91            | Dong et al. [6]     |
| Poly-SiGe (CVD deposited)                                         | N/A                                           | 0.1532                    | – 2               | Sedky et al. [7]    |
| a-Si:H                                                            | N/A                                           | 0.214–0.2987              | – 2.8 to – 3.9    | Syllaios et al. [8] |
| a-Si:H (sandwich)                                                 | N/A                                           | 0.3447                    | – 4.5             | Unewisse et al. [9] |
| a- $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$                         | $2.45 \times 10^2$ –<br>$3.34 \times 10^2$    | 0.3723 and 0.4925         | – 4.86 and – 6.43 | Cheng et al. [2]    |
| $\text{V}_{0.95}\text{W}_{0.05}$                                  | 40                                            | 0.3141                    | – 4.1             | Han et al. [10]     |

**Acknowledgements** This work is partially supported by Army Research Office, Grant Number 72513-RT-REP.

## References

1. Rana MM, Butler DP (2006) Radio frequency sputtered  $\text{Si}_{1-x}\text{Ge}_x$  and  $\text{Si}_{1-x}\text{Ge}_x\text{O}_y$  thin films for uncooled infrared detectors. *Thin Solid Films* 514:355–360
2. Cheng Q, Almasri M (2009) Silicon germanium oxide ( $\text{Si}_x\text{Ge}_{1-x}\text{O}_y$ ) infrared material for uncooled infrared detection. *Proc SPIE* 7298:72980K
3. Iborra E, Clement M, Herrero LV, Sangrador J (2002) IR uncooled bolometers based on amorphous  $\text{Ge}_x\text{Si}_{1-x}\text{O}_y$  on silicon micromachined structures. *IEEE/JMEMS* 11:322–329
4. Rajendra Kumar RT et al (2003) Room temperature deposited vanadium oxide thin films for uncooled infrared detectors. *Mat Res Bull* 38:1235–1240
5. Yon JJ et al (2010) Low resistance a-SiGe-based microbolometer pixel for future smart IR FPA. In: *Proceedings SPIE, infrared technology and applications XXXVI*, vol 7660, p 76600U. <https://doi.org/10.1117/12.850862>
6. Dong L, Yue R, Liu L (2003) An uncooled microbolometer infrared detector based on Poly-SiGe thermistor. *Sens Actuators, A* 105:286–292
7. Sedky S, Fiorini P, Caymax M, Verbiest A, Baert C (1998) IR bolometers made of polycrystalline silicon germanium. *Sens Actuators, A* 66:193–199
8. Syllaios AJ et al (2001) Amorphous silicon microbolometer technology. In: *Symposia proceedings materials research society*, vol 609-A14.4. 10.1557
9. Unewisse MH, Craig BI, Watson RJ, Reinhold O, Liddiard KC (1995) The growth and properties of semiconductor bolometers for infrared detection. *Proc SPIE* 2554:43–54
10. Han Y-H, Kim K-T, Shin H-J, Moon S (2005) Enhanced characteristics of an uncooled microbolometer using vanadium-tungsten oxide as a thermometric material. *Appl Phys Lett* 86(254101):1–3. <https://doi.org/10.1063/1.1953872>

# Author Index

## A

Abedin, Md. Ruhul, 15  
Ahmad, Mohiuddin, 57, 153, 195, 291, 343, 447  
Ahmed, Tofael, 535  
Akhand, Zebel-E-Noor, 497  
Ali Hossain, Md., 163  
Ali, Md. Liakot, 421  
Ali, Shahnewaz, 209, 393  
Al Mahmud, Abdullah, 461  
Al-Qahtani, Kholod Saaed, 95  
Al Rafi, Abdullah, 263  
Anjum, Nipa, 181  
Anwar, Afrin Binte, 153  
Arman, Nafiz, 15  
Arrafuzzaman, Al, 365  
Ashrafal Alam, Md., 235  
Ayman, Syeda Umme, 365  
Azad, Katha, 497

## B

Barua, Labanya, 535  
Basu, Sujit, 331  
Begum, Mahbuba, 1  
Billah, Lubaba Binte, 277  
Bin Nazir, Junayed, 317  
Biswas, Richard Victor, 107  
Brown, Cameron, 209

## C

Cardona, Jaime, 547  
Chowdhury, Md. Reasad Zaman, 251  
Chowdhury, Moinul H., 139

Crawford, Ross, 209

## D

Das, Sufal, 171  
Dayoub, Feras, 393

## F

Fahim, Md. Zubayer Ahmed, 497  
Faisal Khan, Asif, 305  
Faruque, Md. Omar, 435  
Faysal Ahamed, Md., 223

## H

Habib, Adria Binte, 83  
Hafiz Ahamed, Md., 163  
Halder, Kalyan Kumar, 461  
Haque, Khandaker Foysal, 521  
Hasan, K. M. Azharul, 509  
Hasan, Md. Kamrul, 343  
Hasan, Mehedi, 331  
Hasan, Syed Rakib, 15  
Hassan, Rakibul, 263  
Hellen, Nakayiza, 235  
Hossain, Md. Anwar, 153  
Hossain, Mir Sakhawat, 43  
Hossain, Muhammad Iqbal, 83  
Hossain, Tahmim, 223  
Hossen, Monir, 331  
Hridhee, Rubaiyat Alim, 139

## I

Ibtesham, Md. Abrar, 277

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

M. Ahmad et al. (eds.), *Proceedings of International Conference on Information and Communication Technology for Development*, Studies in Autonomic, Data-driven and Industrial Computing, <https://doi.org/10.1007/978-981-19-7528-8>

Islam, A. K. M. Nazmul, 139  
 Islam, Md. Farhadul, 251  
 Islam, Monira, 435  
 Islam, Rima, 153  
 Islam, S. M. Mohidul, 379  
 Islam, Tanvir, 139

## J

Jahirul Islam, Md., 305  
 Jonmohamadi, Yaqub, 209  
 Jui, Julakha Jahan, 471

## K

Khalid, Md. Saifuddin, 71  
 Khan, Ishrat, 407  
 Khan, Md. Rafin, 83  
 Kundu, Paromita, 355

## M

Madhurja, Masshura Mayashir, 343  
 Mamun, Khondaker A., 139  
 Mandal, Nibir Chandra, 125  
 Masudul Ahsan, Sk. Md., 181  
 Mina, Md. Faisal, 483  
 Molla, M. M. Imran, 471  
 Mollik, Debalina, 153  
 Mondal, Sohag Kumar, 435  
 Mukhim, Banteilang, 171  
 Mumenin, Nasirul, 251

## N

Nahiduzzaman, Md., 263  
 Nandi, Pallab Kumar, 195  
 Nisher, Sumiya Akter, 29  
 Nourin, Nymphia, 355

## P

Pandey, Ajay K., 209, 393  
 Paul, Robi, 317  
 Podder, Nitun Kumar, 471  
 Prapty, Aroni Saha, 509  
 Purnendu, Prodip Kumar Saha, 153

## R

Rabiul Islam, Md., 263  
 Rahman, Afridi Ibn, 497  
 Rahman, Faria, 57  
 Rahman, Jeba Fairouz, 447  
 Rahman, Maksuda, 343  
 Rahman, Md. Asadur, 355, 365

Rahman, Md. Mostafizur, 29  
 Rahman, M. M. Hafizur, 95  
 Rana, Humayan Kabir, 471  
 Rana, Mukti, 547  
 Ripan, Rony Chowdhury, 139  
 Roark, Amalie, 71  
 Roberts, Jonathan, 209  
 Robiul Islam, Md., 223  
 Rokib Raihan, Md., 291  
 Roy, Amit Dutta, 483

## S

Sabuj, Hasibul Hasan, 235  
 Sadi, Muhammad Sheikh, 421  
 Saima, Sk., 355  
 Sakib, Abu Noman Md., 181  
 Sakib, Shadman, 83  
 Sarda, Anirudh, 497  
 Sarkar, Ovi, 223  
 Sarker, Farhana, 139  
 Shahariar, G. M., 125  
 Shawon, Md. Tanvir Rouf, 125  
 Shemanto, Tanber Hasan, 277  
 Shill, Pintu Chandra, 407  
 Sumon, Md. Mehedy Hasan, 421  
 Syfullah, Khalid, 223

## T

Tahiat, Hiya Tasfia, 497  
 Takeda, Yu, 209  
 Talukder, Kamrul Hasan, 379  
 Tretow-Fish, Tobias Alexander Bang, 71  
 Turja, Mrinmoy Sarker, 435  
 Tushi, Lubaba Tasnia, 497

## U

Uddin, Md. Bashir, 483  
 Uddin, Mohammad Shorif, 1

## V

Vadakepurathu, Femina, 547

## Y

Yousuf, Mohammad Abu, 251  
 Yusuf, Md. Salah Uddin, 435

## Z

Zabed, Md. Akib, 43  
 Zabin, Rifat, 521, 535